Машинное обучение - состояние и перспективы

© Д.П. Ветров

VetrovD@yandex.ru

1 Задачи машинного обучения

Теория машинного обучения зародилась практически одновременно с появлением первых компьютеров и на протяжении последних 70 лет является активно развивающейся дисциплиной. Ее постоянное развитие вызвано ростом возможностей современных вычислительных систем, еще более стремительным ростом объемов данных, доступных для анализа, а также постоянным расширением области применения методов машинного обучения на все более широкий класс задач обработки данных. Машинное обучение работает с объектами - элементарными единицами данных, естественным образом, возникающими в конкретных задачах, которые характеризуются наблюдаемыми переменными $\vec{x}$ и скрытыми переменными $\vec{t}$, принимающими значения из некоторых заранее известных множеств. Главной задачей машинного обучения является автоматическое определение взаимозависимостей между наблюдаемыми и скрытыми переменными объекта, с тем, чтобы для произвольного объекта по его наблюдаемым компонентам можно было оценить возможные значения скрытых компонент. Как правило, возможные взаимозависимости задаются заранее с помощью параметрических решающих правил, определяемых значением параметров (весов) $\vec{w}$. Конкретные значения $\vec{w}$ определяются в ходе обучения с использованием обучающей выборки, представляющей собой множество объектов с известными наблюдаемыми и скрытыми переменными (X^{tr}, T^{tr}) (обучение с учителем) или только наблюдаемыми переменными X^{tr} (обучение без учителя). При этом задача определения весов решающего правила $\vec{w}$ по обучающей выборке называется задачей обучения или настройки (training), а задача определения допустимых значений скрытой переменной $\vec{t}$ по заданым наблюдаемым компонентам $\vec{x}$ объекта и заданным весам решающего правила $\vec{w}$ — задачей вывода (inference). Обычно (но не обязательно) предполагается, что каждый объект описывается одним и тем же набором переменных, а номенклатура наблюдаемых и скрытых переменных для всех объектов одинакова. Примером такой стандартной задачи является задача классификации, в которой скрытая переменная для каждого объекта одна и принимает значения из конечного дискретного множества, а каждая наблюдаемая переменная может принимать действительные, либо (реже) дискретные значения. Если скрытая переменная объекта является не дискретной, а непрерывной, задача называется задачей восстановления регрессии, являющейся еще одной стандартной и хорошо изученной задачей машинного обучения.

В разное время предпринимались неоднократные попытки ввести некоторый универсальный язык описания различных постановок и методов решения задач машинного обучения. Начиная с 90ых гг прошлого века широкое распространение получил т.н. байесовский формализм. При его использовании предполагается, что зависимости между наблюдаемыми переменными объекта, весами решающего правила и скрытыми переменными объекта моделируются с помощью совместного распределения на эти группы переменных $p(X, T, \vec{w})$. Если нас интересует только задача определения скрытых переменных по наблюдаемым, рассматривают дискриминативные модели (discriminative models) $p(T, \vec{w}|X)$. Значения наблюдаемых переменных X в этом случае не моделируются, предполагаясь известными на всех этапах решения задачи, и совместное распределение становится проще. В стандартных постановках задачи машинного обучения предполагалось, что скрытые переменные каждого объекта зависят только от наблюдаемых переменных этого объекта, причем вид зависимости определяется параметрами $\vec{w}$. Это соответствует представлению

$$p(T, \vec{w}|X) = \prod_{i=1}^{n} p(\vec{t}_i|\vec{x}_i, \vec{w})p(\vec{w}).$$

При использовании такого формализма задача настройки параметров $\vec{w}$ решается, например, нахождением наиболее вероятного значения

$$\vec{w}_{MP} = \arg\max p(\vec{w}|X^{tr}, T^{tr}) =$$
$$\arg\max \frac{p(T^{tr}, \vec{w}|X^{tr})}{p(T^{tr}|X^{tr})} = \arg\max p(T^{tr}, \vec{w}|X^{tr}),$$

а задача вывода — путем нахождения[1]

$$\hat{\vec{t}}(\vec{x}) = \arg\max p(\vec{t}|\vec{x}, \vec{w}_{MP}).$$

Таким образом, для формулировки и решения задачи машинного обучения нам достаточно знать две функции: $p(\vec{t}|\vec{x}, \vec{w})$ и $p(\vec{w})$. Если с первой функцией, называемой функцией правдоподобия (likelihood), проблем обычно не возникает, т.к. она естественным образом характеризует степень «истинности» полученного прогноза на скрытую переменную, то вторая компонента, наызваемая априорным распределением (weight prior) или регуляризатором (regularizer), долгое время вызывала споры. В самом деле, меняя априорное распределение, мы влияем на результат процедуры настройки, т.е. на $\vec{w}_{MP}$. При этом способ адекватного определения априорного распределения неочевиден. В 90ые гг. в ряде работ [6] было убедительно показано, что априорное распределение является эффективным способом контроля сложности решающего правила и позволяет осуществлять регуляризацию процедуры настройки. Вместо нахождения весов, обеспечивающих наименьшую ошибку прогноза на обучающей выбобрке (что чревато эффектом переобучения (overfitting)) мы жертвуем толикой точности ради сохранения способности обеспечить ту же ошибку прогноза на других объектах генеральной совокупности. Оказалось, что в любой модели машинного обучения можно выделить самое простое решающее правило (например, отвечающее нулевым значениям весов), в которое помещается мода унимодального априорного распределения. Чем больше расстояние текущих значений весов от моды, тем меньше значение $p(\vec{w})$. Ширина же априорного распределения задается параметром регуляризации, который может быть сравнительно эффективно найден процедурой скользящего контроля (cross-validation) или байесовской процедурой выбора модели (Bayesian model selection). Еще более привлекательным свойством байесовского формализма оказалась возможность учитывать многочисленные априорные знания о возможных зависимостях между наблюдаемыми и скрытыми переменными объектов, которые имеются во многих прикладных задачах. Например, известно, что надежность заемщика (прогнозируемая переменная) должна положительно коррелировать с его доходом и образованием (наблюдаемые переменные). Такие

[1]Строго говоря, полностью байесовские процедуры настройки и вывода предполагают нахождение апостериорных распределений $p(\vec{w}|X^{tr}, T^{tr})$ и $p(\vec{t}|\vec{x}, X^{tr}, T^{tr})$ вместо соответствующих точечных оценок, поэтому последние можно рассматривать как детерминированные приближения случайных величин, например, в смысле дивергенции Кульбака-Лейблера

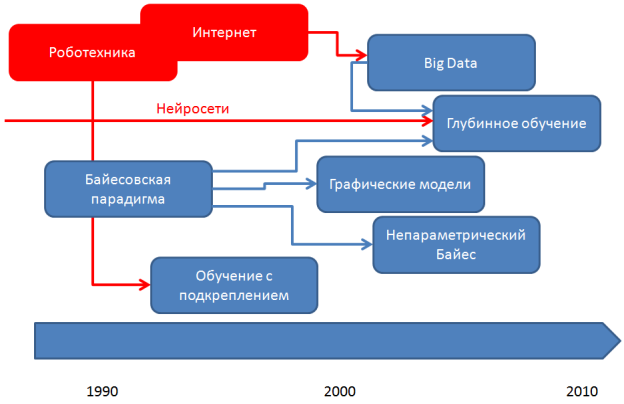


Рис. 1: Приблизительная хронологическая карта появления новых направлений в машинном обучении

«подсказки» алгоритмам общего назначения, выраженые в виде априорного распределения на $\vec{w}$, позволили добиться значительного увеличения точности и снизить эффект переобучения, благодаря адаптации их под специфику конкретной задачи.

Можно показать, что практически любую задачу машинного обучения возможно (с большей или меньшей степенью естественности) свести к такому формализму. Это, в свою очередь, открывает унифицированный способ анализа различных моделей машинного обучения, например, с целью исследования их обобщающей способности или выработки эффективных приближеных методов настройки и вывода общего назначения.

2 Современные направления развития теории машинного обучения

С конца 90ых гг. байесовский формализм при описании алгоритмов машинного обучения получил всеобщее признание [1]. В рамках него удалось разработать ряд общих методов для оценки апостериорных распределений, байесовского вывода, автоматического выбора модели и пр. Не менее важным успехом байесовского формализма стала возможность успешного обобщения результатов и методов классического машинного обучения на совершенно новые задачи (см. например, [2]).

2.1 Глубинное обучение

Методы глубинного обучения (deep learning) являются попыткой реинкарнации нейронных сетей, с конца 80ых гг. прошлого века переживающих кризис. Причинами

кризиса традиционных нейронных сетей стали: критическая зависимость качества настройки весов сети от выбора начального приближения и, как следствие, проблемы с воспроизводимостью «успешных» результатов, публиковавшихся в научных журналах; большая подверженность переобучению вкупе со слабыми возможностями контроля обобщающей способности сети; большоее количество локальных минимумов функционала качества, большинство из которых оказывались плохими. С другой стороны, неоспоримой сильной стороной нейронных сетей явилось открытие метода обратного распространения ошибки (backpropagation), позволявшего отслеживать влияние внутренних слоев сети на качество прогноза скрытых переменных объектов обучающей выборки.

Во второй половине 00ых гг стало активно развиваться направление, получившее название глубинного обучения [4]. В его основе лежат нейроные сети, претерпевшие значительные изменения:

- Глубиннное обучение строит не дискриминативные, а порождающие модели (generative models), в которых моделируется общее распределение $p(X, T, \vec{w})$, в отличие от дискриминативных моделей, позволяющее, например, генерировать новые объекты.
- В наиболее распространенной постановке все переменные объектов предполагаются бинарными. Это облегчает моделирование зависимостей между переменными объекта.
- Каждый слой сети сначала обучается независимо, проходя процедуру предобучения (pre-training). Это позволяет «нащупать» хорошее начальное приближение для последующего запуска алгоритма обратного распространения ошибки. Каждый слой, в зависимости от выбранной модели, представляет собой ограниченную машину Больцмана (restricted Boltzmann machine) или сверточную сеть (convolutional network).
- Для обучения используются сотни тысяч и миллионы объектов. Такие гигантские выборки позволяют настраивать сети с десятками тысяч параметров, без риска переобучения. Обученные таким образом сети, не просто позволяют моделировать сложные объекты (например, тексты или изображения), но и генерируют в процессе обучения информативные признаковые описания, которые могут быть использованы другими, более простыми алгоритмами машинного обучения в качестве наблюдаемых переменных объекта.

Методология глубинного обучения позволила добиться невиданых ранее результатов при обучении на больших и сверхбольших объемах данных. В настоящее время она является одним из наиболее перспективных путей развития машинного обучения.

2.2 Непараметрические байесовские методы

Традиционно, методы непараметрической статистики определялись как раздел статистики, в которой число параметров, описывающих данные (например, парамеры плотности распределения объектов) не фиксированно, а растет с ростом числа объектов. Чтобы разъяснить принципы работы непараметрических байесовских методов (non-parametric Bayes), рассмотрим задачу определения числа кластеров (скоплений объектов) в растущей выборке объектов. Данная задача тем более актуальна, что общепринятых методов определения, а из скольки же кластеров состоит даже зафиксированная выборка, на сегодняшний день не существует. Чем больше объектов поступает в наше распоряжение, тем с большим разрешением мы можем находить в них структуру, выделяя кластеры схожих между собой объектов. В случае достаточно неоднородной выборки число кластеров должно постепенно увеличиваться по мере поступления новых объектов. Возникает вопрос, можно ли задать наши представления о том, как быстро должно расти число кластеров с ростом данных (чтобы их не было слишком много или слишком мало) и как, глядя на выборку объектов, учесть эти представления. Формально, ответ может быть задан знаменитой формулой Байеса, которая как раз и объединяет наши априорные представления с текущими наблюдениями

$$\text{Posterior} = \frac{\text{Likelihood} \times \text{Prior}}{\text{Evidence}}.$$

В непараметрическом случае, нам необходимо задать распределение над всевозможными разбиениями произвольного количества объектов. Такое распределение (как и многие другие в непараметрических байесовских методах) задается с помощью случайных процессов. В данном случае, это процесс Дирихле (Dirichlet process), также известный как процесс китайского ресторана (Chinese restraunt process) [7].[2] С его помощью, удается не только расчитать для любого разбиения проивольного числа объектов на кластеры его априорную вероятность, но и учесть характеристики объектов (их наблюдаемые переменные), чтобы

[2] Вообще, терминология в непараметрическом Байесе грешит восточными гастрономическими наклонностями. Известен еще процесс китайской франшизы [ресторанов] и процесс индийского буфета :)

перейти к апостериорному распределению на всевозможные разбиения. Как это часто бывает при применении байесовских методов, апостериорное распределение имеет острый пик, который соответствует устойчивому разбиению выборки объектов на некоторое число клстеров. Фактически, процесс Дирихле позволяет задавать распределения над всевозможными дискретными распределениями. При выводе используются приближенные методы Монте-Карло с марковскими цепями (Markov chain Monte Carlo) и методы вариационного вывода (variational inference). Описанная схема допускает многочисленные обобщения на случай иерархий кластеров, множественных выборок, и др.

2.3 Обучение с подкреплением

Еще одной активно развивающейся областью машинного обучения является обучение с подкреплением, предназначенное для обучения агентов (автономных модулей, самостоятельно принимающих решения в реальном времени на основании располагаемых данных) в условиях неопределенности, порождаемой, как неполнотой информации об окружающей обстановке, так и возможными действиями других агентов. В зависимости от текущего состояния среды и действий агентов расчитывается функция выгоды, которую получит агент в следующий момент времени. В роли наблюдаемых переменных объекта выступает информация, располагаемая агентом, а скрытыми переменными являются долгосрочные оценки полученной выгоды. Важным достоинством алгоритмов обучения с подкреплением является возможность обучения агента «с нуля» за счет балансируемого сочетания режимов «исследование-использование» (exploration-exploitation) и выучивания стратегий, позволяющих жертвовать малым сейчас ради получения большей выгоды в дальнейшем. Алгоритмы обучения с подкреплением нашли широкое применение не только в таких традиционных областях как роботехника, но и, например, на фондовых рынках.

2.4 Анализ больших объемов данных

Термин «большие данные» (англ. big data) вошел в употребление в конце 2000-х годов, когда стал возможным сбор и хранение огромных объемов данных. Феномен больших данных можно наглядно продемонстрировать на примере большого адронного коллайдера, который в прошлом году произвел около 25 петабайт экспериментальных данных [3]. Традиционные методы машинного обучения не всегда применимы для анализа выборок такого размера, поскольку в них зачастую неявно предполагается, что вся выборка помещается в памяти компьютера, или же они имеют недостаточно высокие показатели масштабируемости (скорости роста вычислительной сложности в зависимости от размера выборки). Для преодоления этих ограничений часто используются приемы из следующих категорий:

- Распараллеливание. Независимые части алгоритма могут выполняться параллельными обработчиками (в т.ч. на разных компьютерах) и в произвольном порядке. В некоторых случаях параллельной реализации классичесского алгоритма может быть достаточно для конкретной задачи. В той или иной форме параллельность лежит в основе практически всех вычислительных систем, ориентированных на большие данные. Примечательно, что параллельность накладывает существенные ограничения на взаимодействие между обработчиками, так как накладные расходы на «общение» между ними может превышать выигрыш от использования большого вычислительного кластера.

- Аппроксимация. Известно, что многие сложные задачи могут быть решены приближенно с достаточно большой (а иногда и контролируемой) точностью, достаточной для данного эксперимента. Примерами могут служить фильтр Блума или приближенный алгоритм поиска ближайшего соседа, которые допускают ошибки первого рода, но имеют существенно более низкую вычислительную сложность чем их «точные» аналоги.

- Стохастичность (рандомизация). При наличии большого числа независимых объектов в выборке, многие необходимые статистики могут быть оценены по случайной подвыборке, при этом сохраняются теоретические гарантии оптимальности и сходимости алгоритма. В случае, если выбирается подвыборка некоторого фиксированного размера это позволяет получать алгоритмы с сублинейной масштабируемостью. Наиболее известным алгоритмом, где применяется данный подход, является метод стохастического градиентного спуска.

В последнее время стали также набирать популярность т.н. потоковые алгоритмы (streaming algorithms, online learning), способные обучаться инкрементально в режиме реального времени на постоянно поступающих данных

без необходимости хранить их где-либо в памяти. Спрос на них возникает, как правило, в приложениях, где данные поступают в таких количествах и с такой скоростью, что нет никакой возможности сохранять их, по крайней мере, надолго. С такими задачами анализа данных сталкиваются, например, исследователи в ЦЕРНе, где данные генерируются со скоростью 700 мегабайт в секунду.[3]

3 Вероятностные графические модели

Одним из наиболее впечатляющих результатов использования байесовского формализма для описания задач обработки данных явился аппарат вероятностных графических моделей, в общих чертах разработанный к концу 90ых-началу 00гг [5]. Графические модели позволили радикально пересмотреть области применения методов машинного обучения и анализа данных за счет отказа от требования независимости скрытых переменных для разных объектов. Дискриминативная модель выборки объектов задается совместным распределением $p(T, \vec{w}|X) = p(T|X, \vec{w})p(\vec{w})$, которое, в отличие от классического случая, больше не факторизуется по отдельным объектам.

Прежде чем продолжить дальнейшее изложение приведем несколько примеров, иллюстрирующих, насколько более широкий пласт задач можно решать за счет отказа от предположения о независимости.

- Социальные сети. Пользователи социальных сетей характеризуются, как наблюдаемыми переменными (например, анкетной информацией, которую пользователь сообщил о себе в сети), так и скрытыми переменными (например, его реальными интересами, предрасположенностью к положительной реакции на адресную рекламу и т.п.). Хотя мы можем формально анализировать каждого пользователя независимо, представляется довольно очевидным, что информация о значениях скрытых переменных его друзей, может значительно расширить наши представления о данном пользователе.

- Компьютерное зрение. В задаче семантической сегментации изображений, являющейся первым этапом любой системы компьютерного зрения, требуется сопоставить каждому пикселю некоторую метку класса, соответствующую предмету, в изображение которого входит данный пискель. Очевидно, что помимо информации о данном пикселе (цвет, значения дескрипторов, интенсивность и др.) или других пикселях, важную роль играют метки соседних пикселей, т.к. неявно предполагается, что соседние пиксели чаще всего имеют одинаковые метки.

- Имитационное моделирование. При моделировании сред взаимодействующих агентов (например, транспортных потоков в городах) состояние каждого агента зависит, помимо прочего, от состояний других агентов, находящихся в пределах зоны взаимодействия. Состояние каждого агента можно рассматривать как скрытую переменную обеъекта, зависящую от скрытых переменных других объектов. Исследование таких взаимодействий играет важную роль, т.к. позволяет установить условия скачкообразных переходов от локальных взаимодействий к глобальным (т.н. фазовые переходы), например, когда из-за резкого кратковременного торможения одной машины в потоке возникает многокилометровая пробка.

- Коллаборативная фильтрация (collaborative filtering). С развитием интернет-коммерции все большую актуальность получают рекомендательные сервисы. В ситуации, когда посетитель физически не может просмотреть весь ассортимент интернет-магазина, включающий в себя десятки тысяч наименований, возникает задача формирования ограниченного списка товаров, которые его потенциально могут заинтересовать. Ясно, что кроме наблюдаемых переменных объекта (клиента), характеризующих его социально-демографический профиль и историю покупок, необходимо анализировать покупки других клиентов и близость их предпочтений к предпочтениям рассматриваемого клиента.

Характерное число объектов в выборке, с которым приходится сталкиваться в современных задачах составляет величину порядка десятков тысяч – миллионов. Основная трудность, возникающая при попытке построить вероятностную модель, содержащую взаимозависимости между скрытыми переменными объектов, заключается в невозможности задать такое распределение в общем виде. В самом деле, пусть имеется тысяча объектов, у каждого из которых есть одна скрытая переменная, принимающая два значения. Для того, чтобы задать $p(T|X, \vec{w})$ нам понадобилось бы задать $2^{1000} \approx 10^{300}$ значений вероятностей. Такое количество

[3]Автор хотел бы выразить благодарность Сергею Бартунову за помощь при написании данного раздела.

на много порядков превосходит объемы доступной памяти любого хранилища данных. При использовании графических моделей предполагается, что совместное распределение может быть представленно в виде произведения т.н. факторов, каждый из которых зависит от небольшого подмножества объектов, причем подмножества пересекаются. Благодаря этому удается смоделировать ситуации, когда скрытая компонента произвольного объекта зависит от скрытой компоненты каждого из оставшихся объектов выборки. С другой стороны, за счет факторизации, можно уменьшить требования к памяти вплоть до линейных по числу объектов, что позволяет хранить совместные распределения на сотни тысяч объектов.

3.1 Условная независимость объектов

Ключевым понятием, необходимым для понимания логики работы аппарата графических моделей, является понятие условной независимости слуайных величин. Случайные величины a и b называются незвисимыми при условии c, если верно[4]

$$p(a, b|c) = p(a|c)p(b|c).$$

Простейшим примером условно независимых величин являются: рост человека (величина a), длина его волос (величина b) и его пол (величина c). Хорошо известно, что рост обратно коррелирует с длиной волос, однако, после добавления в вероятностную модель фактора пола человека, рост и длина волос становятся независимыми величинами.

Напомним, также, два основных правила работы со случайными величинами. Рассмотрим совместную плотность n случайных величин $p(a_1, \ldots, a_n)$. Правило произведения говорит о том, что любую многомерную плотность можно представить в виде произведения одномерных условных плотностей

$$p(a_1, \ldots, a_n) = p(a_n|a_1, \ldots, a_{n-1})\times \\ \times p(a_{n-1}|a_1, \ldots, a_{n-2}) \ldots p(a_2|a_1)p(a_1).$$

Аналогичные представления можно выписать для произвольного переупорядочивания переменных.

Правило суммирования позволяет получать безусловные распределения меньшей размерности путем исключения (маргинализации) части переменных

$$p(a_1, \ldots, a_k) = \int p(a_1, \ldots, a_n)da_{k+1} \ldots da_n = \\ = \int p(a_1, \ldots, a_k|a_{k+1}, \ldots, a_n)\times \\ \times p(a_{k+1}, \ldots, a_n)da_{k+1} \ldots da_n.$$

Все операции, осуществляемые с вероятностными моделями при использовании байесовского формализма, опираются на применение этих двух правил.

3.2 Байесовские сети

Байесовские сети позволяют моделировать причинно-следственные связи между величинами. Для этого на множестве переменных $Y = (X, T, \vec{w})$ нашей вероятностной модели задается ориентированный граф, в котором ребра отражают отношения причинности. По смыслу построения в таком графе запрещены ориентированные циклы. Граф причинности задает систему факторизации совместного распределения

$$p(Y) = \prod_{i=1}^{n} p(y_i|\mathrm{pa}_i),$$

где pa_i — множество родителей i-ой вершины. Заметим, что размер каждого фактора (а именно размерность факторов служит мерой сложности распределения как на этапе его задания, так и на этапе работы с ним) определяется числом родителей вершины. Такая система факторизации значительно упрощает расчеты произвольных условных и маргинальных распределений (а именно к этому, как мы помним, сводятся задачи настройки и вывода в байесовских моделях). Так, используя факторизацию совместного распределения, заданную байесовской сетью на рис. 2 и применяя правила произведения и суммирования, легко получить выражение для, например, такого условного распределения $p(y_5|y_2)$:

$$p(y_5|y_2) = \int p(y_5|y_2, y_3)p(y_3|y_1, y_2)p(y_1)dy_1dy_3.$$

3.3 Марковские сети

Часто возникает необходимость моделировать системы случайных величин между которыми есть зависимости, но некорректно говорить о причинно-следственных связях. Примером таких величин могут быть метки соседних пикселей в задаче сегментации изображений или профили друзей в социальной сети. Для

[4]Не ограничивая общности будем полагать величины непрерывными и имеющими плотности. Индексы у функций плотностей будем опускать, считая, что они однозначно идентифицируются своим аргументом.

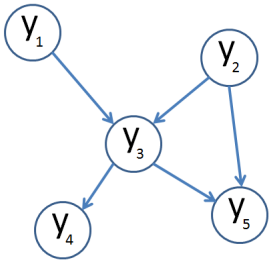


Рис. 2: Пример байесовской сети

моделирования таких зависимоетй на множестве величин задается неориентированный граф, определяющий факторизацию совместного распределения таким образом

$$p(Y) = \frac{1}{Z}\prod_{c\in\mathcal{C}}\psi_c(Y_c) = \frac{\prod_{c\in\mathcal{C}}\psi_c(Y_c)}{\sum_Y\prod_{c\in\mathcal{C}}\psi_c(Y_c)},$$

где $\psi_c(.)$ — неотрицательные функции, заданные на максимальных кликах графа. Заметим, что в отличие от байесовских сетей, множители (факторы) не имеют вероятностного смысла, поэтому необходима дополнительная нормировка произведения факторов. Легко показать, что если величины y' и y'' никогда не входят в один фактор (т.е. не соединены ребром), то они являются независимыми при условии, что все остальные величины известны. Таким образом, ребра графа определяют отношения условной независимости.

Одним из достоинств систем факторизации, задаваемых графическими моделями, наравне с удобством представления многомерных распределений, является возможность параллельной и распределенной обработки информации при подсчете условных распределений, например, с помощью интерфейса передачи сообщений (message-passing interface).

3.4 Основные задачи, возникающие в графических моделях

Аппарат графических моделей активно используется для точного или приближенного решения следующих основных задач

- Обучение с учителем $\arg\max_{\vec{w}} p(\vec{w}|X^{tr}, T^{tr})$;
- Обучение без учителя $\arg\max_{\vec{w}} p(\vec{w}|X^{tr}) = \arg\max_{\vec{w}} \sum_T p(\vec{w}, T|X^{tr})$;
- Подсчет нормировочной константы Z;
- Подсчет наиболее вероятной конфигурации скрытых переменных $\arg\max_T p(T|X, \vec{w})$
- Подсчет маргинального распределения фиксированной переменной $p(t_i|X, \vec{w})$.

Заметим, что все эти задачи сводятся к подсчету тех или иных условных распределений на неизвестные переменные при условии наблюдаемых переменных и, быть может, маргинализации по нерелевантным переменным. Можно заметить, что те же задачи возникают в классическом машинном обучении. Перенесение классических результатов на (более сложные) графические модели является одним из важнейших направлений работ в современном машинном обучении.[5]

Список литературы

[1] C. Bishop. Pattern Recognition and Machine Learning. Springer, 2006.

[2] D. Blei, A. Ng, M. Jordan. Latent Dirichlet Allocation. Journal of Machine Learning Research, 2003, 3(4-5): 993Ц1022.

[3] G. Brumfiel. "High-energy physics: Down the petabyte highway". Nature 469, 2011, pp. 282-283.

[4] G. Hinton, S. Osindero, Y. Teh. A Fast learning Algorithm for Deep Belief Nets. Neural Computation, 2006, 18(7): 1527-1554.

[5] D. Koller, N. Friedman. Probabilistic Graphical Models. MIT Press, 2009.

[6] D. MacKay. Bayesian Interpolation. Neural Computation, 1992, 4, 415-447.

[7] C. E. Rasmussen. The infnite Gaussian mixture model. In Advances in Neural Information Processing Systems, Vol. 12, 2000

[8] R. Sutton, A. Barto. Reinforcement Learning: An Introduction. MIT Press, 1998.

4 Abstract

In the paper we briefly present main active areas in modern machine learning and highlight several new paradigms which became extremely popular since the end of 90s. These paradigms make it possible to include prior domain- and task-specific knowledge in the data model. Among them are Bayesian inference, reinforcement learning, big data processing, non-parametric Bayes, deep learning and probabilistic graphical models. The latter framework is presented in more detail.

[5] Работа выполнена при поддержке гранта РФФИ 12-01-00938.

На пути к большим RDF данным

© А. Г. Марчук
ИСИ СО РАН, НГУ
Новосибирск
mag@iis.nsk.su

Аннотация

Проблематика Big Data [1] связана с накоплением и использованием больших объемов данных разных предметных областей. То же самое можно сказать о направлении Semantic Web [2], для которого «Семантическая сеть привнесёт структуру в смысловое содержание веб-страниц, тем самым создав среду, в которой программные агенты, переходя со страницы на страницу, смогут без особого труда выполнять замысловатые запросы пользователей» [2, 13]. Существенной конкретизацией Semantic Web является опора на единый механизм структуризации данных и знаний RDF, «окруженный» рядом стандартов и технологий наиболее значимыми из которых представляются OWL [3], Sparql [4], Linking Open Data [5], а также множество сформированных и опубликованных онтологий (vocabularies) [6]. Целью исследования, отраженного в докладе, являлось определение возможностей работы с большими RDF данными. Исследование поддержано грантом РФФИ 11-07-00388а, программой РАН Р-15/10, интеграционным проектом СО РАН М-48.

1 Введение

RDF позволяет накапливать и использовать данные произвольной природы, если не прямой их фиксацией, то хотя бы в виде метаинформации. Собственно последнее, похоже имелось ввиду авторами формата структурирования, чье полное название Resource Description Framework. У автора доклада накоплен значительный опыт по использованию RDF в качестве хранилищ данных для информационных систем общего назначения, архивных и музейных систем [7]. В исследовании изучались различные существующие технологии, наборы данных, онтологические построения, однако, работа не претендует на полноту обзорной части и внешние результаты рассматривались под углом зрения решаемых в ИСИ СО РАН фундаментальных и прикладных задач.

Главные вопросы, которые были поставлены: какие технологии обработки RDF лучше подходят для решаемых в коллективе прикладных задач и до каких размеров данных эти технологии будут «работать». Кроме того, интересовали массивы RDF-данных, имеющиеся в открытом пользовании, используемые для больших данных онтологии, сопоставление разных методов выборки данных, наличие в накопленных в мире массивах полезных данных, возможных к использованию в научной деятельности. Эта последняя группа вопросов в данном докладе не отражена.

2 Технологии обработки RDF данных

В рамках проектов, ведущихся в ИСИ СО РАН в течение более 10 лет накапливаются данные, в основном по предметной области исторической фактографии [8]. Эти данные собираются в так называемые кассеты [9], объединяющие в себе как собственно документный контент (фотографии, видео, аудио, сканы страниц документов, документы разных форматов), так и базу данных, в виде RDF-документов. Всего накоплено около 500 Гб. кассет и процесс накопления продолжается. Созданная технология, выполненная в виде платформенного решения Sema2012 [7], позволяет выбранный набор кассет активизировать в виде Web-приложения, сочетающего в себе элементы сервиса и разные пользовательские Web-интерфейсы, и использовать приложение для решения задач просмотра информации, поиска и редактирования. В совокупности, накопленный объем RDF данных оценивается в 2 млн. триплетов. Важным требованием прикладных построений к платформе является возможность обеспечения работы «в реальном времени», т.е. со средними задержками меньше одной секунды и с максимальными задержками не более нескольких секунд. Реализованный в платформенном решении

Труды 15-й Всероссийской научной конференции «Электронные библиотеки: перспективные методы и технологии, электронные коллекции» — RCDL-2013, Ярославль, Россия, 14-17 октября 2013 г.

«движок» RDF позволяет это делать на имеющихся данных. Однако, предполагается, что объем вовлекаемых в обработку данных может возрасти многократно, так что главная задача исследования было выяснение пределов, до которых подход будет «работать». Поскольку платформенное решение позволяет, через создание адаптера, подключать различные СУБД, все реализации разных подходов были сделаны в рамках единого средства, что в частности, упростило сравнение характеристик решений.

Для подключения конкретной СУБД требуется сформировать специфическую структуру базы данных удобную для существования в ней RDF, а также реализовать:

- создание базы данных;

- загрузку данных или в варианте потока триплетов, или в варианте потока RDF записей;

- реализовать «портретную» выборку данных о сущности по ее идентификатору;

- реализовать поиск по частичному совпадению образца и заданного поля.

Полный API платформенного решения требует также реализации нескольких дополнительных методов, связанных с редактированием данных, но это делалось не для всех решений в силу исследовательского характера работы.

Особо надо сказать о «портретной» выборки данных. Речь идет о выборке и специальном оформлении информации из окрестности заданного узла. Формируется XML-структура, соответствующая части RDF графа окрестности некоторого узла. В простом случае, такая структура содержит информацию об узле, свойствах данных для этого узла, ссылках (объектных свойствах), ведущих от этого узла и обратных ссылках (объектных свойствах), ведущих от других узлов к данному. Вид такого представления:

```
<record id="идентификатор узла" type="тип узла">
  <field prop="свойство данных 1">Значение свойства данных 1</field>
  <field prop="свойство данных 2">Значение свойства данных 2</field>
  …
  <direct prop="объектное свойство 1"><record id="ид1"/></direct>
  <direct prop="объектное свойство 2"><record id="ид2"/></direct>
  …
  <inverse prop="объектное свойство 3"><record id="ид3"/></inverse>
  < inverse prop="объектное свойство 4"><record id="ид4"/></inverse>
  …
</record>
```

В более сложном случае, эта структура формируется с использованием т.н. шаблонного дерева и представляет собой управляемую выборку окрестности узла.

Подход был обоснован и реализован в предыдущих работах [7, 8, 9, 10] и показал свою эффективность. Такой способ выборки данных отличается от рекомендаций Sparql [4], но их сферы применения не полностью пересекаются.

Все решения реализованы в обстановке Microsoft .NET, с использованием библиотеки .NET Framework, на языке C# с активным применением LINQ.

3 Использование реляционной СУБД

Движок EngineRDB был реализован вместе с платформенным решением и уже более года применяется в экспериментальных и прикладных работах. Он был реализован с использованием интерфейсов из пространства имен System.Data.Common, что позволяет легко адаптировать его к разным реляционным СУБД. Система испытывалась с MS SQL Server, MySQL, Sqlite.

Структура таблиц достаточно приближена к традиционному решению, когда организуют одну таблицу утверждений (триплетов) и колонками в таблице являются «субъект», «предикат» и «объект». Однако, сделано не столь прямолинейно. Во-первых, утверждения разбиты на объектные (object properties) и свойства данных (datatype properties). И соответственно, организованы две таблицы, также состоящие из трех столбцов, в которые отображаются утверждения. Кроме того, эти таблицы, для эффективности, реализованы в целых значениях для всех трех величин. Поэтому добавляется еще две таблицы для сущностей и данных, устанавливающие соответствие между идентификатором (для сущностей) или строкой (для данных) и их целым индексом. Индексы для сущностей «строгие», т.е. одинаковым идентификаторам обязательно сопоставляется один индекс. Такое требование нужно для того, чтобы сравнивать индексы, а не идентификаторы. Для данных, строгость не обязательна, хотя позволяет экономить память базы данных.

Также для сравнения было спроектировано и реализовано более классическое решение по использованию реляционных СУБД. База данных была спроецирована на две таблицы – таблицу объектных отношений и таблицу свойств данных.

4 Использование MongoDB

MongoDB [11] относится к классу NoSQL СУБД. Она ориентирована на обработку больших объемов данных и на эффективную их обработку, в том числе, на кластерных конфигурациях. Данные распределяются по нужному количеству таблиц. Таблицы хранят записи (документы, в терминах MongoDB) с обязательным первичным ключом. В этом смысле, MongoDB может рассматриваться как key-value хранилище. Записи представляют собой динамически сформированный набор колонок, хранящие скалярные и структурные значения. Структуризация осуществляется по принципам JSON (BSON в бинарном представлении) и легко сопрягается с большинством систем программирования, в первую очередь с JavaScript.

Колонки, хранящие скалярные данные и строки могут быть проиндексированы. Замечательным свойством MongoDB является то, что эффективно индексируются и колонки, содержащие вектора скалярных значений.

MongoDB «тяготеет» к укрупнению используемых структур данных. Поэтому была выбрана реализация RDF движка в виде одной таблицы, хранящей отдельные записи RDF-документов. Структура записи (документа) в синтаксисе C# выглядит следующим образом:

```
    public class EntityInfo
    {
        public ObjectId Id { get; set; } //
Системный идентификатор объекта
        public string LastId { get; set; } //
этот идентификатор и есть последний
        public string TypeId { get; set; } //
тип ресурса (сущности, записи) или null если не
известно
        private bool _isremoved = false;
        public bool IsRemoved { get { return
_isremoved; } set { _isremoved = value; } }
        public string[] MergedIds { get; set; }
// множество идентификаторов группы
эквивалентности, включая EntityId
        private DateTime _timestamp =
DateTime.MinValue; // последняя временная
отметка оригинала определения записи LastId
        public DateTime TimeStamp { get {
return _timestamp; } set { _timestamp = value;
} }
        // Следующее поле - для хранения полей
и объектных свойств записи
        public PredicateInfo[] RecordElements {
get; set; }
        // Следующее поле - для начальных слов
полей name
        public string[] FirstWords { get; set;
}
        // Следующее поле - набор внешних из
записи ссылок
        public string[] ExternalLinks { get;
set; }
    }
    public class PredicateInfo
    {
        public bool IsObjectProperty { get;
set; }
        public string Predicate { get; set; }
        public string Value { get; set; } //
это либо идентификатор объекта, либо строка
данных
        public string Lang { get; set; }
    }
```

Запись выглядит несколько усложненной, но это из-за того, что реализовывался вариант RDF, позволяющий осуществлять редактирование данных. В частности для этого используется поле IsRemoved истинность которого означает, что запись уничтожена. Также к редактированию относится поле MergedIds, хранящая цепочку эквивалентных идентификаторов и поле TimeStamp, предназначенное для фиксации времени изменения записи. Индексируется поле LastId для доступа к записи по ее идентификатору. Причем семантика идентификатора та, что он «последний» в наборе эквивалентных идентификаторов. Индексируются также два векторных поля: FirstWords и ExternalLinks. В вектор FirstWords помещаются первые слова вариантов, в том числе и языковых вариантов, имени объекта и используются для поиска по частичному совпадению с образцом. Вектор ExternalLinks имеет принципиальное значение в подходе. В него помещаются все (последние) идентификаторы объектных свойств, хранимых в RecordElements. Эта индексация позволяет экономно получать доступ ко всем записям, ссылающимся на заданный идентификатор.

5 Использование Open Link Virtuoso

Virtuoso фирмы Open Link Software [12], является коммерческой СУБД в которой, в частности, реализован RDF-движок с осуществлением запросов через Sparql. Фирма ставит перед собой амбициозные цели и хорошо известна в мире разработчиков, использующих RDF. К счастью, кроме дорогого коммерческого варианта СУБД, имеется свободно распространяемый вариант, который был использован для исследовательских целей.

Отображение RDF на Virtuoso и построение требуемого для платформы API – задача достаточно простая. Единственная существенная сложность оказалась в реализации поискового запроса. Рекомендуемый Sparql способ поиска по неполному совпадению через фильтрацию с использованием регулярного выражения, хоть и работает, но работает не быстро даже на небольших данных. Имеющееся в Virtuoso дополнительное средство имеет свои особенности применения.

6 База данных в оперативной памяти – RGraphEngine

Для целей сравнения, был адаптирован к платформенному решению наш «старый» движок RGraph. Это – прямая реализация RDF в виде графа в оперативной памяти. Такой вариант понадобился в исследовании для получения сопоставительных данных как «предельный» по скорости обработки и формирования выходных структур. Кроме того, такой движок вполне годится для проектов

небольшого и среднего размера. В частности, в ИСИ уже три года эксплуатируется публичный интерфейс фотоархива СО РАН. Архив пополняется новыми данными, так что внутренняя база данных, построенная на RGraph перезагружается ежедневно. Кроме того, собственно загрузка выполняется достаточно быстро. Например, база данных, содержащая 700 тыс. триплетов загружается за 6 секунд. Это быстрее, чем «холодный» запуск сложного сайта.

7 Использование Apache Cassandra

В качестве еще одной СУБД, которая вошла в систему экспериментов по реализации RDF, явилась Apache Cassandra, созданная и развиваемая компанией DataStax [14] совместно с вовлеченным в проект сообществом программистов. Это решение уже считается «ветераном» среди NoSQL СУБД и использовано в большом количестве коммерческих и некоммерческих систем. К сожалению, нам не удалось подобрать адаптер к Кассандре, позволяющий стабильно работать из C#/.NET. Тем не менее, кое-какие выводы относительно возможности ее использования для реализации RDF-хранилища удалось сделать.

Поскольку Кассандра считается хорошим Key-Value-хранилищем, схема реализации была основана на множестве RDF-записей, идентифицированных текстовыми ключами (использовались URI сущностей):

```
CREATE TABLE DB.Records (id text PRIMARY KEY,
type text, xbody text, invlist set<text>);
```

Для простоты, сама запись сохранялась в виде текста её XML-представления (поле xbody), а для эффективности, были добавлены в виде списка идентификаторы сущностей, ссылающихся на данную (поле invlist). Такая схема не достаточно удобна для расширения графа при его редактировании, но для целей, поставленных в исследовании, является вполне адекватной. Кроме построения информационного портрета, была реализована достаточно быстрая схема поиска сущностей заданных классов по образцу, хотя Кассандра и не приспособлена для подобных действий. Нестабильность адаптера СУБД не позволила загрузить данные большие, чем 2 млн. триплетов.

8 Специализированная СУБД FSRDF

Поскольку решения, базирующиеся на свободно распространяемых СУБД оказались неспособными загружать 1 млрд. триплетов, решено было попробовать написать свою упрощенную и специализированную СУБД, основанную на специальных файловых структурах. Такая СУБД была создана и погружена в платформенное решение. В итоге, удалось загрузить тест, имеющий миллиард триплетов и временные характеристики доступа к данным оказались весьма неплохими.

Решение основывается на трех файлах. Первый файл – собственно хранилище триплетов, организован как набор строк формата tsv (Tab Separated Values), где субъект, предикат и объект разделены традиционным символом-разделителем. Строки неупорядочены и записаны в порядке поступления при сканировании RDF-документов. Строка доступна через длинное целое – позицию в файле. Второй файл – бинарная файловая структура, хранящая наборы специально организованной информации для каждого объектного узла. Эта информация имеет ссылки (длинные целые) на триплеты, хранящиеся в первом файле. Один набор группирует множество триплетов в элементах которого идентификатор узла является либо субъектом, либо объектом высказывания.

Третий файл – достаточно просто организованный словарь, устанавливающий соответствие между идентификаторами объектных узлов и наборами второго файла.

Опыт и эксперименты показали, что даже на больших данных, RDF-хранилище работает довольно быстро и тратит мало оперативной памяти.

9 Методика проведения экспериментов и некоторые результаты

Проведенное исследование состояло в изучении возможностей реализации загрузки и работы графа RDF для разных подходов и СУБД и второе – изменение скоростей и других характеристик загрузки и обработки.

По первой части, можно констатировать, что почти для каждой СУБД, пришлось формировать адекватный подход к эффективному решению задачи. Причем, кроме логических и программистских задач, приходилось преодолевать значительное количество задач инженерных, связанных с особенностями развертывания, настройки и функционирования той или иной системы.

Для измерения характеристик были сформированы тестовые наборы данных по следующей схеме. Сначала была собрана из ряда имеющихся в ИСИ СО РАН источников тестовая база данных tm (two millions of triples), представляющая реальные, увязанные между собой данные в количестве около 2 млн. триплетов. Потом, путем мультиплицирования с изменением идентификаторов и имен объектов, были сделаны базы данных tm3 и tm10, представляющие 6 и 20 млн. триплетов соответственно. Для базы данных tm, в результате «ручного» просмотра в имеющемся интерфейсе, через навигационные действия и поисковые запросы, была создана последовательность запуска построения информационных портретов и поиска по образцу.

Была произведена трассировка на данном наборе запросов каждого из набора данных. Результаты таких трассировок, фиксировались в лог-файле. Потом были сведены воедино и обработаны.

Измерялись и вычислялись следующие характеристики: время формирования (загрузки) базы данных, время выполнения отдельных запросов на построение информационные портреты и объем получившихся информационных портретов, время выполнения поисковых запросов и количество получившихся вариантов.

Тестирование осуществлялось на рабочей станции с процессором Intel Core i7, 3.5 GHz, 16 Gb RAM, 64-разрядная операционная система Windows-7, .Net Framework 4.0. Результаты расчетов приведены в таблицах 1-3.

	tm	tm3	tm10
mssql	185	638	3700
mongo	26	70	359
virtuoso	402	800	2729
rgraph	17		
cassandra	433		
fsrdf	26	57	204

Таб. 1 Время загрузки наборов тестовых данных, сек.

	tm	tm3	tm10
mssql	360	1290	5733
mongo	272	816	2636
virtuoso	248	644	2425
rgraph	147		
cassandra	14.6		
fsrdf	1.9	1.6	2.3

Таб. 2 Среднее время поиска по текстовому образцу, мс.

	tm	tm3	tm10
mssql	92	380	391
mongo	1223	3190	10112
virtuoso	290	267	331
rgraph	3.4		
cassandra	209		
fsrdf	37	42	45

Таб. 3 Среднее время построения информационного портрета, мс.

10 Заключение

В докладе рассматриваются различные схемы реализации работы с RDF-данными. Показано, что традиционные и специализированные схемы отображения базы данных на СУБД позволяют решать основные задачи визуализации, поиска и навигации для данных среднего размера. Произведен сопоставительный анализ характеристик использования разных СУБД. Некоторые решения прошли опытную эксплуатацию и используются в реальных проектах.

Полученные результаты нельзя рассматривать как точную оценку возможностей различных СУБД для решения задач работы с RDF-информацией. Это зависит от слишком многих факторов, включая: примененные схемы реализации RDF-графа и вспомогательные структуры, улучшающие работу с этой информацией, инженерные конфигурационные настройки, качество адаптеров, операционная система и ее разрядность, количество ядер, объем имеющейся в наличии оперативной памяти, скорость работы дисковой памяти и др.

В рамках исследования был поставлен еще один вопрос: какие предельные объемы данных можно обрабатывать в рамках предложенных подходов и схем реализации? В качестве тестового материала был выбран набор данных freebase-rdf-2013-02-10-00-00.nt2, содержащий около 1 млрд. триплетов. При использовании усечения данного набора какой-то длины можно было проследить поведение той или иной СУБД. При росте объемов вводимых данных, увеличивались и трудности работы с конкретными СУБД. В большинстве случаев, все «ломалось» при объемах в десятки миллионов триплетов. Как правило, объемы захватываемой оперативной памяти становились слишком большими.

Все это привело к созданию простой специализированной СУБД FSRDF, ориентированной на работу с графами RDF и реализованную средствами файловой системы и прямого доступа к файлам. Оказалось, что современная стыковка файлов и виртуальной памяти настолько оптимизирована, что простое, но специализированное средство работы с RDF-данными может быть заметно более эффективным, чем настройка на эти цели универсальной СУБД. Приведенные цифры это показывают. Задача загрузки 1 млрд. триплетов была решена. Причем скорость построения упрощенного информационного портрета (множество исходящих из узла дуг и множество входящих) оказалась приемлемой и для работы в традиционном стиле Web-интерфейса.

Литература

[1] White, Tom (10 May 2012). Hadoop: The Definitive Guide. O'Reilly Media. p. 3. ISBN 978-1-4493-3877-0.

[2] Berners-Lee Tim, Hendler James, Lassila Ora, The Semantic Web. In Scientific American, volume 284(5), pages 34-43, 2001.

[3] Web Ontology Language (OWL). — http://www.w3.org/2004/OWL

[4] SPARQL Query Language for RDF. - http://www.w3.org/TR/2008/REC-rdf-sparql-query-20080115/

[5] Sören Auer, Lorenz Bühmann, Christian Dirschl, Orri Erling, Michael Hausenblas, Robert Isele, Jens Lehmann, Michael Martin, Pablo N. Mendes, Bert Van Nuffelen, Claus Stadler, Sebastian Tramp, and Hugh Williams. Managing the Life-Cycle of Linked Data with the LOD2 Stack.. In

Philippe Cudré-Mauroux, Jeff Heflin, Evren Sirin, Tania Tudorache, Jérôme Euzenat, Manfred Hauswirth, Josiane Xavier Parreira, Jim Hendler, Guus Schreiber, Abraham Bernstein, and Eva Blomqvist (Eds.), International Semantic Web Conference 2, (7650):1-16, Springer, 2012.

[6] Vocabularies, - http://www.w3.org/standards/semanticweb/ontology

[7] А.Г.Марчук, П.А.Марчук Платформа реализации электронных архивов данных и документов // Электронные библиотеки: перспективные методы и технологии, электронные коллекции: Труды XIV Всероссийской научной конференции RCDL'2012. Переславль-Залесский, Россия, 15-18 октября 2012 г. – г. Переславль-Залесский: изд-во «Университет города Переславля», 2012, С. 332-338.

[8] Marchuk A.G. Methods and Technologies of Digital Historical Factography // Knowledge Processing and Data Analysis. First International Conference, KONT 2007, Novosibirsk, Russia, September 14-16, 2007,and First International Conference, KPP 2007, Darmstadt, Germany, September 28-30, 2007. Revised Selected Papers. Series: Lecture Notes in Computer Science, Vol. 6581, Subseries: Lecture Notes in Artificial Intelligence, Wolff, K.E.; Palchunov, D.E.; Zagoruiko, N.G.; Andelfinger, U. (Eds.), 2011, ISBN 978-3-642-22139-2, pp 217-231

[9] Марчук А.Г., Марчук П.А. Особенности построения цифровых библиотек со связанным контентом // Электронные библиотеки: перспективные методы и технологии, электронные коллекции: Сб.трудов / XII Всеросс. научн. Конф. RCDL'2010, Казань, Россия 13–17 октября 2010 г. — Казань: Казан. ун-т, 2010. — С. 19–23.

[10] Ануреев И.С. и др. Модели и методы построения информационных систем, основанных на формальных, логических и лингвистических подходах // отв. ред. А.Г. Марчук ; Рос. акад. наук, Сиб. отд-ние, Ин-т систем информатики им. А.П. Ершова. – Новосибирск: Изд-во СО РАН, 2009. – 330 с.

[11] Data Modeling Considerations for MongoDB Applications, - http://docs.mongodb.org/manual/core/data-modeling/

[12] Open Link Software, - http://www.openlinksw.com/Donald E. Knuth, Tracy L. Larrabee, and Paul M. Roberts. Mathematical Writing. Mathematical Association of America, 1989. http://www-cs-faculty.stanford.edu/knuth/klr.html

[13] Тим Бернерс-Ли, Джеймс Хендлер и Ора Лассила Семантическая Сеть – перевод [2] Евгения Золина, 2004. http://ezolin.pisem.net/logic/semantic_web_rus.html/

[14] Сайт компании DataStax // http://www.datastax.com

On the Way to Big RDF Data

Alexander G. Marchuk

The goal of research work, reflected in the report was to determine the possibilities of working with large RDF data.

Some principles and details of different DBMS use for RDF data are presented. Various approaches were used to create efficient solutions for work with such graphs.

It was shown that traditional and specialized schemes of the RDF-graph on the database can solve the basic problems of visualization, search and navigation data of the medium size. A comparative analysis of the characteristics of the use of several database systems was done. Some solutions are currently used in real projects.

Semi-automatic Data Extraction from Tables

© Nikita Astrakhantsev
ISPRAS
Moscow
astrakhantsev@ispras.ru

© Denis Turdakov
ISPRAS
Moscow
turdakov@ispras.ru

© Natalia Vassilieva
HP Labs
Saint-Petersburg
vassilieva@hp.com

Abstract

This paper describes a novel approach to automate extraction of useful information from tables and to record the knowledge procured in a structured data repository. The approach is based on modeling a behavior of an expert, who collects tabular data and maps them to a predefined relational schema. Experimental results demonstrate that the proposed approach predicts expert decisions with high accuracy and thus significantly minimizes the time required of an expert for data aggregation.

1 Introduction

Tables are widely used in scientific, financial and other analytical documents to concisely communicate information to human readers. A table usually contains information about objects of the same type. These objects can be easily perceived by the reader through content and relations between different table elements (cells, rows, etc.), because he/she is experienced in table reading and is able to take in both semantic and structural information. Sometimes two tables with exactly the same structure are interpreted completely different just because of slight difference in the external context or because the content of some cells differs.

However, it is hard to automate table analysis and information is extracted mainly by hand by interested parties. A common scenario used by experts is manual cell by cell extraction of data from a table into a relational database, and then using OLAP or other techniques to generate reports and perform analytics over these data. Efforts required for manual extraction are considerable, while the time of experts is always costly.

There is no end-to-end solution for automatic information extraction from arbitrary tables. And as it appears to us, construction of a fully automatic instrument is hardly feasible. It might be possible to parse automatically the structure of any table, but semantic interpretation of tabular data requires the knowledge of a domain expert. Embley et al. [1] suggest The Periodic Table of the Elements as an example of the table requiring semantic knowledge for interpretation; the authors of this survey also state that it is easy to contrive other examples "that are challenging even from a human perspective".

We present a semi-automatic approach that tracks actions of a domain expert, when he/she begins to process a table (map cell data to a relational scheme), derives regularities/patterns of expert behavior, and applies them to the rest of the table in order to predict further mappings.

This paper is organized as follows: In Section 2 we give an overview of related works. Section 3 discusses the table format used in our work. Section 4 describes the general architecture of our prototype, while Section 5 describes it in details. Section 6 presents our experimental evaluation. We conclude in Section 7.

2 Related work

Silva et. al [11] survey about 50 works devoted to tables processing in details. Authors outline several table-related tasks and corresponding table representation models, which serve as input and output for these tasks. The span of tasks goes from location of a table in a document to semantic interpretation of the information contained in the table. There are more works focused on the basic low-level table related tasks than on the more knowledge based ones. A deep interpretation of the table is almost always requires context specific knowledge. Existing solutions for extracting information are very domain specific and designed for particular table types.

Zanibbi et. al [15] present the table recognition literature in terms of the interaction of table models, observations, transformations, and inferences. Most of described methods are fully automatic and consider the task of table processing in isolation from further usage of the extracted information.

Embley et al. [2] extract data from XML tables and map them to a given target database schema with 96/85 precision and 93/91 recall (depending on a domain — car advertisement and cell-phone correspondingly). However, their approach requires a hand-crafted ontology, which is costly and, more important, cannot always be in place due to high specificity of certain documents.

More recent work of Embley and Krishnamoorthy [3] transforms CSV or HTML tables

Proceedings of the 15th All-Russian Conference "Digital Libraries: Advanced Methods and Technologies, Digital Collections" — RCDL-2013, Yaroslavl, Russia, October 14-18 2013.

into a canonical representation in order to obtain a target representation, one of which is Relational table. A canonical table representation is based on Header Paths, a purely syntactic technique that relates headers and data cells. Further transformation into a target representation is performed by specially defined relational algebra. In contrast to the previous approach, this one does not use any semantic knowledge; also it demands a user to point out the top-left data cell in cases where proposed heuristics cannot define such a cell automatically. Precision of correct Header Paths construction is about 74%. In [8] the authors evolve this approach by using interactive tool VeriClick [9], 'a macro-enabled spreadsheet interface that provides ground-truthing, confirmation, correction, and verification functions for CSV tables'. However, this tool is used only to locate so-called 'critical cells': corner cells that allows to distinguish header and data regions in a table.

Vasudevan et. al [14] address a closely related task: to automate data extraction from financial reports presented in PDF documents with many tables. The proposed approach shows good results (95.7% precision and 78.4% recall), but it requires a quality review stage and, since it is based on domain knowledge heuristics, the range of its application is severely limited.

Gatterbauer et al. [4] extract information from web tables by using two-dimensional visual model provided by web browsers instead of tree-based (HTML) representation. Fumarola et al. [5] combine knowledge about the visual structure of the Web page and the HTML markup for web lists extraction. Their tool is applicable for web tables, too; moreover, the authors evaluate accuracy on the same dataset as Gatterbauer and report very good quality (more than 99% precision and recall on table records). But this dataset is domain-independent, and target format of table analysis is more general than ours; therefore, it cannot be directly compared with our method.

Looking at semi-automatic tools, Google Refine[1] should be noted. It is "a power tool for working with messy data, cleaning it up, transforming it from one format into another, extending it with web services, and linking it to databases like Freebase". However, Google Refine is mostly for data cleansing and is not capable of performing information extraction and interpretation. It also cannot track user decisions and anticipate them.

Similarly, Microsoft Excel[2] can perform such transformations using formulas and macros, but it demands user to define these formulas explicitly.

Praedea[3] is another semi-automatic tool that uses predefined text-mining models (chosen by user) for extracting required data from unstructured documents and mapping them to database or XML scheme. It is not capable to process a table in case there is no suitable predefined model for it.

Thus, except for the highly specialized programs like VeriClick and commercial tools like Google Refine or MS Excel, we are not aware of any research on interactive information extraction from tables; and the authors of VeriClick confirm this observation [5]. It also should be noted that most works try to convert a table into some more usable format, but not to extract needed parts of information from a table.

3 Table format

Like other basic notions, 'table' has a lot of different definitions.

Peterman et al. [10] suggest the following intuitive definition: "tables have a regular repetitive structure along one axis so that the data type is determined either by the horizontal or vertical indices." The definition given by Lopresti et al. [7] consists of similar items:

1. 2-D cell assembly for presenting information;
2. Regular, repetitive structure along at least one axis;
3. Datatype determined by either horizontal or vertical index.

Tijerino et al. [12] uses standard definition of a relational table.

In this work we consider a table to be a set of cells with some text content. In other words, it is the only property of a table that is used explicitly; other properties like repetitive structure or the same datatype in a column or a row are considered by our method implicitly.

Our prototype takes HTML tables as an input; therefore we use terminology from HTML in order to describe cell properties; for example, colspan as a relative width of a table cell. However, our method does not depend on any specific properties of HTML format.

We use classical spreadsheet addressing for cells. For example, A2 of Table 1 is an empty cell in the first column and the second row with rowspan equal 2.

Table 1: Example of a source table with spreadsheet coordinates

	A	**B**	**C**	**D**	**E**
1	Secret budget				
2		FY 2009		FY 2010	
3		Oper.	Capital	Oper.	Capital
4	HP	10	20	30	40
5	Oracle	50	60	10	20
6	Samsung	12	34	56	78

4 General architecture

Assume that a user processes a table step-by-step. During each *step* the user selects multiple cells from the table and maps them to a record in a relational database. Let us define *user feedback* as information about mappings from a single user step. See Table 2 for an example of the initial user feedback provided by him/her while processing Table 1.

1 http://code.google.com/p/google-refine

2 http://office.microsoft.com/en-us/excel/

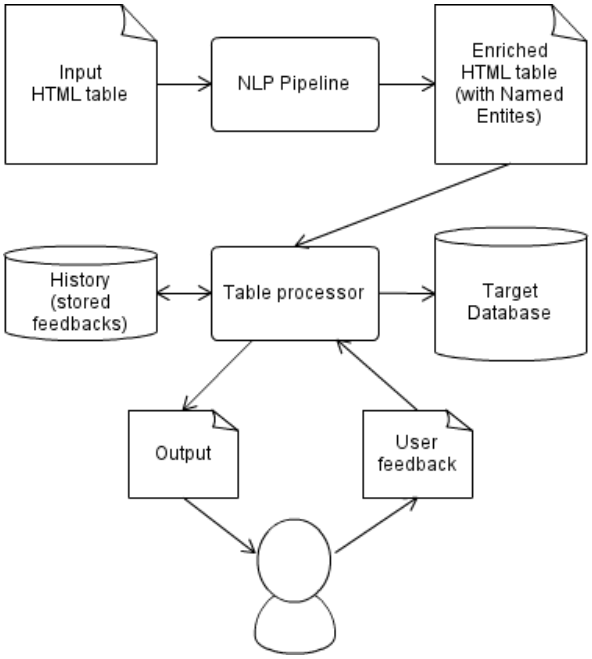
3 http://www.praedea.com

Table 2: User feedback example: two first columns represent data about the target relational scheme; two last ones represent table cell data

Relation name	Attribute name	Cell position	Value
Report	Company	A4	HP
Report	Operating	B4	10
Report	Financial Year	B2	2009

Figure 1 shows the general architecture of the prototype.

Figure 1: General architecture



The prototype takes as input an HTML table and enriches it by the predefined *Natural Language Processing pipeline*; currently we add only information about Named Entity types.

Enriched table is further processed by the main part of the prototype - *Table processor* on the diagram. It generates output, which has the same format as the user feedback, i.e. information about mappings that assumed to be obtained by the user. This output is reviewed by the user and he/she returns feedback. The prototype stores information from the user feedback into the *Target database* and uses this feedback in order to generate the next output. *History* means a local storage of the previous feedbacks, including ones from other tables that were already processed by the user. Information about previous actions is also used by the prototype for the output generation.

5 Shift approach

Our approach is based on two observations: (a) each table is a layout structure for storing similar objects; and (b) closely located tables (e.g. from the same document or Web-site) usually share similar structure and context. A user processes similar objects from the same table sequentially by repeating similar actions; and this repetition allows the system to learn and anticipate the forthcoming actions. The key idea of the approach is to consider the *shift* from one user step to the next one. More precisely, a shift consists of row and column offsets (possibly zero) for each cell represented in a pair of sequential user feedbacks. Since any table contains several objects of the same type and structure, we try to recognize it by tracking sequential user steps.

Table 3 shows an example of a shift from the first feedback (underlined words) to the second one (bolded words). Note that some cells remain fixed, e.g. *B2*.

Table 3: Shift example: from (HP, 10, 2009) to (Oracle, 50, 2009)

	A	B	C	D	E
1	Secret budget				
2		FY **2009**		FY 2010	
3		Oper.	Capital	Oper.	Capital
4	HP ↓	10 ↓	20	30	40
5	**Oracle**	**50**	60	10	20
6	Samsung	12	34	56	78

Shift approach is used in different ways depending on the number of the available user feedbacks. In section 5.1 we describe a regular phase of our algorithm: when at least two user feedbacks are available and we can explicitly compute shifts. Section 5.2 presents the phase when we have only one user feedback and have to guess a correct shift. Section 5.3 considers the phase when user starts to process a table and we have to predict his/her actions based on the previously processed tables.

5.1 Regular phase

After two user steps we can construct a shift by computing row and column offsets explicitly for every cell. For example, after two steps of processing Table 1 the following shift is constructed: the first two cells are shifted down by one cell and the third cell stays at the same position.

The constructed shift is further applied to cells of the last user step in order to get new cells and create a new row in a relational database with the same relation/attribute information and new values taken from new cells.

As it was shown in the example above, sometimes the value returned by the user is not just a text content of the cell, but its derivative: compare "*FY 2009"* in Table 1 with just "*2009"* in the feedback, Table 2. Solution of this problem is based on the observation that shifted cells share similar structure of their content. We store the way of producing the value from the cell in form of a regular expression. More precisely, we have prepared a set of patterns, which should be tested to match cell content against the value returned by the user within his/her feedback. Currently there are two types of patterns: (a) taking a substring of original cell content, and (b) replacing a substring of original cell content with another predefined string. Patterns of both types contain corresponding regular expressions to be applied to cell content.

Table 4: Example of cell value patterns

Pattern	Description	Cell value	Feedback value
.*::::0	Takes the value as is	HP	HP
(.*)\s(.*)::::1::::2	Splits by space and tries each part	Financial year	year
(\d+)::::1	Takes only digits from the content	146%	146
thousand::::000::::ReplaceAll	Replaces all words *thousand* by 3 zeros	45 thousand	45000

Patterns of the first type also contain matching group numbers to be taken into account; patterns of the second type contain a replacement string and a key word *ReplaceAll* in order to distinguish this type. See Table 4 for the examples of cell value patterns. Sequence of colons (::::) serves as a delimiter.

These patterns work as follows. Assume, have a cell with content *"FY 2009"* and the corresponding feedback value is *"2009"*. We try to apply the first pattern (.*) to cell content, which means that we simply take the whole content as is. The result does not match with the feedback value. Then we try the next pattern ((.*)\s(.*)::::1::::2). The result matches with the feedback value and we store this pattern for the corresponding cells of the shift.

If there are more than two steps produced by the user, then we construct candidate shifts for all paired combinations of user feedbacks, assess them, and choose the best shift in order to apply it to cells of the last user feedback. For example, if there are 3 feedbacks, we construct 3 candidate shifts: 1-to-2, 1-to-3, and 2-to-3; all 3 candidate shifts are assessed independently, and the best shift is then applied to the last, 3rd feedback.

Shift assessment is performed by computing linear combination of the following 4 features:

Average shift length: normalized sum of lengths of cells offsets. For the example shift, there are 2 offsets having length 2 and 1 offset having length 0, so the feature value is 2/3.

Motivation: people tend to process the table sequentially, taking closest cells if possible. In other words, shorter shifts are preferable by users.

We also tried different weights for different directions, e.g. bigger weight for the right offset than for the down one, because top-down processing seem to be more common, but experiments show that different weights do not introduce any positive effect.

Cells offsets consistency: normalized number of most common cells offsets. For the shift in the example, there are 2 types of offsets: 2 down-by-1 offsets and 1 remain-fixed offset, so the feature value is 2/3 again.

Motivation: a shift usually contains similar offsets of cells, e.g. all 4 cells are shifted down by 1 more often, then when 2 cells are shifted down by 1 and other 2 cells are shifted down by 2 or right by any number.

Average text similarity: normalized value of string similarity metrics computed over corresponding (shifted) cells contents.

Motivation: when we shift one cell to another, both of them must contain values of the same attribute, and different values of the same attributes are usually similar text strings.

To compute such similarity we tried several string metrics taken from SimMetrics[4], the best results have been obtained using Levenstein distance [6]. In addition, we modified string metric as follows:

1. All digits are considered to be equal characters, because, as it is written above, we want to capture strings of the same attribute, or data type, and difference between numeric strings tells nothing about difference between attributes. Note that we do not consider all numbers to be equal, because much difference in orders of magnitude can indirectly indicate different attributes.
2. If one of the strings is empty, we use pre-defined value (0.5): sometimes cell values are missed, for example, it can mean that the previous value or some default value should be taken instead. Unmodified metrics return zero similarity for such cases, but cell value absence does not necessarily indicate the difference in attributes.
3. If both strings are long (more than 3 words), we use pre-defined value (0.8): again, difference in long texts does not reflect difference in attributes.

Named entity type consistency: predefined value for 3 cases depending on named entity types of cells contents:

1. Named entity types are equal – value is 1;
2. Named entity types are unequal – value is 0;
3. Named entity types are both undefined – value is 0.5;

Motivation is the same as in the previous feature, but here we utilize information about named entity types in order to check attribute consistency. We use a conditional random field (CRF) model with a combination of different popular features applied in supervised named entity recognition [13]. There are 6 supported named entity types: Acronym, Date, Location, Numeric, Organization, Person.

Coefficients for the linear combination are chosen experimentally to maximize the accuracy: we test all possible values from 0 to 1 with step 0.2 so that their sum equals to 1. We found the best accuracy to be obtained with 0, 0.4, 0.2, 0.4 correspondingly; the further granulation does not change the result.

5.2 One User Feedback Phase

Given one user feedback we need a shift to apply it to this user feedback as it is done in Regular phase. There are two ways: take a most appropriate shift from the previously processed tables or construct a shift from

[4] SimMetrics is a Similarity Metric Library provided by UK Sheffield University
http://sourceforge.net/projects/simmetrics/

scratch. To find the most appropriate shift we check all stored shifts for applicability, and then assess similarity of all applicable shifts modified in accordance with the current table. To assess the modified shift we use the linear combination as in section 5.1, but coefficients are re-estimated (0.2, 0, 0.6, 0.2).

This way is very similar to No user feedback phase, see Section 5.3 for details. In short words, modification means that we replace contents of cells in the shift by the contents of the corresponding cells of the current table. For example, if there is a stored shift (from some previously processed table) with just one cell offset – C4-to-C5 with contents "USA"-to-"Russia" – then we modify the shift so that now contents of the cell offset is "20"-to-"60": we take contents of C4 and C5 cells of Table 1. Note that if the shift contains cell offset like B1-to-B2 with ordinary colspan and rowspan (all equal to 1), then such the shift is inapplicable for the Table 1, because B1 and B2 cells of this table have different colspans.

If there is no suitable shift, i.e. similarity of the best shift does not reach a predefined threshold, then we construct a new one. For this purpose, we iterate over combinations of all possible offsets of row and column for each cell. To limit combinatorial explosion we do not consider offsets above predefined thresholds: 5 for rows and 3 for columns. Each offset combination is actually a shift that can be assessed as it is described above; coefficients are left the same.

5.3 No User Feedback Phase

When no user feedback is available for the current table, we can use information about previously processed tables. We store all user feedbacks during table processing for the case if some of them have a structure (set of cell positions) appropriate for a new table. To choose the most appropriate user feedback we first check all of them for applicability, i.e. current table must have cells in all cell positions of the user feedback, and these cells must have the same characteristics — particularly, colspan and rowspan. Then all applicable user feedbacks are assessed by constructing and assessing special "fake" shifts from the stored feedback to the feedback obtained by applying the stored feedback structure to current table. Assume we have a feedback shown in Table 5 from the already processed Table 6 and we just begin to process Table 1.

Table 5: Example of stored feedback

Relation	Attribute	Cell	Value	Cell pattern
Report	Company	A4	Lenovo	.*::::0
Report	Capital	B4	105	.*::::0
Report	Market share	C4	33	(\d+)::::1

Then we take a structure of the stored feedback, that is actually a cell position and a cell pattern, and apply it to the considered Table 1, see Table 7.

Table 6: Source of stored feedback

	A	B	C
1	Public budget		
2	Company	Capital	Share of market
3	Dell	100	27%
4	**Lenovo**	**105**	**33%**

Table 7: Applied stored feedback

Relation	Attribute	Cell	Value	Cell pattern
Report	Company	A4	HP	.*::::0
Report	Capital	B4	10	.*::::0
Report	Share of market	C4	20	(\d+)::::1

After that we construct a shift from the stored feedback (Table 5) to the obtained feedback (Table 7) and assess it in the way described in section 3.1 with re-estimated coefficients (0.2, 0, 0.4, 0.4). Such assessment allows us to choose the most similar table among all already processed ones.

Note that the stored feedback from the 3rd row, but not 4th, is not applicable to Table 1, because A3 cells have different rowspans.

6 Evaluation

We compiled a set of 30 tables containing financial reports and a set of more than 150 corresponding user feedbacks[5]. We use the following test metrics: accuracy, precision and recall. Accuracy shows the fraction of outputs that fully match the user feedback: if at least one value in the tool output is wrong, the whole answer is considered to be wrong. Precision and recall characterize the number of correct cell mappings. Obviously, precision and recall can be much higher than accuracy, because many answers are partially correct.

Table 8 shows the results for the regular phase; the 2nd and the 3rd rows show efficiency of shift constructing module and shift choosing module respectively. We consider the shift constructor to work correctly if there is a right output (maybe not the chosen one); for the shift chooser we count only those outputs when there is a correct shift constructed and thereby the shift chooser had a chance to choose it.

Table 8: Regular phase results

Total accuracy	74%
Shift constructor accuracy	78%
Shift chooser accuracy	94%
Precision	84%
Recall	80%

Table 9 shows the results for the first 2 phases. Easy to see that these results depend on feedbacks from previously processed tables, because each user feedback is stored during the processing and affects the following outputs. However, the order of tables inside the

[5] Dataset's URL : http://modis.ispras.ru/datasets/td.zip

document may be valuable; therefore, we shuffle order of blocks containing tables from the same document.

It is worth mentioning that the results of the first two phases also depend on stored shifts (feedbacks, for the first phase) and at the begging of the work, without any stored shift, we face the problem of cold start. That is why we add tests with prepared set of 5 simple stored feedbacks and shifts from other tables.

In addition, we also run tests when all feedbacks and shifts from the same 30 test tables are stored and used for the testing. Of course, it is not an absolutely fair test, because there are stored feedbacks and shifts with strictly the same text content, but these tests may help to understand if the mistakes are caused by the problem of missing stored shifts or the incorrect choice of the stored shift to be applied.

Table 9: Results for No user feedback and One user feedback phases

Phase	Number	Accuracy	Precision	Recall
No user feedback	0	4%	11%	7%
	5	4%	8%	6%
	all	16%	28%	25%
One user feedback	0	43%	91%	90%
	5	48%	87%	86%
	all	86%	87%	86%

Table 10 shows results for each module for one user feedback phase. The similarity estimator chooses the most similar shift among all stored ones. The similarity threshold is estimated perfectly: if the similarity estimator chooses an appropriate shift among the stored ones, then we always choose it and never try to construct our one instead. The shift constructor works not bad, it means that our algorithm constructs most of possible shifts, but the shift chooser makes a lot of mistakes.

Table 10: Accuracy of modules for one user feedback phase

Evaluated module	0 stored shifts	5 stored shifts	All stored shifts
Similarity	100%	100%	100%
Similarity	100%	100%	100%
Shift chooser	25%	20%	0%
Shift	80%	77%	33%

Some mistakes in the regular phase could be explained by the following: sometimes a user goes from the top to the bottom of the left part of the table (e.g. takes all values from the first and the second columns) and then he/she repeats the similar actions for the right part (takes all values from the first and the third columns). Our tool cannot predict a correct shift at the moment when the user switches to the right part, that is why results for shift constructor are low.

7 Conclusions and Future Work

In this paper, we focused on the task of semi-automatic data extraction from tables and mapping them to relational scheme; we introduce novel approach that tracks user decisions to predict forthcoming ones; we evaluated it on our test data.

Shift approach provides general scheme for semi-automatic table processing, but many problems are out of scope of this research. One of them is extraction of value from table cell. In most cases it is sufficient to take text substrings (see cell B2 in Table 1), but sometimes certain table cell is actually a set of different attribute values that should be processed by more complex methods. For instance, a price list of a hardware store often contains cells like the following:

HP "Pavilion dm4-2102er" QJ453EA (Core i5 2430M-2.40GHz, 6144MB, HD6470M, WebCam)

Another direction of further research is related to target scheme: currently we copy information about relation and attribute for shifted cells, but methods that are more sophisticated can consider semantics of both table content and relational scheme.

References

[1] D. W. Embley, M. Hurst, D. Lopresti, G. Nagy. Table-processing paradigms: a research survey. International Journal of Document Analysis and Recognition (IJDAR), 8(2-3), p. 66-86, 2006.

[2] D. W. Embley, C. Tao, S. W. Liddle. Automating the Extraction of Data from HTML Tables with Unknown Structure. Knowledge Engineering, 54 (1), p. 3-28, 2005.

[3] D. W. Embley, M. Krishnamoorthy. Factoring Web Tables. Proceedings of 24th International Conference on Industrial Engineering and Other Applications of Applied Intelligent Systems, p. 253-263, 2011.

[4] F. Fumarola, T. Weninger, R. Barber, D. Malerba, and J. Han. HyLiEn: a hybrid approach to general list extraction on the web. Proceedings of the 20th international conference companion on World wide web, pp. 35–36, 2011.

[5] W. Gatterbauer, P. Bohunsky, M. Herzog, B. Krüpl, and B. Pollak. Towards domain-independent information extraction from web tables. Proceedings of the 16th international conference on World Wide Web, pp. 71–80, 2007.

[6] V. I. Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. Soviet Physics Doklady, 10: p. 707–10, 1966.

[7] D. Lopresti, G. Nagy. A tabular survey of automated table processing. Graphics Recognition Recent Advances, p. 93-120, 2000.

[8] G. Nagy, S. Seth, D. Jin, D. W. Embley, S. Machado, and M. Krishnamoorthy. Data extraction from web tables: The devil is in the details. Document Analysis and Recognition (ICDAR), pp. 242–246, 2011.

[9] G. Nagy and M. Tamhankar. VeriClick: an efficient tool for table format verification. IS&T/SPIE Electronic Imaging, p. 82970M–82970M, 2012.

[10] C. Peterman, C.H. Chang, H. Alam. A system for table under-standing. Proceedings of the Symposium on Document Im-age Understanding Technology (SDIUT'97), p. 55–62, 1997.

[11] A. C. Silva, A. M. Jorge, L. Torg. Design of an end-to-end method to extract information from tables. International Journal on Document Analysis and Recognition (8), No. 2-3, p. 144-171, 2006.

[12] Y. A. Tijerino, D. W. Embley, D. W. Lonsdale, Y. Ding, G. Nagy. Towards ontology generation from tables. World Wide Web 8, no. 3, 261-285, 2005.

[13] M. Tkachenko, A. Simanovsky. Named entity recognition: Exploring features. Proceedings of KONVENS 2012, p. 118-127, 2012.

[14] B. G. Vasudevan, A.G. Parvathy, A. Kumar, R. Balakrishnan. Automated Knowledge-based Information Extraction from Financial Reports. Knowledge Engineering and Management, 7(5), p. 61-68, 2009.

[15] R. Zanibbi, D. Blostein, J. R. Cordy. A survey of table recognition: Models, observations, transformations, and inferences. International Journal of Document Analysis and Recognition, 7(1), Springer, Heidelberg, p. 1–16, 2004.

Новый источник данных для наукометрических исследований

© М.Р. Когаловский

Институт проблем рынка РАН

kogalov@cemi.rssi.ru

© С.И. Паринов

Центральный экономико-математический институт РАН

Москва

sparinov@gmail.com

Аннотация

Обсуждается подход к созданию новых источников данных для наукометрических исследований, основанный на технологии семантического структурирования контента научных электронных библиотек, который был предложен авторами в ранее опубликованных работах. Рассматриваются основные принципы предлагаемого подхода, а также результаты его реализации в среде системы Соционет. В качестве нового наукометрического источника данных рассматриваются коллекции семантических связей между различными информационными объектами из контента системы. Семантические связи классифицируются с помощью заданной таксономии, представляемой в виде набора контролируемых словарей. Такие источники данных позволяют проводить более многоаспектные по сравнению с традиционными наукометрические исследования представленного в системе корпуса научных знаний. По мнению авторов, создание глобальных автономных репозиториев семантических связей, интегрирующих коллекции связей из различных научных электронных библиотек в анализируемых областях знаний, является перспективным направлением развития наукометрии. Работа поддержана РФФИ, проект 12-07-00518-а и РГНФ, проект 11-02-12026-в.

1 Введение

В настоящее время важную роль в оценке деятельности исследовательских организаций и ученых стали играть наукометрические измерения, результаты которых существенным образом влияют на формирование их научного авторитета, а также и на распределение финансовых ресурсов, предназначенных для поддержки исследовательских программ и отдельных проектов.

Сложившаяся практика оценки научной результативности ученых и научных периодических изданий базируется, в частности, на использовании *индексов цитирования*. Нужно отметить, что термин *индекс цитирования* имеет два значения. Это, одной стороны, библиографическая информационная система, предназначенная для наукометрических измерений, в которой регистрируются связи цитирования, определяемые пристатейными списками источников, между научными публикациями из охватываемого ими корпуса периодических изданий. С другой стороны – это имеющий несколько разновидностей наукометрический показатель, представляющий собой статистическую оценку количества цитирований научных публикации какого-либо автора (традиционное простое количество цитирований, индекс Хирша и др.).

Более значимыми считаются статьи, опубликованные в журналах с более высоким импакт-фактором, а также индексируемые признанными международными системами - индексами цитирования SCOPUS, Web of Science, Web of Knowledge, Springer и др. Активно ведется работа по формированию отечественной системы РИНЦ, которая пока, к сожалению, не располагает достаточно представительным корпусом публикаций для наукометрических измерений, но охватываемое ею научное публикационное пространство интенсивно развивается.

Несомненно, количественные характеристики внимания представителей научного сообщества к той или иной публикации, а также интегральные характеристики внимания к публикациям конкретного автора или периодического издания, представляемые индексами цитирования, являются важными показателями качества научной деятельности. Однако одной из слабых сторон традиционной практики наукометрических измерений является их базирование на связях цитирования с неопределенной явным образом семантикой. Мы называем такие связи *«немыми»* [6], т.к. они сами по себе не несут какой-либо информации, характеризующей, например, мнение автора цитирующей работы о цитируемом источнике или цель цитирования. В связи с

Труды 15-й Всероссийской научной конференции «Электронные библиотеки: перспективные методы и технологии, электронные коллекции» — RCDL-2013, Ярославль, Россия, 14-17 октября 2013 г.

этим возможны такие парадоксальные ситуации, когда высоким количеством цитирований обладает статья, содержащая грубые ошибки и/или принципиальные заблуждения, касающиеся обсуждаемой проблемы, и в связи с этим вызывающая активный отклик научного сообщества.

Избежать указанных ситуаций и вместе с тем существенно обогатить информационную основу наукометрических исследований позволяет использование в системах - индексах цитирования семантических связей между научными публикациями. Мы называем *семантическими* связи с явным образом декларированной семантикой. Такие связи могут создаваться, поддерживаться и использоваться для наукометрических исследований и в научных электронных библиотеках, а также в других научных информационных системах, обладающих необходимыми для этого механизмами.

Актуальность такого подхода отмечалась, в частности, еще в работе [5]. Такие более информативные источники для наукометрических исследований позволяют генерировать не только традиционные наукометрические показатели, но и, что более важно, получать новые многоаспектные более утонченные количественные и качественные характеристики отдельных публикаций или групп публикаций, авторов публикаций, а также научных организаций в целом. Исследуя структуру семантических связей, можно выявлять формирующиеся направления в науке, исследовать историю их развития, получать другие полезные результаты.

В последние годы для явного описания семантики связей разработан ряд специальных онтологий, учитывающих не только различные классы связей цитирования, но и связей другой природы, таких как «автор-публикация», «организация-автор», «публикация-фрагмент публикации» (аннотация, оглавление, предисловие, библиография и т.п.), связи между ее версиями, вариантами представления и др. Одним из ранних проектов в этой области является комплекс онтологий SPAR (The Semantic Publishing and Referencing Ontologies) [28, 32]. Следует упомянуть также онтологию SWAN (Semantic Web Applications in Neuromedicine) [26], созданную специалистами в области нейромедицины, рекомендацию SKOS (Simple Knowledge Organization System) [33] консорциума W3C. Работы в этой области ведутся в европейской ассоциации euroCRIS в рамках рабочей группы CERIF (The Common European Research Information Format) [12], а также другими научными коллективами.

На основе указанных онтологий реализуется ряд исследовательских проектов, в которых предусматривается явная спецификация семантики связей. Известны, например, проекты Nanopub.org [16] и SiteULike.org [29]. Авторами данной работы с использованием разработанных онтологий конструктивизирован и развит подход, представленный в уже упоминавшейся статье [5]. При этом учитываются не только связи цитирования и обеспечиваются новые функциональные возможности для наукометрических исследований по сравнению с известными проектами, прежде всего, за счет иного способа представления семантических связей [6-8, 24, 25]. Описания семантических связей представляются в данном проекте как самостоятельные информационные объекты, а не встраиваются в метаданные связанных публикаций. Особенности принятых авторами решений, некоторые новые результаты в развитии и реализации ранее опубликованного подхода, а также функции необходимых для этого механизмов системы Соционет [10, 23] обсуждаются в следующих разделах статьи.

В разделе 2 обсуждаются возможные способы представления семантических связей (их описаний) в контенте научной электронной библиотеки. Аргументируются достоинства их представления как самостоятельных информационных объектов. Раздел 3 посвящен рассмотрению методов создания и описания семантических связей. В разделе 4 обсуждаются вопросы использования онтологий связей и основанных на них контролируемых словарей для описания семантики связей, кратко характеризуются опубликованные проекты онтологий связей, результаты которых используются в данной работе. В разделе 5 рассматривается организация совокупности определенных в электронной библиотеке семантических связей как нового источника данных для наукометрических исследований. Наконец, в разделе 6 описаны результаты реализации предлагаемого подхода в среде системы Соционет. В заключении обсуждаются элементы новизны обсуждаемого подхода и кратко представлены предполагаемые направления развития данного проекта.

2 Представление семантических связей в электронных библиотеках

В научной электронной библиотеке для каждой представленной в ней публикации и объектов других типов, например, профилей авторов или организаций, существует некоторый информационный объект (описатель), содержащий совокупность определяющих его свойства метаданных. Если необходимо учредить и явным образом отразить существование связи между данным и некоторым другим объектом, например, связи цитирования, то такая связь также явным образом должна быть описана. Явная спецификация семантических связей между информационными объектами, составляющими контент научной электронной библиотеки, обогащает представленный в ней корпус научных знаний. Описатель связи (ее метаданные) является ее представителем в информационной системе, как и описатели связываемых объектов.

В научных электронных библиотеках представляют интерес прежде всего *бинарные ориентированные семантические связи*. Далее будут рассматриваться именно такие связи. Информационный объект, из которого исходит связь, будем называть *исходным объектом связи*, а на который связь направлена – ее *целевым объектом*.

В известных авторам проектах [16, 29] для описания факта существования такой связи вводятся дополнительные метаданные в описателе исходного информационного объекта описываемой связи. Для семантической связи значения таких атрибутов характеризуют, в частности, не только идентификатор ее целевого объекта, но и ее семантику, а также другие свойства этой связи. При таком *«встроенном»* способе представления связей между существующими в библиотеке информационными объектами создавать связи в системе может только лицо, обладающее полномочиями обновления описателей этих объектов. Обычно такими полномочиями наделяются администраторы информационных ресурсов, но не пользователи электронной библиотеки.

Другой, *«автономный»*, способ представления связей между информационными объектами, который предложен и развивается в работах авторов данной статьи [6-8], предусматривает поддержку описателей связей в системе как *самостоятельных информационных объектов*. Такой автономный описатель связи содержит метаданные, описывающие уникальные идентификаторы связываемых информационных объектов, семантику связи, идентификатор учредившего связь лица, дату ее создания и некоторые другие необходимые характеристики. При использовании такого способа представления связей учреждать их в системе могут не только администраторы информационных ресурсов, но и обычные зарегистрированные пользователи системы. Таким образом, открывается возможность для нового вида научной деятельности, позволяющая ученым в онлайновом режиме высказывать свои мнения о представленных в электронной библиотеке публикациях, оценивать описываемые в них результаты, характеризовать семантику научных отношений между различными публикациями. Можно, например, создавая связь, указать, что исходная ее публикация использует методы или данные, описанные в целевой публикации этой связи, или что в исходной публикации обнаружен плагиат результатов, представленных в целевой публикации связи и т.п.

Представление семантических связей в научной электронной библиотеке как самостоятельных информационных объектов обеспечивает ряд преимуществ по сравнению со «встроенным» способом их представления. Действительно, на основе контента электронной библиотеки может быть создан автономный по отношению к ее контенту *репозиторий семантических связей*, который может служить новым источником данных для наукометрии. Такие репозитории, созданные в различных библиотеках, могут интегрироваться, и благодаря этому формировать глобальные репозитории семантических связей для той или иной области знаний. На этом пути возможно создание достаточно представительных источников данных для наукометрии в различных областях науки. Если электронные библиотеки построены на основе технологии открытых архивов OAI (Open Archives Initiative) [21], интеграция созданных в их среде репозиториев семантических связей легко обеспечивается точно так же, как и интеграция контентов самих электронных библиотек.

Кроме того, как уже отмечалось, автономное представление связей позволяет поддерживать виртуальную социальную среду *для совместной деятельности* пользователей-ученых в качестве экспертов, добровольно высказывающих мнения и оценки, касающиеся представленных в электронной библиотеке публикаций и других научных информационных ресурсов. Эта деятельность дополняет традиционную практику анонимного рецензирования печатных изданий, а открытость высказанных оценок для научного сообщества позволяет ответным образом реагировать на них, способствует более высокой ответственности и объективности их авторов. Возможности для такой деятельности обеспечиваются, например, сервисом F1000Research [17] проекта FACULTYof1000. Однако подход, обсуждаемый в этой статье, предусматривает поддержку также структурированных данных в форме семантических связей, которые позволяют учитывать высказанные экспертами оценки в наукометрических исследованиях.

В обсуждаемом здесь подходе используется именно «автономный» способ представления семантических связей.

3 Методы описания и создания семантических связей

Будем далее предполагать, таким образом, что семантические связи между информационными объектами контента электронной библиотеки представляются как «автономные» информационные объекты. При этом возможны три метода описания и создания таких объектов.

Первый из них основан на компетенции ученых-экспертов и предусматривает «ручное» описание и создание ими связей при поддержке имеющегося в библиотеке механизма со специальным пользовательским интерфейсом. Создавая связь, эксперт указывает в ее описателе идентификаторы исходного и целевого объектов связи. Используя контролируемые словари (таксономии), основанные на поддерживаемой онтологии связей, он специфицирует класс, к которому относится данная связь. В описателе связи также указываются идентифицирующие данные эксперта, дата создания связи, при необходимости и комментарий [24, 25].

Второй метод применим в случае, когда исходный информационный объект связи является текстовой публикацией. При этом требуется предварительная его обработка. Она заключается в том, что эксперт просматривает не пристатейный список литературы, а текст публикации, и выявляет встречающиеся в нем библиографические ссылки на использованные источники. Анализируя контекст каждой ссылки и используя контролируемые словари семантических связей, он осуществляет ее *онтологическое аннотирование* [3]. Эту работу, конечно же, может выполнить и сам автор данного текста.

Такой размеченный текст далее обрабатывается специальным механизмом системы, и для каждой аннотированной ссылки генерируется описатель соответствующей связи, помещаемый в некоторую коллекцию связей, которая представлена в библиотеке. При использовании данного метода, как и рассмотренного выше, созданные связи обладают авторством, которое приписывается эксперту или автору публикации, выполнившему указанную обработку ее текста. Проблема онтологического аннотирования библиографических ссылок в научных публикациях подробно рассматривается в работе [3].

Наконец, третий метод может рассматриваться как автоматизированный вариант второго. В последние годы активно развиваются исследования, посвященные анализу эмоциональной окраски, тональности текста [2, 14, 15, 20, 22]. Это направление исследований называется в зарубежной литературе *Sentiment analysis* (или *Opinion mining*). Задача такого анализа заключается в определении мнения автора анализируемого текста относительно предмета обсуждения. Если применить методы *Sentiment analysis* к окрестности внутритекстовой ссылки, т.е. к ее контексту, как это делается в работе [34], то можно выявить мнение автора о цитируемой работе. Используя такой метод, можно тем самым автоматизировать ту работу по онтологическому аннотированию ссылок, которую во втором подходе выполняет эксперт. Далее, аналогично предыдущему, на основе таких аннотаций можно автоматически генерировать описатели связей. Их автором является, естественно, автор текста, содержащего аннотированные ссылки. Конечно же, возможности такой «диагностики» семантики ссылок ограничиваются лишь классами оценочных ссылок.

В настоящее время в системе Соционет реализован только первый - экспертный *«ручной»* метод создания семантических связей как самостоятельных информационных объектов.

4 Описание семантики связей

Основой для реализации пользовательских инструментов, позволяющих создавать семантические связи между информационными объектами и оперировать ими, стали разработанные в последние годы онтологии семантических связей. В частности, это онтологии связей между объектами научной сферы деятельности (публикациями различных типов, темами исследований, учеными, исследовательскими организациями и др.). В этих онтологиях определяются иерархии классов связей, соответствующие различного рода отношениям, в которых могут состоять информационные объекты – участники связей. Рассмотрим их кратко.

Среди основательно проработанных проектов следует назвать, прежде всего, модульный комплекс онтологий SPAR (*the Semantic Publishing and Referencing Ontologies*) [28, 32], созданный сотрудниками Оксфордского и Болонского университетов. SPAR включает восемь независимых онтологий, позволяющих описывать семантику библиографических объектов, а также их отношений. Эти онтологии описаны средствами языков OWL2 DL и RDF консорциума W3C. Первые четыре из них (FaBiO, CiTO, BiRO and C4O) позволяют описывать библиографические объекты, библиографические записи и источники в списках литературы публикаций, а также связи цитирования, контексты цитирования и их связи с релевантными разделами цитируемых публикаций. Четыре остальных онтологии (DoCO, PRO, PSO and PWO) могут использоваться для описания семантики компонентов документов, ролей и состояний публикаций, потоков работ в издательских процессах.

Другой заслуживающий внимания проект в рассматриваемой области – модульный комплекс онтологий SWAN (*Semantic Web Applications in Neuromedicine*) [26], разработанный в Главном госпитале Массачусетса и Медицинской школе Гарварда. Авторы характеризуют его назначение как обеспечение в Семантическом Вебе комфортной среды - *социально-технической экосистемы*, которая позволяет создавать и сохранять семантический контекст научных коммуникаций, обеспечивает доступ к нему, его интеграцию, а также обмен неструктурированной и слабоструктурированной цифровой научной информацией. Онтологии комплекса описаны в его спецификации средствами языка описания онтологий уровня OWL DL.

Примерно в то же время консорциумом W3C была принята рекомендация SKOS (*Simple Knowledge Organization System*) [33], предназначенная для поддержки систем организации знаний - тезаурусов, схем классификации, таксономий и рубрикаторов (*Subject Heading Systems*) - в среде Семантического Веба. SKOS определяет концептуальную схему, называемую *общей моделью данных,* служащую для совместного использования и связывания систем организации знаний средствами Веба. Благодаря унифицированной концептуальной схеме SKOS упрощается интеграция существующих систем организации знаний в Семантический Веб.

Следует отметить, что принцип модульности организации таких сложных комплексных онтологий, как SPAR и SWAN, облегчает их повторное использование. В некоторых приложениях нет необходимости использовать полную онтологию. Тогда могут использоваться отдельные ее модули. Облегчается также интеграция онтологий. Так, в комплексе SPAR используются элементы SWAN, а в SWAN используется SKOS.

Существенный вклад в рассматриваемую область вносит также европейская научная общественная организация euroCRIS, инициировавшая и развивающая проект CERIF. Одним из главных результатов этого проекта является создание унифицированной концептуальной схемы, называемой в материалах проекта *полной моделью данных* (*Full Data Model*) [11]. Эта модель рассматривается как единая основа создания информационных систем (*Current Research Information Systems*, CRIS) для поддержки научно-организационной деятельности в разных странах и

различных научных организациях. Благодаря стандартизации концептуальной схемы обеспечивается интероперабельность таких систем. В последнее время в рамках проекта CERIF была предложена спецификация стандартизованной семантики полной модели данных [12], онтология, определяющая систему терминов для обозначения сущностей этой модели и отношений между ними.

Рассмотренные онтологии могут быть использованы для создания онтологии семантических связей информационных объектов конкретной научной электронной библиотеки, адекватной характеру представленных в ней информационных ресурсов и ее функциональных механизмов, в частности, ее наукометрического аппарата. Структура семантических связей, определенная на контенте электронной библиотеки, является *многослойной* [6-8]. Каждый ее слой соответствует некоторому классу связей, определенному в используемой онтологии. Семантическая структура контента библиотеки может использоваться как источник данных для наукометрических исследований, а также служить основой семантической навигации в ее контенте. Для практического использования на основе используемой онтологии для конкретной электронной библиотеки может быть сформирована *таксономия семантических связей*, представленная в виде набора *контролируемых словарей* имен классов семантических связей.

В электронной библиотеке может поддерживаться несколько таксономий связей, основанных на разных онтологиях, точно так же как для рубрикации информационных объектов может использоваться несколько рубрикаторов научно-технической информации. Так, в системе Соционет поддерживаются рубрикатор ГРНТИ [1] и классификатор JEL (Journal of Economic Literature Classification System) [18]. На основе их рубрик по запросам генерируется наукометрическая статистика [9].

При спецификации пользовательских запросов в электронной библиотеке по умолчанию может использоваться основная встроенная в систему таксономия с ее набором управляемых словарей. В противном случае используемая таксономия должна специфицироваться в запросе.

5 Описания семантических связей - источник данных для наукометрии

Организованные совокупности семантических связей между информационными объектами контента научной электронной библиотеки, созданные описанными выше методами и средствами, служат более информативным источником данных для наукометрических исследований по сравнению с традиционной наукометрией, которая базируется на множестве «немых» ссылок цитирования.

Описания семантических связей могут использоваться наукометрическими сервисами библиотеки, продуцирующими разнообразные показатели как на основе предоставляемой описанием связей информации о том, какие информационные объекты связаны, так и информации о семантике существующих связей.

Семантические связи, организованные в виде коллекций информационных объектов специального типа, могут составлять подмножество контента электронной библиотеки. Они могут быть также организованы в виде самостоятельного информационного ресурса, сосуществующего с ее контентом.

Представляется заманчивой интеграция таких информационных ресурсов, созданных и поддерживаемых в различных научных электронных библиотеках, и формирование открытых (свободно доступных) глобальных репозиториев семантических связей для наукометрических исследований [24, 25] в различных областях знаний. Такие источники данных более полно представляют корпус научных знаний рассматриваемой области науки. На их основе можно формировать адекватную наукометрическую статистику и другие характеристики состояния данной области знаний.

Интеграция ресурсов семантических связей различных научных электронных библиотек может быть осуществлена известными методами виртуальной или материализованной интеграции данных из множества информационных источников [4, 19, 36]. Интеграция относительно легко реализуется при условии базирования библиотек-источников на технологии Инициативы открытых архивов OAI и протоколе OAI-PMH [21, 35].

В обоих случаях может возникнуть проблема неоднородности онтологий семантических связей, используемых в библиотеках–источниках. Для ее решения возможны два подхода. При первом подходе таксономия интегрированного репозитория строится как объединение таксономий, применяемых в электронных библиотеках–источниках. В такой ситуации интегрированный источник представляет собой фактически федерацию семантически неоднородных наборов связей. Статистические запросы при этом будут учитывать только связи, семантика которых определена указанной в запросе таксономией. При втором подходе осуществляется семантическая интеграция всех охватываемых наборов связей, представленных в библиотеках-источниках. Для интегрированного репозитория создается некоторая общая таксономия таким образом, чтобы было возможно определить отображения таксономий библиотек-источников в эту общую таксономию. Анализ и обработка контента интегрированного глобального репозитория связей осуществляется при этом на основе общей таксономии связей.

Помимо возможности обработки глобального репозитория семантических связей средствами созданных для него наукометрических сервисов, целесообразно обеспечить интерфейс прикладного программирования (API) для доступа к его контенту. Таким образом, обеспечивается возможность разработки разнообразных приложений, оперирующих контентом глобального репозитория семантических связей.

6 Реализация предлагаемого подхода в среде системы Соционет

В рассматриваемом проекте в качестве среды реализации предлагаемого подхода используется система Соционет. В настоящее время реализованы механизмы системы, обеспечивающие описание, создание, хранение, модификацию, удаление и просмотр семантических связей, формирование коллекций связей. Кроме того, реализованы основные средства создания и поддержки контролируемых словарей семантических связей. Реализован также ряд статистических сервисов, обрабатывающих семантику связей и генерирующих новые наукометрические показатели. Рассмотрим несколько подробнее основные особенности уже реализованных средств и сервисов.

Создание и организация связей в системе. Совокупности связей поддерживаются в системе в форме коллекций информационных объектов специального типа *linkage*. Наряду с таким «автономным» вариантом представления семантических связей поддерживается и встроенный вариант, который, однако, реализован лишь для обеспечения полноты возможностей при дальнейшем развитии системы. Пока реализованы только средства для «ручного» создания связей экспертом-пользователем системы (см. разд. 3). К коллекциям связей применимы все имеющиеся в системе функциональные возможности управления коллекциями любого типа данных.

Как уже указывалось, в Соционет поддерживаются бинарные ориентированные семантические связи. Информационными объектами–участниками связей могут быть объекты различных видов (электронные монографии, статьи в периодике, диссертации и авторефераты диссертаций, классификаторы, авторы публикаций, исследовательские или образовательные учреждения и др.). Среди объектов-участников связей в Соционет могут быть и библиографические описания публикаций, возможно, дополненные аннотациями. Эти объекты в системе относятся к типу *artifact.* Коллекциями таких объектов могут представляться тематические библиографии.

Информационные объекты-участники связей могут быть внутренними для системы (содержатся в ее контенте) или внешними. Внешние объекты не содержатся в контенте системы. Они доступны в Вебе по их адресу (URL). Допустимость внешних информационных объектов, а также публикаций, представленных их библиографическими описаниями (объекты типа *artifact*), в качестве участников связей позволяет охватить семантическими связями все доступное ученым цифровое научное информационное пространство.

Метаданные коллекций связей являются составной частью репозитория метаданных системы Соционет, основанной на технологии *открытых архивов OAI.* Поэтому этот фрагмент репозитория может быть представлен в виде самостоятельного репозитория метаданных, обеспечивая организацию совокупности коллекций связей в системе как самостоятельного открытого архива. Этот архив при необходимости может интегрироваться с аналогичными архивами связей других электронных библиотек на основе протокола OAI-PMH. Таким образом могут формироваться глобальные репозитории семантических связей, более представительные полигоны для наукометрических исследований. Если интегрируемые архивы семантических связей не основаны на единой стандартной для них таксономии связей, то, как уже отмечалось, возникнет необходимость решения проблемы преодоления неоднородности таксономий.

Связи в системе Соционет могут создавать зарегистрированные в ней пользователи. Такие пользователи имеют в системе свои профили и, тем самым, они идентифицируемы как авторы создаваемых ими семантических связей. Имеющиеся в Соционет средства мониторинга состояния связей в необходимых случаях (например, при появлении в системе семантически противоречивых созданных разными авторами связей между информационными объектами некоторой пары) могут оповещать их сообщениями по электронной почте, адрес которой должен указываться в профиле пользователя, формируемом в процессе его регистрации в системе.

При создании связи в Соционет формируется ее описатель (метаданные связи), включающий следующие метаданные: уникальный идентификатор связи, тип и идентификатор ее исходного объекта, тип и идентификатор целевого объекта (или URI для внешнего целевого объекта), описание ее семантики - класс таксономии, к которому она относится, дату ее создания, уникальный идентификатор, идентификатор автора связи и при необходимости его комментарий [24, 25].

В зависимости от типов информационных объектов-участников создаваемой связи она может принадлежать только к какому-либо классу таксономии, соответствующему этой паре типов ее объектов-участников. Однако для заданной пары объектов может быть создано несколько связей. Разные авторы связей могут создать несколько связей одного класса. Один и тот же автор не может создать несколько связей одного класса для заданной пары объектов, но имеет возможность создать несколько связей разных классов.

В некоторых ситуациях при создании связи включаются системные механизмы, которые автоматически генерируют другие связи. Это имеет место, например, в случае, когда целевой объект создаваемой связи, являющийся текстом статьи, одновременно состоит в связях с другими объектами, которые характеризуются связями как копии этой статьи. Тогда, если создаваемая экспертом связь является оценочной для такого объекта, то автоматически будут генерироваться аналогичные оценочные связи того же исходного объекта с объектами, представляющими другие копии.

Автоматическая генерация явно описанных связей может осуществляться также как побочный эффект при создании связи, порождающей транзитив-

ные отношения между информационными объектами в системе (см. пример ниже).

Спецификация семантики связей. В инструментарии создания и использования семантических связей в системе Соционет для спецификации семантики создаваемых связей использована *гибридная онтология*, являющаяся расширением объединения некоторых фрагментов онтологий CiTO [27, 30], DoCo [31], SWAN [26], SKOS [33] и CERIF [11, 12]. На ее основе создана двухуровневая таксономия классов семантических связей, представленная в виде набора контролируемых словарей, каждый из которых соответствует одному из классов верхнего уровня иерархии классов таксономии, а значения в словарях соответствуют подклассам этих классов.

Действующая версия системы Соционет поддерживает набор контролируемых словарей семантических связей, включающий словари: связей научного вывода; связей использования; связей между компонентами научного произведения; а также между его версиями или копиями; связей научных оценок (оценочных связей); иерархических и ассоциативных связей между публикациями; связей объектов вида «персона-персона», «персона-организация», «персона-публикация». Более подробно используемая в Соционет таксономия семантических связей и представляющие ее контролируемые словари описаны в работе [8].

Важно здесь отметить, что созданная для Соционет таксономия семантических связей позволяет классифицировать связи не только между объектами-научными текстами. Это обстоятельство имеет существенное значение, поскольку, участниками связей в системе могут быть как научные публикации и их компоненты, так и наборы научных данных, организации и их сотрудники – авторы информационных объектов и пользователи системы, а также информационные объекты других типов, представленные в контенте системы.

В системе Соционет реализован механизм расширения таксономии семантических связей пользователями путем дополнения классов к существующим словарям и создания новых контролируемых словарей в режиме модерирования администратором системы.

Сервисы системы для операций с семантическими связями. Эти сервисы выполняют довольно большой набор функций, позволяющих получать разнообразную информацию о структуре связей в библиотеке. Прежде всего, это статистическая информация. Ряд таких сервисов уже реализован в системе. Реализованы возможности семантической навигации по слою структуры связей, соответствующему заданному классу связей. Предстоит также реализация ряда других сервисов, осуществляющих исследование топологии графа связей и позволяющих на этой основе получать ряд полезных результатов, характеризующих состояние и генезис представленного в системе корпуса научных знаний.

Поскольку, как уже отмечалось, в системе могут одновременно поддерживаться несколько таксономий семантических связей, при обращении к таким сервисам необходимо указывать на основе какой из них или какого конкретного контролируемого словаря должна проводиться обработка пользовательского запроса. Например, если пользователя интересует статистика оценочных связей, то в запросе должен быть указан словарь или класс какого-либо словаря научных оценок из той или иной поддерживаемой в системе альтернативной таксономии.

Характер генерируемой по запросам статистической информации может быть весьма разнообразным. Например, можно запросить для конкретного информационного объекта количество входящих или исходящих из него связей (т.е. связей, в которых данный объект участвует как целевой или, соответственно, как исходный), относящихся к некоторому классу верхнего уровня, т.е. к некоторому контролируемому словарю таксономии в целом, или к его подклассу, т.е. к одному из значений в заданном словаре. Содержательная интерпретация полученной статистики, естественно, зависит от заданного словаря. Здесь возможно большое количество вариантов: сколько имеется позитивных или негативных мнений о данной статье, в каком количестве работ используются предложенные в ней методы или содержащиеся в ней научные данные, сколько обнаружено случаев плагиата данной статьи и т.д.

При формировании статистики для некоторых классов таксономии могут учитываться транзитивные связи этих классов. Например, существует связь между статьей A - исходным объектом связи и статьей B – целевым объектом этой связи, указывающая, что в B предлагается более широкое обсуждение проблемы, которой посвящена A. Кроме того, существует связь между статьей B как исходным объектом и статьей C – целевым объектом этой связи, которая также указывает, что в C предлагается более широкий взгляд на предмет обсуждения по сравнению с B. Тогда фактически существует явно не описанная транзитивная связь A с C с той же семантикой. Это обстоятельство должно учитываться при формировании статистики статей, в которых более широко обсуждается проблема, рассматриваемая в статье A.

Статистические запросы могут быть обобщены на все множество информационных объектов заданного типа или на все связи класса верхнего уровня таксономии (на указанный словарь в целом). Например, может запрашиваться статистика мнений обо всей совокупности монографий из какой-либо коллекции или общее количество работ, в которых высказано позитивное мнение о данной работе.

Поскольку в описателях связей указывается дата их создания, возможны запросы статистики, относящейся к заданным промежуткам времени или, что более сложно, динамической статистики (временных рядов некоторых статистических показателей).

Другая группа запросов позволяет получить перечень конкретных информационных объектов биб-

лиотеки, связанных с заданным объектом как исходным или целевым в связях заданных классов. Содержательные интерпретации получаемого при этом результата также различаются в зависимости от используемого словаря или конкретного его класса связей. Запросы этого вида позволяют, например, выяснить, на результаты каких публикаций опирается некоторая конкретная работа или, наоборот, в каких публикациях получены результаты, основанные на данной работе. При этом можно учитывать не только непосредственные, но и транзитивные связи. В критерии отбора интересующих пользователя связей может также использоваться идентификатор автора связей.

Используя цепочки связей вида «автор-публикация» + «публикация-публикация» можно получить количество публикаций, в которых выражено негативное или позитивное отношение к публикациям данного автора, либо список таких публикаций. С использованием более длинных цепочек вида «организация-сотрудник (автор)» + «автор-публикация» + «публикация-публикация» можно получить аналогичные сведения для интересующей организации, агрегированные по всем ее сотрудникам – авторам представленных в системе работ.

Наконец, важную группу запросов составляют запросы операций над полным графом связей. Здесь можно решать множество различных задач, связанных как с анализом топологии графа и вычленением подграфов с заданными свойствами, так и с визуализацией подграфов. Так, можно вычленить и визуализировать из многослойной структуры семантических связей слой, соответствующий связям некоторого класса, например, указывающего на использование одной публикации из контента системы как основополагающей для других публикаций. Можно также запросить подграф, образованный связями, относящимися к классу развития научных результатов, и указать, что ему должна принадлежать некоторая имеющаяся в библиотеке общепризнанная основополагающая публикация в некоторой области исследований. Полученный подграф будет характеризовать логику развития данной области науки, конечно, если в контенте системы достаточно основательно представлены публикации, относящиеся к этой области. Еще одним примером операций над полным графом связей библиотеки является операция вычленения из него подграфа связей, установленных данным пользователем, возможно, с указанием в запросе также конкретного класса связей.

7 Заключение и направления дальнейшей работы

Мы рассмотрели подход, позволяющий использовать коллекции или репозитории семантических связей информационных объектов контента научных электронных библиотек, расширенного внешними информационными объектами, как источник данных для новых нетрадиционных многоаспектных наукометрических исследований. Прототип необходимых для этого программных средств реализован в системе Соционет, и продолжается работа по развитию его функциональных возможностей в рассмотренных направлениях.

Некоторые идеи рассмотренного подхода опубликованы другими авторами практически одновременно с нашими ранними работами в этой области. Однако рассмотренный здесь подход, по мнению авторов, обладает существенной степенью новизны по отношению к известным работам.

Одно из главных отличий заключается в представлении семантических связей как самостоятельных информационных объектов специального типа данных *linkage*, которые хранятся автономно от объектов-участников связей, а не встраиваются в их описатели. Отчуждение описателей связей от описателей связываемых объектов позволяет строить коллекции связей, формировать на их основе репозитории связей, которые могут интегрироваться с другими репозиториями. Благодаря этому становится возможным формирование представительных глобальных репозиториев семантических связей для различных областей знаний. Может быть обеспечено также *повторное использование* коллекций семантических связей как информационного ресурса для наукометрических исследований, более содержательных по сравнению с традиционной наукометрией, основанной на «немых» связях цитирования.

Важно также, что благодаря этому связи может создавать лицо, не являющееся создателем информационных объектов и/или метаданных (описателей) информационных объектов-участников связей. Это могут делать любые пользователи, зарегистрированные в системе, тем самым развивая семантическую структуру ее контента. Если описание связи встроено в метаданные информационного объекта (обычно исходного объекта связи), то без вторжения в них создавать связи невозможно. А эта операция доступна только владельцу метаданных рассматриваемого объекта.

Другое достоинство предлагаемого подхода заключается в том, что в нашем случае используется более многоаспектная онтология и основанная на ней таксономия семантических связей. Связываемыми информационными объектами могут быть не только научные публикации, но и научные информационные объекты иных типов, а также объекты, представляющие авторов публикаций, пользователей системы, организации, в рамках которых выполнялись публикуемые работы и т.п. Поэтому естественно, что онтология связей Соционет определяет более богатое множество отношений, воплощаемых семантическими связями, поддерживаемыми в системе. Благодаря этому, анализируя структуру таких связей, можно получать различные новые количественные и качественные результаты, которые не позволяют получать существующие индексы цитирования. Важно отметить, что при этом предусматривается возможность использования одновременно нескольких альтернативных таксономий связей.

Рассматриваемый подход обеспечивает создание и динамическое развитие пользователями системы семантической структуры ее контента, которая представляет собой своего рода его *«семантический ореол»* (Semantic Halo [13]). Благодаря ему пользователи получают информационно насыщенное представление о структуре мнений ученых по поводу существующих в системе информационных объектов. Вместе с тем, обеспечивается *семантическая навигация* по контенту системы, которая может осуществляться по слоям семантической структуры, соответствующим классам используемой таксономии, и создает комфортные условия для доступа пользователей к информационным ресурсам системы.

Отметим, наконец, еще одну важную особенность обсуждаемого в данной работе подхода. Он открывает возможности для новых форм научной деятельности, воплощаемой в виртуальной среде онлайновой системы. Электронная библиотека, в которой реализован обсуждаемый подход, представляет собой фактически социальную сеть, в среде которой совместно действуют представители научного сообщества. Результатами их деятельности являются представленные в явном виде мнения о научных публикациях, а также развивающаяся семантическая структура контента системы, позволяющая использовать новые методы наукометрических исследований. Представление семантических связей в системе как самостоятельных информационных объектов позволяет декларировать мнения о них точно так же, как и относительно других информационных объектов. Такая поддержка мнений о мнениях образует своеобразный дискуссионный форум в среде системы.

Авторы намерены продолжить работу по развитию инструментария, реализующего рассмотренный подход, в нескольких направлениях. В частности, предполагается создать механизм поддержки альтернативных таксономий семантических связей. Планируются также эксперименты по интеграции репозиториев семантических связей. Наконец, будут созданы дополнительные сервисы для наукометрических исследований, учитывающие новые информационные возможности семантически структурированного контента системы.

Литература

[1]. ГРНТИ – рубрикатор научно-технической информации. Редакция 2007 года. – URL: http://www.grnti.ru/ [Дата обращения 15 июля 2013 г.]

[2]. Клековкина М.В., Котельников Е.В. Метод автоматической классификации текстов по тональности, основанный на словаре эмоциональной лексики. Труды 14-й Всероссийской научной конференции «Электронные библиотеки: перспективные методы и технологии, электронные коллекции» — RCDL-2012, Переславль-Залесский, Россия, 15-18 октября 2012 г. Переславль-Залесский, Институт программных систем РАН, 2012. – С. 118-123.

[3]. Когаловский М.Р. Онтологическое аннотирование библиографических ссылок в научных публикациях и его использование в наукометрии //Информационные ресурсы России (в печати).

[4]. Когаловский М.Р. Методы интеграции данных в информационных системах. Депонент Соционет, 2010.http://socionet.ru/pub.xml?h=RePEc:rus:rssalc:web-39 [Дата обращения 15 июля 2013 г.]

[5]. Когаловский М.Р., Паринов С.И. Использование связей цитирования для наукометрических измерений в системе Соционет. Институт проблем рынка РАН, Центральный экономико-математический институт РАН, 2009. Электронный депонент Соционет. http://socionet.ru/publication.xml?h=repec:rus: rssalc:web-32 [Дата обращения 15 июля 2013 г.]

[6]. Когаловский М.Р., Паринов С.И. Семантическое структурирование контента научных электронных библиотек на основе онтологий. В кн.: "Современные технологии интеграции информационных ресурсов: сборник научных трудов». - Санкт-Петербург: Президентская библиотека им. Б.Н. Ельцина, 2011.

[7]. Когаловский М.Р., Паринов С.И. Классификация и использование семантических связей между информационными объектами в научных электронных библиотеках //Информатика и ее применения. 2012. Т. 6. Вып. 3. С. 32-42.

[8]. Паринов С.И., Когаловский М.Р. Технология семантического структурирования контента научных электронных библиотек. Труды XIII Всероссийской научной конференции «Электронные библиотеки: перспективные методы и технологии, электронные коллекции – RCDL-2011. Воронеж, 19-22 октября 2011 г.». – г. Воронеж: Воронежский государственный университет, 2011. – С. 94-103.

[9]. Когаловский М.Р., Паринов С.И. Наукометрические измерения в электронных библиотеках на основе рубрикаторов научной информации //Электронные библиотеки (электронный журнал). 2012. Т. 15, № 6. http://www.elbib.ru/ index.phtml?page=elbib/rus/journal/2012/part6/KP [Дата обращения 15 июля 2013 г.]

[10]. Паринов С.И., Ляпунов В.М., Пузырев Р.Л. Система Соционет как платформа для разработки научных информационных ресурсов и онлайновых сервисов //Российский научный электронный журнал «Электронные библиотеки». – 2003. – Том 6. – Вып. 1. http://www.elbib.ru/index.phtml?page=elbib/rus/journal/2003/part1/PLP [Дата обращения 15 июля 2013 г.]

[11]. CERIF 1.3 Full Data Model (FDM): Introduction and Specification. euroCRIS, 2012. http://www.eurocris.org/Uploads/Web%20pages/CERIF-1.3/Specifications/CERIF1.3_FDM.pdf [Дата обращения 15 июля 2013 г.]

[12]. CERIF 1.3 Semantics: Research Vocabulary. CERIF Task Group, euroCRIS, 2012. http://www. eurocris.org/Uploads/Web%20pages/CERIF-1.3/ Specifications/CERIF1.3_Semantics.pdf [Дата обращения 15 июля 2013 г.]

[13]. Dix A., Levialdi S. & Malizia A. Semantic halo for collaboration tagging systems. In the Social Navigation and Community-Based Adaptation Technologies Workshop-June 20th, 2006.

[14]. Feldman R. Techniques and Applications for Sentiment Analysis. Communications of the ACM, April 2013, vol. 56, no. 4, pp. 82-89.

[15]. Galassini C., Malizia A., and Bellucci A. An approach for developing intelligent systems for sentiment analysis over social networks. Intelligent Systems and Control /742: Computational Bioscience, J.F. Whidborne, P. Willis, G. Montana, Eds. Cambridge, United Kingdom, July 11 – 13, 2011.

[16]. Groth P., Gibson A., Velterop J. The Anatomy of a Nano-publication. Information Services and Use 30(1/2) (2010). http://iospress.metapress.com/ content/ftkh21q50t521wm2/ [Дата обращения 15 июля 2013 г.]

[17]. F1000Research. http://f1000research.com/ [Дата обращения 15 июля 2013 г.]

[18]. Journal of Economic Literature (JEL) Classification System. – URL: http://www.aeaweb.org/jel/jel _class_system.php[Дата обращения 15 июля 2013 г.]

[19]. Kalinichenko L.A., Briukhov D.O., Skvortsov N.A., Zakharov V.N. Infrastructure of the subject mediating environment aiming at semantic interoperability of heterogeneous digital library collections. Вторая Всероссийская научная конференция «Электронные библиотеки: перспективные методы и технологии, электронные коллекции», Протвино, 2000, с. 78-90.

[20]. Karlsson S. About LogEc, 2011. http://logec.repec. org/about.htm [Дата обращения 15 июля 2013 г.]

[21]. Open Archives Initiative. http://www. Openarchi ves.org/ [Дата обращения 15 июля 2013 г.]

[22]. Pang B., Lee L. Opinion Mining and Sentiment Analysis //Foundations and Trends in Information Retrieval. 2008. Volume 2, Issue 1-2. January 2008, pp. 1-135. http://dl.acm.org/citation.cfm?id= 1454712 [Дата обращения 15 июля 2013 г.]

[23]. Parinov S. The electronic library: using technology to measure and support Open Science. In: Proceedings of the World Library and Information Congress: 76th IFLA General Conference and Assembly, Gothenburg, Sweden, August 10-15, 2010. http://www.ifla.org/files/hq/papers/ifla76/155-parinov-en.pdf [Дата обращения 15 июля 2013 г.]

[24]. Parinov S. Open Repository of Semantic Linkages. In: Proceedings of 11th International Conference on Current Research Information Systems e-Infrastructure for Research and Innovations (CRIS 2012), Prague, 2012. http://socionet.ru/publication. xml?h=repec:rus:mqijxk:29 [Дата обращения 15 июля 2013 г.]

[25]. Parinov S. Towards a Semantic Segment of a Research e-Infrastructure: necessary information objects, tools and services. Metadata and Semantics Research, Communications in Computer and Information Science. J. M. Dodero, M. Palomo-Duarte, P. Karampiperis, Eds. Springer, vol. 343, 2012, pp. 133-145. http://socionet.ru/pub.xml?h= RePEc:rus: mqijxk:30 [Дата обращения 15 июля 2013 г.]

[26]. Semantic Web Applications in Neuromedicine (SWAN) Ontology. W3C Interest Group Note, 20 October 2009. http://www.w3.org/TR/2009/NOTE -hcls-swan-20091020/ [Дата обращения 15 июля 2013 г.]

[27]. Shotton D. CiTO, the Citation Typing Ontology. J. of Biomedical Semantics 2010, 1(Suppl 1): S6. http://www.jbiomedsem.com/content/1/S1/S6 [Дата обращения 15 июля 2013 г.]

[28]. Shotton D. Introduction the Semantic Publishing and Referencing (SPAR) Ontologies. October 14, 2010. http://opencitations.wordpress.com/2010/10/ 14/introducing-the-semantic-publishing-and-refere ncing-spar-ontologies/ [Дата обращения 15 июля 2013 г.]

[29]. Shotton D. Use of CiTO in CiteULike. 2010. http://opencitations.wordpress.com/2010/10/21/use -of-cito-in-citeulike/[Дата обращения 15 июля 2013 г.]

[30]. Shotton D., Peroni S. CiTO, The Citation Typing Ontology, v2.0. – 2011. http://purl.org/spar/cito/ [Дата обращения 15 июля 2013 г.]

[31]. Shotton D., Peroni S. DoCO, the Document Components Ontology. – 2011. http://speroni.web.cs. unibo.it/cgi-bin/lode/req.py?req=http:/purl.org/ spar/doco [Дата обращения 15 июля 2013 г.]

[32]. Shotton D., Peroni S. Semantic annotation of publication entities using the SPAR (Semantic Publishing and Referencing) Ontologies /Beyond the PDF Workshop, La Jolla, 19 January 2011. http://imageweb.zoo.ox.ac.uk/pub/2010/Publicatio ns/Shotton& Peroni_semantic_annotation_of_ pub lication_entities.pdf [Дата обращения 15 июля 2013 г.]

[33]. SKOS Simple Knowledge Organization System Reference. W3C Recommendation, 18 August 2009. http://www.w3.org/TR/skos-reference/ [Дата обращения 15 июля 2013 г.]

[34]. Small H. Interpreting maps of science using citation context sentiments: a preliminary investigation. Scientometrics, Springer Netherlands, Volume 87, Issue 2, 2011, pp. 373-388

[35]. The Open Archives Initiative Protocol for Metadata Harvesting. Protocol Version 2.0 of 2002-06-14. Document Version 2008-12-07T20:42:00Z. http://www.openarchives.org/OAI/2.0/openarchive sprotocol.htm [Дата обращения 5 июля 2013 г.]

[36]. Wache H., Vogele T., Visser U., Stuckenschmidt H., Schuster G., Neumann H. and Hubner S. Ontology-Based Integration of Information — A Survey of Existing Approaches. In Proceedings of

IJCAI-01 Workshop: Ontologies and Information Sharing, Seattle, WA, 2001, pp. 108-117. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.12.7857[Дата обращения 15 июля 2013 г.]

New data source for scientometric studies

Mikhail R. Kogalovsky, Sergey I. Parinov

The paper is dedicated to discussion of an approach to creation of new data sources for the scientometric researches, based on technology of semantic structuring the content of scientific digital libraries. The technology was described by authors in earlier published their papers. The basic principles of offered approach and the results of its implementation in the environment of Socionet system are considered. Mentioned new scientometric data source is collection of semantic linkages between various information objects of the system content. The linkages are classified by given taxonomy that is represented by a set of semantic controlled vocabularies. Such data sources allow more multifold analysis in comparison with traditional scientometric researches of the scientific knowledge corpus supported by the system. According to authors opinion the creation of the global autonomous repositories of semantic linkages which integrate the linkage collections from various scientific digital libraries in analyzed knowledge areas is perspective direction of scientometrics evolution. This research is supporting by RFBR, project 12-07-00518-a, and RFH, project 11-02-12026-в.

Извлечение знаний и фактов из текстов диссертаций и авторефератов для изучения связей научных сообществ*

© Ю.В. Леонова © А.М. Федотов
Институт вычислительных технологий СО РАН,
Новосибирск
juli@ict.nsc.ru fedotov@sbras.ru

Аннотация

В данной работы выполнено исследование диссертаций и авторефератов с целью изучения структуры научных связей ученого (научное окружение ученого), структуры и динамики развития научных коллективов (научные школы), статистического исследование текста диссертаций. Такие исследования дают возможности изучения и оценивания тенденций развития различных научных направлений, идентифицировать персоны, научные центры и организации, научные школы, изучать взаимосвязи между отдельными сообществами.

1 Введение

Целью данной работы является изучение связей научных сообществ, в рамках которых осуществляется научная деятельность, основанное на анализе диссертаций и авторефератов. Научное сообщество понимается как совокупность исследователей-профессионалов, объединенных вокруг единой цели, научной школы или направления и представляет собой сложную систему, в которой действуют как отдельные ученые, так и разнообразные государственные институты, общественные организации, неформальные группы и т.д. Реализация этой цели включает в себя решение следующих задач: статистическое исследование текста диссертаций, исследование структуры научных связей ученого (научное окружение ученого), исследование структуры и динамики развития незримых научных коллективов (научные школы). Такие исследования дают возможности изучения и оценивания тенденций развития различных научных направлений, идентифицировать персоны, научные центры и организации, научные школы, изучать взаимосвязи между отдельными сообществами.

В настоящее время существует много работ [1-7], направленных на анализ диссертаций. Однако в литературе не было найдено примеров использования методов в приложении к техническим наукам. Большинство работ посвящены статистическому анализу диссертаций.

2 Информационная модель фактов

Согласно «Логико-философскому трактату» Л.Витгенштейна [8]мир состоит не из предметов (вещей), а из фактов. Факт выступает как нечто отличное от вещи, как некоторое отношение, как взаимодействие двух предметов. Мир рассматривается как нечто, определяемое связями (взаимодействиями). Любой факт при этом — фиксация некоего отношения. Все факты фиксируются фразами, например «молоток забивает гвоздь». Любое предложение структурировано вполне конкретным образом: оно может быть представлено как 2 (или 3, 4…) объекта, которые как-то связаны между собой. Элементарное предложение связывает 2 объекта, а вещь – нечто общее совокупности фактов. Таким образом, отношения и факты объявляются первичными, а вещи представляют собой пересечение, совокупность возможных отношений. То есть с вещью можно соотнести общую область «пересечения» множества фактов. Атомарный факт есть соединение (двух) объектов. Анализ фактов дает объекты или предметы. При этом по мере накопления фактов представление о вещи может меняться. Благодаря такой трактовке мира вещь выступает не как нечто данное, застывшее, вполне определенное, а как некоторая сущность с размытыми границами, и эти границы уточняются по мере выявления класса возможных для данной сущности отношений (фактов). Чтобы определить вещь, надо зафиксировать все факты (положительные — где может встречаться эта вещь и отрицательные, где не может).

Таким образом, мир подразделяется на факты. Факт — существование событий. Событие – связь объектов (предметов, вещей).

Труды 15-й Всероссийской научной конференции «Электронные библиотеки: перспективные методы и технологии, электронные коллекции» — RCDL-2013, Ярославль, Россия, 14-17 октября 2013 г.

Факты в тексте можно представить в виде языковой модели, способной содержать, хранить и передавать информацию. Языковые модели, содержащие целенаправленно отобранную информацию, принято называть информационными моделями.

3 Модель документа в системе

Информационная система представляет собой множество связанных различными отношениями документов, описывающих некие сущности (объекты, факты или понятия) [9]. Информация о той или иной сущности содержится в системе либо непосредственно в виде документа, который ее представляет, описывает или моделирует, либо в виде упоминаний об этой сущности, которые имеются в других документах, т. е. содержат опосредованную информацию об этой сущности.

Согласно стандартам построения открытых систем (OSI) [10] структура и содержание документа должны описываться в соответствии с международными схемами данных. Для описания соответствующих схем данных используются метаданные, которые определяют структуру и смысловое содержание документа. В нашей системе документом называется информационный ресурс, снабженный метаописанием (метаданными) в соответствии с рекомендациями OSI.

Дадим два определения:

Документом d_i называется пара $d_i = <S_i, V_i>$, где S_i - структура документа в соответствии с выбранной схемой данных; V_i - содержание документа (информационное наполнение).

Коллекция - множество документов с выделенной фиксированной структурой, содержание которых имеет одинаковую тематическую направленность.

С точки зрения унификации работы с документами будем представлять информационную систему в виде набора коллекций. Метаданные, описывающие структуру и содержание документов в коллекциях, подразделяются на описательные и структурные.

Структурные метаданные определяют структуру и свойства документов, в соответствии с которыми осуществляется их обработка (типы, связи, форматы представления, ограничения на управление доступом и т. п.).

Описательные метаданные описывают смысловое содержание документа (его название, краткое содержание и т. п.).

Отметим, что описательные метаданные, характеризующие документ, могут являться частью документа и в то же время могут содержать в соответствии с выбранной схемой данных сведения о документе (основные и дополнительные, такие, как, например, авторы, название, дата создания и т. д.).

Элемент схемы данных данной коллекции будем называть структурным элементом.

Структурный элемент (далее просто элемент) имеет идентификатор и обладает некоторыми свойствами. Таким образом, элемент E –- это совокупность <ID, P>, где ID –- идентификатор элемента, P - свойства элемента.

Экземпляр элемента имеет значение (или содержание). Свойства элемента определяют характер работы с элементом. Элемент обладает типом, выбираемым из словаря. Тип определяет правила работы с элементом и, следовательно, является свойством элемента.

Примеры элементов: заголовок документа, аннотация документа, фамилия в визитной карточке, авторы документа. Значение элемента – его конкретная содержательная часть, а свойства элемента описывают его структуру. Для элемента визитной карточки “Фамилия” значение - Матвеев, идентификатор – 1, свойства – тип “word”.

Структура документа – это набор структурных элементов.

Содержание документа – объединение значений экземпляров элементов, составляющих документ.

Информационная система содержит коллекции:

1) Персоны и организации, диссертационные советы
2) Авторефераты и диссертации. Диссертация обладает документной и лингвистической информативностью. Документная информативность связана с реализацией сигнальной функции, которая дает информацию организационного характера, т.е. извещает о том, что диссертация подготовлена и поступила в библиотеку организации по месту работы диссертационного совета, о месте и времени защиты, об ученых, являющихся оппонентами по диссертации. Она реализуется в таких атрибутах описания, как «соискатель», «тема», «специальность», «дата защиты», «организация, в которой выполнена работа», «шифр совета», «научный руководитель» (ФИО, ученая степень, звание), «оппоненты», «ведущая организация», «название организации, где можно ознакомиться с диссертацией», «дата рассылки автореферата», «ученый секретарь», «УДК». Лингвистическая информативность реализуется в автореферате или диссертации в атрибуте «Текст».
3) Термины. Особым видом объектов ИС является Термин. Термин – слово или словосочетание название определённого понятия какой-нибудь специальной области науки, техники, искусства, общественной жизни и т.п. Термин называет специальное понятие и в совокупности с другими терминами данной системы является компонентом научной теории определенной области знания [11]. Примером терминов являются ключевые слова, описывающие содержание диссертации.

4 Модель отношений между документами в системе

Для решения сформулированных выше задач мы должны определить связи (отношения) между документами.

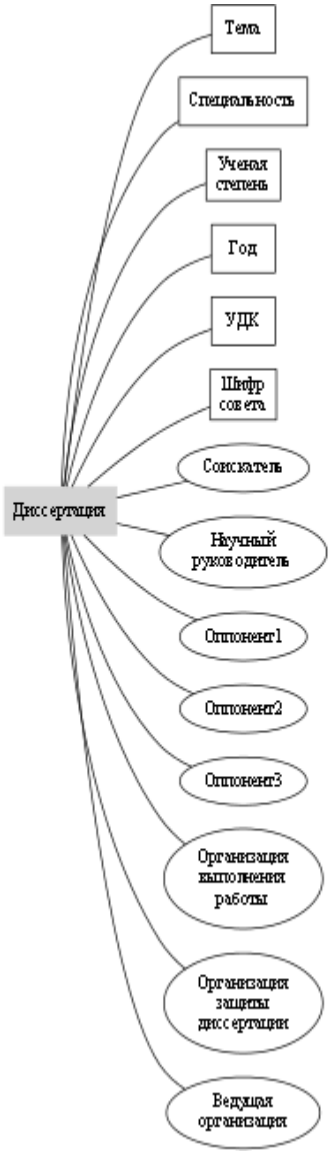


Рис. 1. Информационная модель описания диссертации

В основу нашей модели отношений [9] между документами в информационной системе легла модель RDF. В нашей системе связи между документами устанавливаются путем задания на множестве документов бинарных отношений, которые в соответствии с правилами RDF могут быть записаны в виде A(R,V): объект R имеет атрибут A со значением V. Например, тот факт, что Барахнин В.Б. занимает некоторую должность (post) в ИВТ СО РАН, записывается как Post(`ИВТ СО РАН', `Барахнин В.Б.'), где Post - то или иное значение из списка (тезауруса) должностей.

Связь — это направленное или ассоциативное отношение между объектами системы, например Петров А.А. преподает в НГУ. Факт — событие (как правило, зафиксированное и произошедшее), которое может сопровождаться временной и географической метками и др., например, Иванов П.П. защитил кандидатскую диссертацию в 1994 году в г. Новосибирск. События представляют действия, происходящие в реальном мире, и определяются указанием типа действия и ролей, которые играют сущности в этом действии. Факт может быть извлечен из текста документов либо определен экспертом.

Как говорилось ранее, событие – связь объектов, то факт может определить как отношение между объектами, которое может иметь временные и географические атрибуты, например, год – 1994, географическая привязка - Новосибирск.

Можно выделить следующие виды связей:

- Прямые. В этом случае есть факт о связи двух объектов, например, отношение соискатель-оппонент
- Нечеткие (не представленные фактом):
  - по общему месту и времени у пары различных фактов различных объектов, например, дата и место защиты диссертации позволяет установить соискателей, защитивших диссертацию в один день в одном совете;
  - косвенные (транзитивные) — через общий третий объект-отношение у пары фактов различных объектов, например, связь диссертация-ключевые слова. Установление связи подобных диссертаций выполняется через ключевые слова

Факты можно выразить посредством высказываний с использованием предикатов. Методы математической логики позволяют формализовать эти утверждения и представить их в виде, пригодном для анализа.

Рассмотрим высказывание: "Преподаватель Иванов А.А, родился в 1962 году". Оно выражает следующие свойства сущности "Иванов А.А.":

- в явном виде – год рождения;
- в неявном виде – принадлежность к преподавателям.

Первое свойство устанавливает связь между парами сущностей "Иванов А.А." и "год рождения", а второе свойство устанавливает связь между парами сущностей "Иванов А.А." и "множество преподавателей". Формализация этого высказывания представляется как результат присваивания значений переменных, входящих в следующие предикаты:

РОДИЛСЯ (Иванов А.А., 1962)

ЯВЛЯЕТСЯ ПРЕПОДАВАТЕЛЕМ (Иванов А.А.)

Пример информационной модели описания диссертаций (Рис. 1). Существенными характеристиками диссертации являются «соискатель», «тема», «специальность», «ученая степень», «год»,«организация, в которой выполнена работа», «организация, в которой защищалась диссертация», «шифр совета», «научный руководитель», оппоненты», «ведущая организация», «УДК». Связи между документом и его элементами представлены на рисунке, который дает схемное описание рассматриваемой модели. В этом описании используются следующие элементы: соискатель, оппонент1, оппонент2, оппонент3, научный руководитель, организация выполнения работы и организация защиты диссертации, ведущая организация - объекты, тема, специальность, ученая степень, шифр совета, УДК - текстовые значения, год - числовое.

Формализованное описание данной модели является предикатом с именем диссертация:

диссертация (Соискатель, тема, год, специальность, ученая степень, организация выполнения работы, организация защиты диссертации, ведущая организация, шифр совета, научный руководитель, оппонент1, оппонент2, оппонент3, УДК).

Для конкретных значений аргументов этот предикат превращается в факт. Например, если Барахнин В.Б. защитил диссертацию “Программные системы информационного обеспечения научной деятельности: модели, структуры и алгоритмы” в 2011 году, то имеет место факт: Диссертация (Барахнин В.Б., Программные системы информационного обеспечения научной деятельности: модели, структуры и алгоритмы, 2011, 05.13.17 , доктор технических наук, Институт вычислительных технологий СО РАН, Московский государственный университет печати, Институт математики СО РАН, Д 212. 147.03, Федотов А.М., Шайдуров В.В., Хорошевский В.Ф., Мальцева С.В., 004). С помощью таких фактов можно выделить различные характеристики диссертаций, например, можно выделить соискателей, защитивших диссертацию по специальности 05.13.17 в 2011 году.

5 Статистическое исследование текста диссертации

При исследовании текста диссертаций используется метод контент-анализа – метод качественно-количественного анализа содержания документов с целью выявления или измерения различных фактов и тенденций, отраженных в этих документах. Сущность метода контент-анализа состоит в выделении в содержании научных документов некоторых ключевых признаков (содержательных единиц анализа, проблем, категорий), которые отражают существенные (фактические и смысловые) стороны содержания с последующим подсчетом частоты употребления этих единиц [12, 13].

В данной работе используется тезаурусный метод, являющийся разновидностью контент-анализа, суть которого состоит в сведении рассматриваемого текста к ограниченному набору элементов и терминов, которые затем подвергаются анализу.

Не все документы могут выступить объектом контент-анализа. Необходимо, чтобы исследуемое содержание позволило задать однозначное правило для надежного фиксирования нужных характеристик (принцип формализации), а также чтобы интересующие исследователя элементы содержания встречались с достаточной частотой (принцип статистической значимости). Можно выделить следующие направления применения контент-анализа:

а) выявление того, что существовало до текста и что тем или иным образом получило в нем отражение (текст как индикатор определенных сторон изучаемого объекта — окружающей действительности, автора или адресата);
б) определение того, что существует только в тексте как таковом (различные характеристики формы – язык, структура и жанр сообщения, ритм и тон речи);
в) выявление того, что будет существовать после текста, т.е. после его восприятия адресатом (оценка различных эффектов воздействия).

Основой содержания диссертации является принципиально новый материал, включающий описание новых фактов, явлений и закономерностей, или рассмотрение имеющегося материала в совершенно ином аспекте. Таким образом, автор диссертации сосредоточен на описании новых фактов, их точном представлении научной общественности и их контент-анализ предполагает выявление фактов, существовавших до написания текста диссертации.

В разработке и практическом применении контент-анализа выделяют несколько стадий. После того, как сформулированы тема, задачи и гипотезы исследования, определяются категории анализа, т.е. наиболее общие, ключевые понятия, соответствующие исследовательским задачам.

В данном исследовании категорией анализа содержания диссертации является ее тема, соответствующая специальности ВАК.

После того, как категории сформулированы, необходимо выбрать соответствующую единицу анализа – лингвистическую единицу речи или элемент содержания, служащие в тексте индикатором интересующих исследователя явлений.

Единицы анализа, взятые изолированно, могут быть не всегда правильно истолкованы, поэтому они рассматриваются на фоне более широких лингвистических или содержательных структур, указывающих на характер членения текста, в пределах которого идентифицируется присутствие или отсутствие единиц анализа — контекстуальных единиц. Например, простейшим элементом текста является слово, для единицы анализа «слово» контекстуальная единица – «предложение».

Смысловыми единица контент-анализа могут быть:

а) понятия, выраженные в отдельных терминах;
б) темы, выраженные в целых смысловых абзацах, частях текстов, статьях;
в) имена, фамилии людей, названия организаций;
г) события, факты и т. п.;

Наконец необходимо установить единицу счета – количественную меру взаимосвязи текстовых и внетекстовых явлений. Выделение единиц счета, которые могут совпадать либо не совпадать с единицами анализа. В нашем случае процедура сводится к подсчету частоты упоминания выделенной смысловой единицы (интенсивность).

6 Научные связи

Научное пространство учёного N определим как совокупность учёных {S}, связанных с N различными научными отношениями, как например,

связи типа соискатель – научный руководитель, соискатель – оппонент, автор книги – редактор, автор книги – рецензент (не анонимный) и т.д.[14].

7 Научные коллективы

Коллектив – устойчивая во времени организационная группа взаимодействующих людей со специфическими органами управления, объединенных целями совместной общественно-полезной деятельности и сложной динамикой

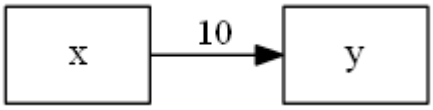


Рис. 2. Элемент графа

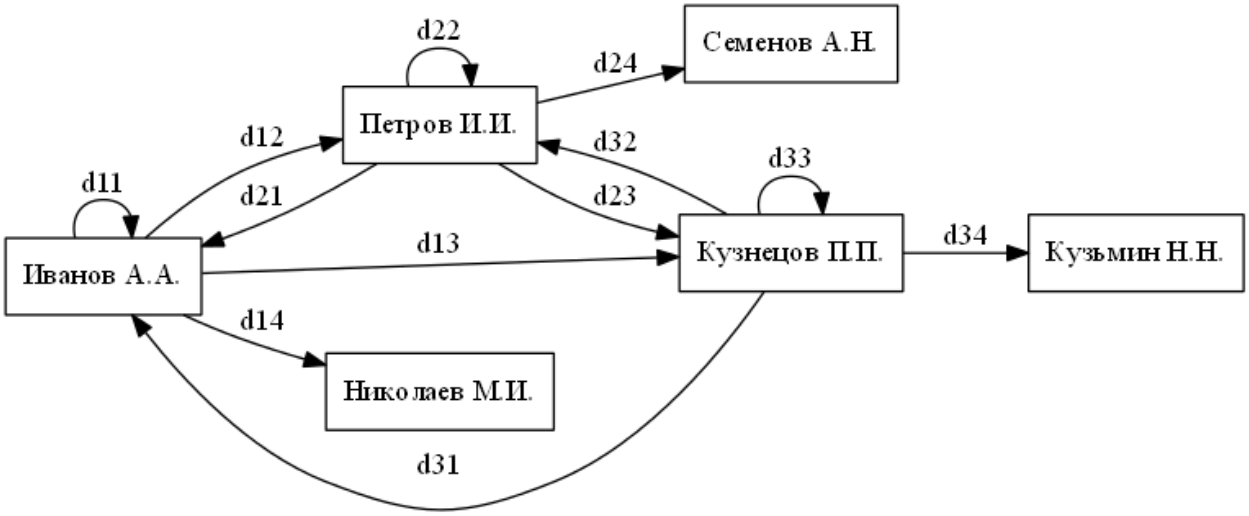


Рис.3. Фрагмент графа

формальных (деловых) и неформальных взаимоотношений между членами группы. Т.о. коллектив имеет сложную структуру, спектр всевозможных отношений, связей и взаимосвязей его членов весьма широк. Аппаратом описания структур коллективов, как и аппаратом описания отношений вообще является теория графов.

.Средством представления незримых коллективов является сеть (сеть идейного, творческого и пр. влияния) (рис. 2, 3). Звено сети (рис. 2) характеризует степень влияние x на y, и может означать, например, что «y цитирует x» 10 раз. Иначе говоря, y использовал концепции, идеи, факты x, развивал их и т. д. Тем самым между x и y имеется устойчивая информационная связь, причем число 10 — характеристика интенсивности этой связи [14].

Если построить сеть взаимных ссылок, то можно выделить подграфы, элементы которых интенсивно связаны друг с другом. Такие подграфы образуют незримые коллективы (на рис. 3 и подграф (Иванов А.А., Петров И.И., Кузнецов П.П.) — научный неформальный коллектив).

Неформальный коллектив из N элементов (N = 3) может быть представлен следующей матрицей NxN:

$$\mathbf{D} = \begin{array}{c} \\ a \\ b \\ c \end{array} \begin{array}{c} \begin{array}{ccc} a & b & c \end{array} \\ \begin{pmatrix} d_{11} & d_{12} & d_{13} \\ d_{21} & d_{22} & d_{23} \\ d_{31} & d_{32} & d_{33} \end{pmatrix} \end{array}.$$

Здесь d_{ij} — количество ссылок j на i (иначе говоря, мера неформального воздействия i на j). Например, d_{13} — количество ссылок с на а, и наоборот,— количество ссылок а на с. Здесь также можно ввести меру m(x) неформального (идейного, научного и пр.) статуса индивидуума x, например, следующего вида:

$$m(a) = \frac{d_{13}}{d_{31}} + \frac{d_{12}}{d_{21}} \quad \text{или} \quad m(a) = \frac{d_{13} + d_{12}}{d_{31} + d_{21}}$$

Эти меры используют различные выражения отношения «влияния а на остальных» к «влиянию остальных на а».

Лицо x с максимумом m(x) может быть названо лидером неформального коллектива. Между формальными и неформальными отношениями существуют определенные причинно-следственные связи. Например, может наблюдаться следующая последовательность их развития:

- a и b образуют неформальный коллектив (взаимные ссылки);

- a и b печатаются в соавторстве;
- a и b начинают работать вместе.

Выявление неформальных лидеров и коллективов способствует лучшей организации выполнения проектов путем привлечения в формальный коллектив единомышленников.

Описанный выше подход является статическим. Можно рассматривать развитие коллектива в динамике, когда с течением времени к графу добавляются новые вершины и рёбра и одновременно часть прежних элементов удаляется. Такие графы достаточно наглядно отображают перемены в коллективе, связанные, например, с уходом прежнего формального лидера.

Другим видом научных коллективов являются научные школы, информацию о которых можно получить на основе анализа таких реквизитов

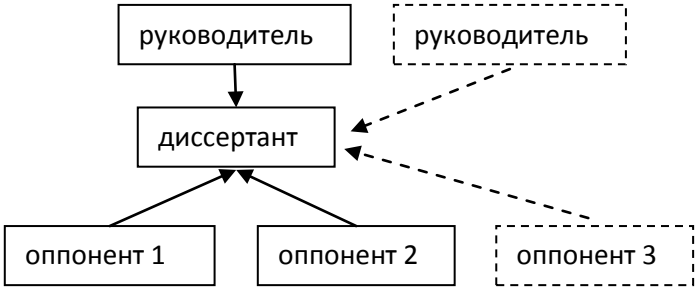


Рис. 4. Фрагмент графа диссертаций

диссертации, как учебное заведение, в котором выполнена работа, научный руководитель, ведущая организация, дата и время защиты, шифр совета и

т.д.Понятие «научной школы» употребляют «применительно к относительно небольшому

научному коллективу, объединенному не столько организационными рамками, не только конкретной тематикой, но и общей системой взглядов, идей, интересов, традиций – сохраняющейся, передающейся и развивающейся при смене научных поколений» .

Рассмотрим структуру графа диссертаций [15].Вершины ориентированного графа диссертаций соответствуют диссертантам, руководителям и оппонентам диссертантов. Бинарное отношение на парах вершин задается естественным образом: дуги выходят из вершин-руководителей и вершин-оппонентов и входят в соответствующую вершину-диссертант. Сохраняется информация о годе защиты, совете защиты, ведущей организации и т.п. Типичный фрагмент графа должен содержать 4 или более вершин (см. рис. 4).

1. **Вершины ориентированного графа диссертаций** соответствуют диссертантам, руководителям и оппонентам диссертантов. Бинарное отношение на парах вершин задается естественным образом: дуги выходят из вершин-руководителей и вершин-оппонентов и входят в соответствующую вершину-диссертант. Сохраняется информация о годе защиты, совете защиты, ведущей организации и т.п.
2. **Число входящих дуг** в вершину-диссертант лежит в границах от 3 до 8. Максимальная входящая степень будет у вершин-диссертантов, которые защитили кандидатскую и докторскую степени, имеют несколько руководителей и консультантов. Степени вершин-руководителей и вершин-оппонентов могут быть очень большими.
3. **Из вершины-диссертанта дуга будет выходить**, если он в дальнейшем стал руководителем или оппонентом какой-либо диссертации.

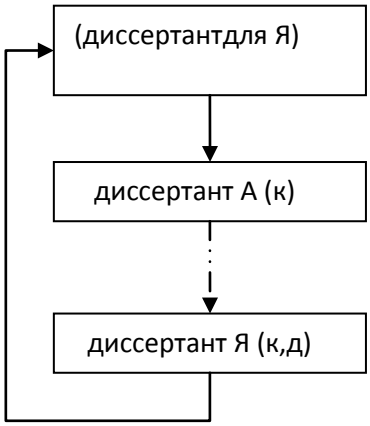


Рис. 5. Контур графа диссертаций

4. **Большие степени в графе** выявляют персон, оказавших большой влияние на формирование коллектива специалистов в данной области. Длинная цепь в графе показывает протяженный во времени процесс защит диссертаций, где в качестве руководителя выступает бывший диссертант и т.д. Таким образом, наличие больших степеней и длинных цепей позволяет предполагать существование школы по рассматриваемому направлению.

Граф может иметь контуры. На рисунке ниже показан пример образования контура: диссертант А защитил кандидатскую (к) диссертацию, далее стал руководителем другого диссертанта и т.д. После последовательности защит диссертант Я защитил кандидатскую и докторскую (д) диссертации и затем стал оппонентом докторской диссертации для кандидата наук, бывшего оппонентом диссертанта А.

8 Методы извлечение понятий из текста диссертации

Рассмотрим подробнее методику извлечения фактов из текста диссертации. Извлечение понятий из текста представляет собой технологию,

обеспечивающую получение информации в структурированном виде. В качестве структур могут запрашиваться как относительно простые понятия (ключевые слова, персоны, организации, географические названия), так и более сложные, например, имя персоны, ее должность в конкретной организации и т.п.

Данная технология включает три основных метода:

а) извлечение слов или словосочетаний, важных для описания содержания текста. Это могут быть списки терминов предметной области, персон, организаций, географических названий, и др.;

б) прослеживание связей между извлеченными понятиями;

в) извлечение сущностей, распознавание фактов и событий.

Подходы к извлечению различных типов понятий из текстов существенно различаются. Например, для выявления принадлежности

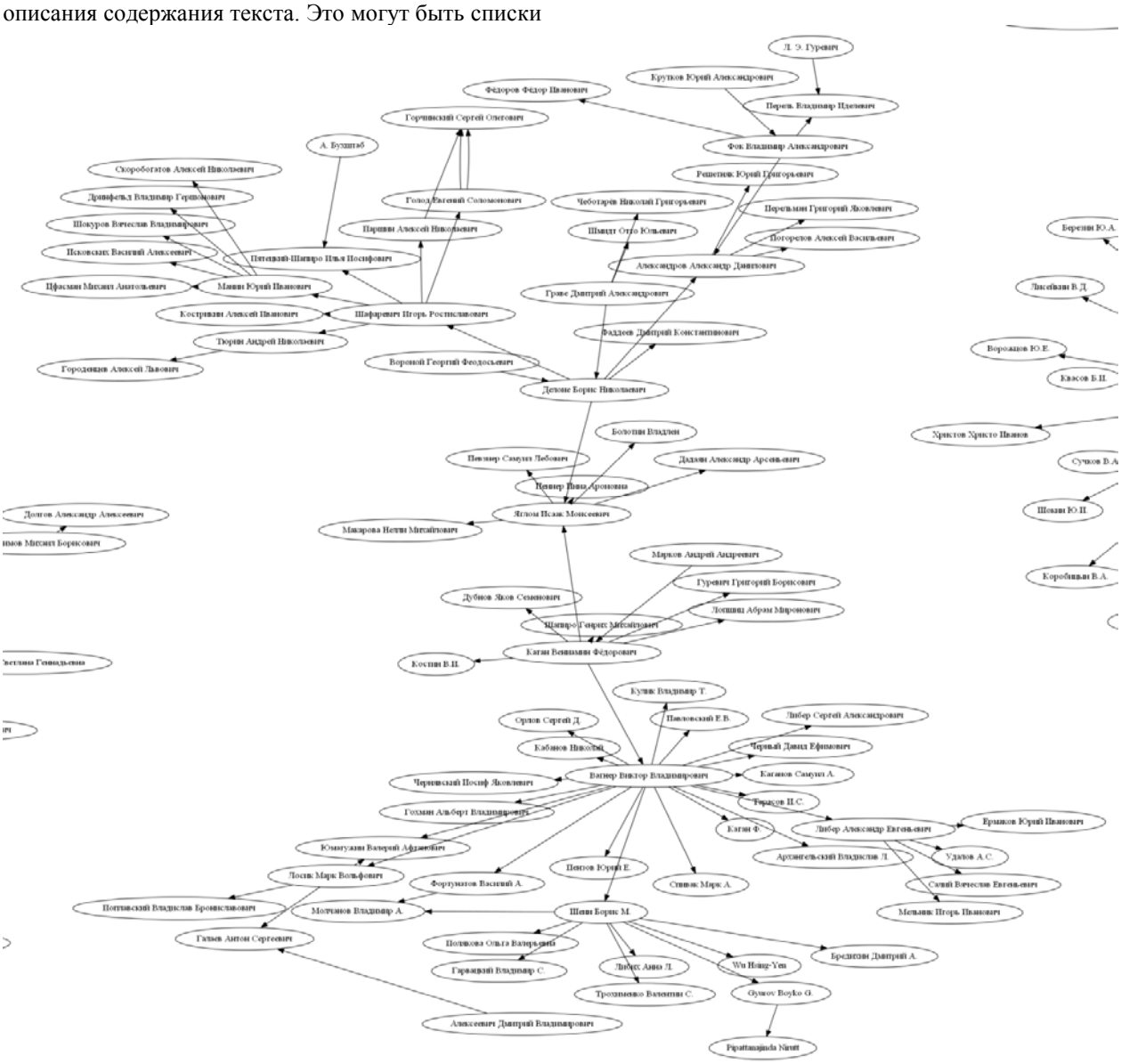


Рис. 6. Фрагмент графа диссертаций

документа к тематической рубрике могут использоваться методы классификации. Для выявления названий организаций и персон применяются как система шаблонов, так и результаты структурного исследования текста, например, используется таблица префиксов названий организаций. Выявление географических названий предполагает использование таблиц, в которых кроме шаблонов написания этих названий используются коды и названия стран, регионов и отдельных населенных пунктов. Таким образом, методы извлечения из текста сущностей и терминов имеют свою специфику для каждого типа.

Методы автоматического извлечения понятий можно разделить на 2 типа:

- Методы машинного обучения. Основываются на статистических (вероятностных) методах

извлечения знаний. Для обучения системы необходим размеченный корпус текстов.

- Методы, основанные на знаниях. Основываются на языках описания правил-шаблонов, которые составляются экспертами. Основой недостаток метода – написание правил может занимать много времени.

Методы, основанные на знаниях, используются при необходимости обеспечить максимально возможное качество извлечения, однако для их работы необходимо иметь словари, списки слов и экспертов – инженеров по знаниям, но при этом отсутствует необходимость иметь много размеченных данных.

Методы машинного обучения используются при необходимости обеспечить хорошее качество извлечения, при этом отпадает необходимость в экспертах и словарях, необходимо иметь большой объем размеченных данных.

Наиболее эффективными являются комбинированные методы.

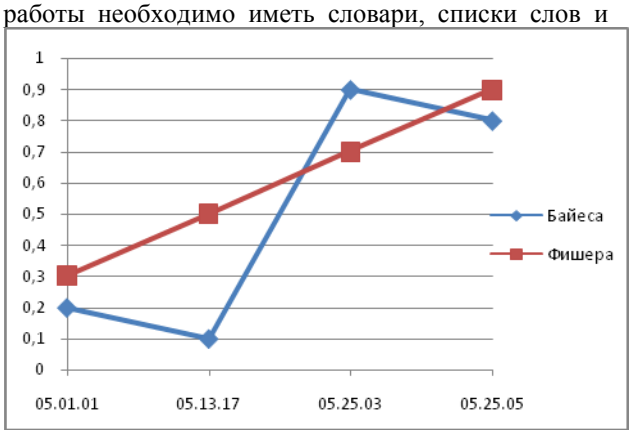


Рис.7. Точность. Зависимость от категории

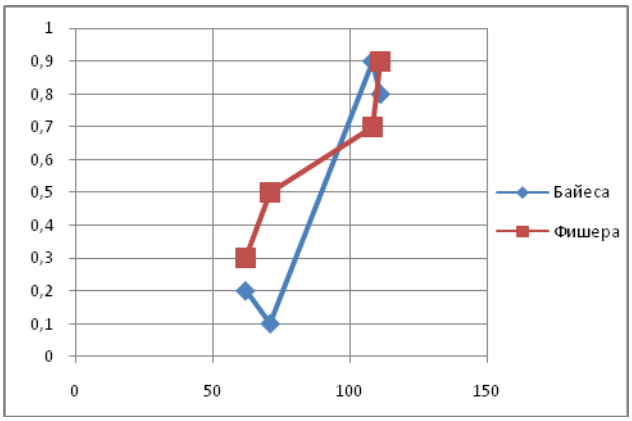


Рис.8. Точность. Зависимость от количества документов в рубрике

8.1 Извлечение именованных сущностей

Выделение сущностей является ключевым этапом предобработки текста для решения более сложных задач извлечения информации.

Под термином *именованная сущность* будем понимать объект определенного типа, имеющий имя, название или идентификатор.

Особенностями этого вида объектов являются:

- Большое множество разных сущностей;
- Отсутствуют строгие правила именования сущностей;
- Постоянно появляются новые сущности.

Какие типы выделяет система, определяется в рамках конкретной задачи. Для диссертаций и авторефератов – это *люди* (PER), *места* (LOC), *организации* (ORG), *время* (TIME). В общем случае системе на вход поступает текст, на выходе система сообщает информацию о положении имен в тексте и информацию о классах, которые им соответствуют.

Набор классов фиксируется заранее. Приведем пример размеченного текста:

[PER БарахнинВладимирБорисович].
Программныесистемы информационного обеспечения научной деятельности : модели, структуры и алгоритмы : диссертация доктора технических наук: 05.13.17 / Место защиты: [ORG Моск. гос. ун-тпечати].- [LOC Новосибирск], [TIME 2010].- 315 с.

Для извлечения именованных сущностей применяются несколько типов признаков [16]:

1. **признаки уровня слов** (N-граммы, суффиксы, префиксы, части речи и т.д.);
2. **признаки уровня документа** (наличие акронимов в корпусе, позиция термина в предложении, наличие термина в заголовке или тексте и т.д.);
3. **дополнительная информация** (слова указатели, например, Inc. ,Corp., списки стоп-слов, слов с капитализацией, которые не являются именованными сущностями и т.д.).

В пределах одного документа может быть несколько вхождений одного и того же имени, которое может относиться к одной сущности или же к различным объектам. В простейшем случае обычно исходят из предположения, что в одном документе одно и то же имя относится к одной и той же сущности.

Базовый набор признаков составлен из признаков первой группы для слов, находящихся в скользящем по тексту окне размера до 5 токенов. Под токеном подразумеваются не только слова, но и символы пунктуации.

Были проанализировано 4587 диссертаций и авторефератов и получен граф связей между персонами в диссертации на основании вышеприведенной модели (Рис. 4). Граф распадается на множество несвязанных компонент, в которых можно отыскать подграфы (Рис.6) с длинными цепями с длинной 2, что позволяет говорить о наличии научной школы.

8.2 Извлечение ключевых терминов из текста

Ключевыми терминами (ключевыми словами или ключевыми фразами) являются важные термины в документе, которые могут дать высокоуровневое описание содержания документа для читателя. Извлечение ключевых терминов является базисным этапом для многих задач обработки естественного языка, таких как классификация документов, кластеризация документов, суммаризация текста и вывод общей темы документа [17,18].

В данной работе используется метод выделения терминов на основе морфологических шаблонов, аключевые термины выражаются именными словосочетаниями. В именных словосочетаниях главным словом (основным носителем смысла) является, как правило, первое слева существительное, а остальные слова служат для уточнения значения главного слова.

Для выделения ключевых терминов используются следующие виды шаблонов

П+С – согласованные прилагательное + существительное;

С+Срод.п. – существительное + существительное в родительном падеже;

С+Ств.п. – существительное + существительное в творительном падеже;

П+П+С – согласованные прилагательное + прилагательное + существительное;

С+П+Срод.п. – существительное + согласованное прилагательное + существительное в родительном падеже;

С+П+Ств.п. – существительное + согласованное прилагательное + существительное в творительном падеже.

После выделения терминов определяется их тематика с помощью метода классификации – отнесение документа к одной из нескольких категорий на основании семантического содержания документа.

Для классификации применяются методы обучения с учителем, которые позволяют провести классификацию или спрогнозировать значение исходя из ранее предъявленных примеров.Из множества существующих методов были выбраны метод наивной классификации Байеса и метод Фишера.

Существенное преимущество наивных байесовских классификаторов по сравнению с другими методами заключается в том, что их можно обучать и затем опрашивать на больших наборах данных [19]. Даже если обучающий набор очень велик, обычно для каждого образца есть лишь небольшое количество признаков, а обучение и классификация сводятся к простым математическим операциям над вероятностями признаков.

Это особенно важно, когда обучение проводится инкрементно, – каждый новый предъявленный образец можно использовать для обновления вероятностей без использования старых обучающих данных. (Отметим, что код для обучения байесовского классификатора запрашивает по одному образцу за раз, тогда как для других методов, скажем деревьев решений или машин опорных векторов, необходимо предъявлять сразу весь набор.) Поддержка инкрементного обучения очень важна для таких в случаях расширении набора категорий в классификаторе, который постоянно обучается на вновь поступающих документах, должен обновляться быстро и, возможно, даже не имеет доступа к старым документам. Еще одно достоинство наивных байесовских классификаторов – относительная простота интерпретации того, чему классификатор обучился. Метод Фишера - альтернативный метод классификации, обеспечивает большую гибкость при настройке параметров классификации.

Результаты тестирования точности алгоритмов классификации терминовприведены на Рис.7 и Рис.8., что позволяет сделать выводы о точности алгоритмов классификации около 90%при количестве документов в рубрике более ста.

Литература

[1] К.В. Бугаев Отграничение криминалистики от иных наук методами информационного анализа текста// Юридический мир. -2011. - № 8. - С. 40 – 43.

[2] Бескаравайная Е. В.. Анализ базы данных дисертаций ПНЦ РАН / Е. В. Бескаравайнова, И. А. Митрошин // Информационное обеспечение науки: новые технологии. - М.: Научный Мир, 2011. - С. 124-133.

[3] Прошанов С.Л.Докторские диссертации по социологии (1990-2010 гг.) // Социологические исследования. - 2011.-№1. - С.30-39.

[4] Липский С. И. Проблемно-тематический анализ диссертационных исследований по социальной педагогике (1971-2008 гг.) Автореферат диссертации, Кострома - 2009

[5] H. Anil Kumar, Mallikarjun Dora Citation analysis of doctoral dissertations at IIMA: A review of the local use of journals // Library Collections, Acquisitions, and Technical Services - Vol. 35, Issue 1, Spring 2011, P. 32–39

[6] Kam C. Chan, Kam C. Chan, Gim S. Seow, Kinsun Tam Ranking accounting journals using

dissertation citation analysis: A research note // Accounting, Organizations and Society - Vol. 34, Issues 6–7, 2009, P. 875–885

[7] DilekAltun, ÇağlaÖnerenŞendil, İkbal Tuba Şahin Investigating the National Dissertation and Thesis Database in the Field of Early Childhood Education in Turkey // Procedia - Social and Behavioral Sciences - Vol. 12, P. 1-654 (2011) - International conference on education and educational psychology, 2–5 December 2010, Cyprus

[8] Гайдадымов Евгений - Философия (Конспект лекций) //ЭЛЕКТРОННАЯ БИБЛИОТЕКА ModernLib.Ru

[9] Барахнин В.Б., Леонова Ю.В. Информационная модель отношений между документами в информационной системе. Вычислительные технологии. – 2005. - Том 10. Специальный выпуск. - С. 129-137.

[10] Концепция открытых систем // Материалы к межотраслевой Программе “Развитиеи применение открытых систем”. [http://www.informika.ru/text/inftech/opensys/3/concept/os_1.html]

[11] Большой Энциклопедический словарь. 2000

[12] О.Т. Манаев Контент-анализ как метод исследования // «ПСИ-ФАКТОР»

[13] Хайтун С.Д. Наукометрия: Состояние и перспективы. — М.: Наука, 1983.

[14] Элементы математической теории организации //Портал Cadmium http://cadmium.ru/content/view/832/45/

[15] Леонова Ю.В., Добрынин А.А., Веснин А.Ю. Построение графа диссертаций // XIV Российская конференция с участием иностранных ученых «Распределенные информационные и вычислительные ресурсы» (DICR-2012): программа конференции и тезисы докладов (Новосибирск, Россия, 26-30 ноября 2012). – Новосибирск: ИВТ СО РАН. – 2012. – с. 17 [ISBN 978-5-905569-05-0].

[16] Л.М. Ермакова Методы извлечения информации из текста // Вестник Пермского университета. Сер.: Математика. Механика. Информатика. - 2012. - Вып. 1 (9). - С. 77-84.

[17] Manning, C. D., and Schtze, H. 1999. Foundations of Statistical Natural Language Processing. The MIT Press.

[18] Гринева М., Гринев М., Лизоркин Д. Анализ текстовых документов для извлечения тематически сгруппированных ключевых терминов // Тр. Ин-та системного программирования РАН. — URL: http://citforum.ru/database/articles/kw_extraction/

[19] Сегаран Т. Программируем коллективный разум. – Пер. с англ. – СПб: Символ-Плюс, 2008.

Extraction of knowledge and facts from texts of theses and abstracts for studying of communications of scientific communities

Yuliya V. Leonova, Anatolii M. Fedotov

In this work a research of theses and abstracts for the purpose of studying of structure of scientific communications of a scientist (a scientific environment of a scientist), structure and dynamics of development of research teams (schools of sciences), statistical research of the text of theses is undertaken. Such researches give the chance of studying and estimations of trends of development of various scientific directions, to identify persons, scientific centers and the organizations, schools of sciences, to study interrelations between separate communities.

Извлечение информации о ситуациях отставок-назначений в новостных текстах. Опыт разметки коллекции. Результаты тестирования.

© Н.А.Власова
Исследовательский Центр Искусственного Интеллекта
Института Программных Систем РАН имени А.К.Айламазяна,
г. Переславль-Залесский
nathalie.vlassova@gmail.com

Аннотация

В настоящей работе описан эксперимент по разметке коллекции новостных текстов с целью оценить эффективность подхода к извлечению информации о ситуациях отставки-назначения в системе ИСИДА-Т. Система ИСИДА-Т разрабатывается в ИПС РАН в течение нескольких лет и реализует инженерный подход к извлечению информации из текстов. В данной статье описывается попытка реализации подхода, описание тестовой коллекции, а также приводятся полученные рузультаты.

1 Введение

В ИЦИИ ИПС РАН в рамках проекта ИСИДА-Т[1],[6],[7] ведётся работа по извлечению из новостных текстов информации о ситуациях назначения-отставки. На RCDL-2012 был представлен доклад о концепции извлечения информации о ситуациях отставки-назначения в рамках инженерного подхода, реализуемого в системе ИСИДА-Т [11]. Для экспериментальной оценки эффективности данной разработки была размечена коллекция из 231 документа (новостные сообщения), в которых встречается 868 ситуаций отставки-назначения. Была проведена настройка системы правил, более детально проработаны контексты, описывающие целевые ситуации, и отлажено программное обеспечение. Были получены первые числовые данные, позволяющие судить об эффективности работы системы при извлечении ситуаций. При написании статьи были учтены вопросы и замечания, полученные автором на RCDL-2012.

2 Понятие текстовой ситуации и принципы разметки текстов. Исходные данные для анализа ситуаций

Прежде всего необходимо определить, что же будет считаться ситуацией, информация о которой должна быть извлечена из текста. Будем различать текстовую ситуацию и ситуацию внетекстовую. **Внетекстовая** ситуация – это ситуация, произошедшая во внеязыковой действительности. Например, факт, что 19 апреля 2013 года правительство России отправило в отставку руководителя “Почты России” Александра Киселёва, - это факт из внеязыковой действительности. В ситуации увольнения – три участника:

кто уволил (Правительство России) – **1-й участник**,

кого уволил (Александра Киселёва) – **2-й участник**,

с какой должности (руководителя “Почты России”) – **3-й участник**.

Вне текста наше знание о ситуации базируется на информации о её участниках и отношениях между ними. В текстах об этом событии может быть рассказано по-разному:

1. Решение об отставке руководителя "Почты России" правительством принято.

2. Правительство России приняло решение об отставке главы "Почты России" Александра Киселёва

3. Бывший глава «Почты России» Александр Киселев получит после отставки с должности более 3 миллионов рублей,

4. На фоне коллапса, возникшего в работе «Почты России», правительство отправило в отставку генерального директора предприятия Александра Киселёва

5. Александр Киселев, об отставке которого с поста генерального директора «Почты России»

Труды 15-й Всероссийской научной конференции «Электронные библиотеки: перспективные методы и технологии, электронные коллекции» — RCDL-2013, Ярославль, Россия, 14-17 октября 2013 г.

стало известно 19 апреля, получит после увольнения чуть более трех миллионов рублей.

6. Отставку Александра Киселева спровоцировал «посылочный коллапс» в аэропортах московского авиаузла.

7. Александр Киселев в ближайшее время может покинуть пост главы «Почты России»

8. За что увольняют директора "Почты России" Александра Киселёва?

9. Александр Киселев, уволенный с поста генерального директора "Почты России", после ухода в отставку получит золотой парашют на 3 млн рублей.

10. Уволен глава "Почты России"

(все примеры взяты с новостных порталов в сети Интернет)

Легко заметить, что в новостных текстах далеко не всегда упоминаются все три участника ситуации. Чаще всего попадаются контексты с 2 участниками. И почти никогда не упоминается дата события. Кроме того, ситуация может быть описана не одним предложением, а несколькими (см. пример 2). То есть в тексте может не содержаться исчерпывающая информация о ситуации. А в отдельном предложении это вообще встречается редко. Таким образом, построить полную картину, ограничившись рамками одного предложения, практически невозможно. Более того, в тексте может говориться о возможной ситуации - ситуации, которая не произошла в действительности, и, возможно, не произойдёт (например: *на заседании речь шла о возможной отставке министра образования*).

При разметке ситуаций назначения-отставки в текстах коллекции для экспериментальной работы необходимо чётко представлять, что будет считаться ситуацией. Под **текстовой ситуацией** мы будем понимать ситуацию, описанную в одном предложении (возможно, не произошедшую в действительности, но просто упомянутую в тексте) и выраженную с помощью целевого слова-маркера ситуации (слова, называющего ситуацию, – **уволил, назначил, отставка, назначив** и т.п.) и именных групп, описывающих участников ситуации. При этом мы исходим из предположения, что участники расположены контактно справа и слева от маркера ситуации (между участником и словом-ситуацией допускаются наречия, указания на время, частицы).

Наша задача – максимально полно и точно извлечь информацию из текстовых ситуаций, то есть определить слово-ситуацию, собрать всех участников и правильно распределить роли. В дальнейшем, располагая информацией о текстовых ситуациях в разных предложениях текста и зная их последовательность, мы сможем приблизиться к полной информации о внетекстовой ситуации.

В силу особенностей работы всех модулей системы ИСИДА-Т не размечаются и не включаются в общую группу ситуации:

- если хотя бы один из участников выражен именной группой с главным словом во множественном числе,

- если хотя бы один из участников выражается несколькими синтаксически однородными именными группами (*уволил Иванова и Петрова*),

- в случае эллипсиса (*Иванова назначили директором, а Петрова – его заместителем*),

- при наличии отрицания у целевого слова.

Конструкции с модальными глаголами с учётом определения текстовой ситуации учитываются и размечаются.

Итогом извлечения информации о текстовой ситуации должен стать набор отношений, связывающих ситуацию, обозначенную в тексте словом-маркером, и её участников, выраженных в тексте именными группами. Рассмотрим пример:

Президент Украины Виктор Янукович отправил в отставку премьер-министра Николая Азарова.

В результате работы модуля извлечения информации о ситуации должен получиться следующий набор отношений (см. рис. 1)

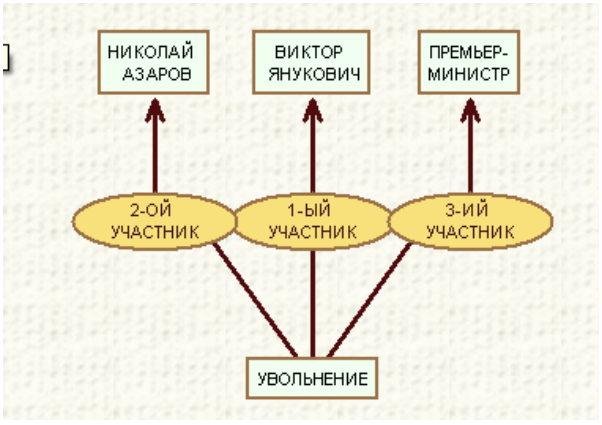


Рис.1. Представление информации, полученное в результате работы модуля анализа ситуаций

Какой уже извлечённой информацией мы можем располагать перед началом работы модуля извлечения ситуаций? Про каждое слово в предложении известна его морфологические характеристики, с помощью ресурса знаний системы ИСИДА-Т[6],[8] и специального модуля извлечения имён построены специальные аннотации для имён людей (включая разбиение на имя, отчество и фамилию), названий должностей, организаций, геополитических единиц. Отдельными аннотациями помечаются временные указатели (включая составные – например, *вчера вечером, в следующем году* и т.п.). Возможности синтаксического анализатора системы ИСИДА-Т ограничиваются анализом именных групп с зависимыми прилагательными или существительными в родительном падеже, а также с приложениями, включая предложные группы, в которые могут входить более простые именные группы. Для каждого слово известны его графематические характеристики (написано с большой или маленькой буквы, латинский или русский шрифт – вся информация о написании слова в тексте.)

Основная идея описываемого подхода – извлечь всё, что возможно извлечь из текста средствами локального микросинтаксиса и информацией об извлечённых сущностях.

Есть “ядерные” вещи, которые можно классифицировать и систематизировать, а есть словарно-текстовые, которые надо будет задать списком. Конечно, есть контексты с текстовыми ситуациями, которые средствами системы ИСИДА-Т не могут быть обработаны. Например, такие:

1. *им стал долгое время проработавший руководителем клиники, в Москве, заведующий кафедры ортопедической стоматологии факультета после-дипломного образования Московского государственного медико-стомато-логического университета им. А.И.Ев-докимова, доктор медицинских наук, профессор, человек, который знает дагестанскую и современную мировую медицину — Танка Ибрагимов* – анализ именной группы, которая называет участника ситуации получения должности, выходит за рамки возможностей синтаксического анализатора системы ИСИДА-Т.
2. *Президент Украины Виктор Янукович сменил главу Центрального управления Службы безопасности страны, назначив на этот пост человека, якобы близкого к своему сыну* – в данном примере участник отделён от слова-ситуации НАЗНАЧИВ синтаксическими группами, которые не являются участниками ситуации.

В разметке участвовали следующие ситуации:

@увольнение, @назначение,
@получение_должности, @уход_с_поста.

Аннотация разметки, атрибуты которой сравниваются с результатами, полученными в ходе обработки текста, содержит следующие атрибуты:

1. **Situation** – название ситуации
2. **first** – первый участник (при разметке сюда записывается главное слово именной группы, соответствующей по значению данному участнику ситуации);
3. **second** – второй участник;
4. **third** – третий участник.

Если в текстовой ситуации нет информации о каком-либо из участников, соответствующий атрибут остаётся незаполненным.

См. на рисунке 2 пример аннотации разметки:

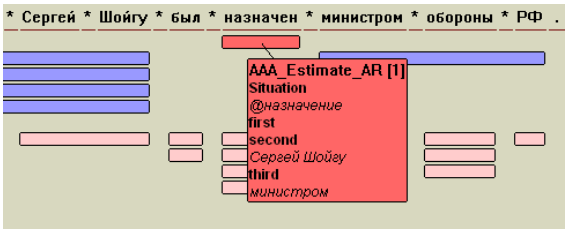


Рис.2. Пример аннотации разметки.

Одной из основных задач при разметке и тестировании коллекции текстов было оценить, какой результат мы сможем получить при существующих возможностях системы для текстовых ситуаций. Принципы разметки ситуаций для тестирования примерно соотвествуют принципам разметки событий ACE [13] в том, что помечаются ключевые слова и они могут быть выражены не только глаголом в личной форме, но и отглагольным существительным, причастием, деепричастием, помечаются целиком участники. Отличия же заключаются в том, что не размечаются ситуации, обозначенные местоимением (например, *она (отставка) произошла сегодня утром*).

3 Классификация слов-маркеров ситуации

Ниже приводятся слова, которые маркируют ситуации, участвующие в разметке и тестировании:

Ситуация	**Слова-маркеры**
Увольнение	увольнение, уволить, увольнять снятие, снять, снимать освобождение, освободить, освобождать отстранение, отстранить, отстранять отправить, отправлять (в отставку) прекратить, прекращать (полномочия)
Назначение	назначение, назначить, назначать Переназначение, переназначить, переназначать ставить поставить утверждение, утвердить, утверждать
Получение_должности	стать, становиться сохранить занять получение, получить, получать назначаться приход, придти, приходить (на должность) Быть (формы будущего времени)
Уход_в_отставку	уход, уйти, уходить покинуть, покидать оставить, оставлять лишиться, лишаться подать, подавать (в отставку) сложить, складывать (полномочия)

В следующей таблице приведена классификация контекстов со словом-маркером ситуации в зависимости от формы этого слова:

	Форма целевого слова	**Образец (на примере глагола увольнять)**
1	Глагол в личной форме (в единственном числе)	Уволил, увольняет и т.п.
2	Возвратный глагол	Уволился
3	Глагол в 3 лице мн.числа	Уволили, увольняют
4	Глагол в инфинитиве	Уволить (например, в приказе или с модальным глаголом)
5	Причастие действительное	Уволивший, увольняющий (оч. редко)
6	Причастие страдательное	Уволенный, увольняемый
7	Отглагольное существительное	увольнение
8	Деепричастие	Уволив, увольняя
9	Относительное предложение	Иванов, который уволил

4 Классификация участников ситуации

Прежде всего, нужно определить, какие именные группы будут считаться участниками ситуации. Помимо именных групп, которые задают непосредственно участников ситуаций, - это

именные группы, которые могут находиться между целевым словом и именной группой основного участника. В свою очередь такие группы делятся на **вспомогательные** (они нужны для того, чтобы целевое слово указывало на ситуацию, например, глагол **уйти** обозначает ситуацию @уход_в_отставку только при наличии зависимой именной группы с главным словом ОТСТАВКА) и **дополнительные** (например, указание на время, место, организацию – такая информация не относится к основным участникам, но при этом такие именные группы могут отделять группу основного участника от целевого слова). Кроме того, основные участники делятся на **атомарные** и **неатомарные** группы. Атомарные именные группы указывают на одного основного участника ситуации, а неатомарные – на двух. Например:

1. *Директор предприятия Виктор Сергеев уволил своего заместителя Романа Николаева.*
2. *Сегодня был уволен заместитель директора Роман Николаев.*

В рассматриваемых примерах 2-й (кого уволили) и 3-й (с какой должности) участники выражены в одной именной группе.

Итак, ниже приводится классификация участников ситуаций отставки-назначения:

I. Основные. Атомарные
1. Именные группы с главным словом ПОСТ, ДОЛЖНОСТЬ, КРЕСЛО, МЕСТО + название должности или + местоимение ЕГО, ЕЁ, СВОЙ, ЭТОТ;
2. Именная группа – имя человека;
3. Именная группа – название должности
4. Именные группы, устроенные особым образом и выражающие первого участника (решением президента, по распоряжению правительства и т.п.)

II. Основные. Неатомарные
1. Имя + должность (могут называть 2-го и 3-го участников), при этом должность может называть прежнее место работы

III. Вспомогательные. Именные группы – фиксированные выражения при ситуациях назначения-отставки (задаются списком) – *по собственному желанию, по собственной инициативе* и т.п.

IV. Дополнительные
1. Именные группы – указание на время
2. Именные предложные группы с указанием на организацию

Кроме общей классификации, отдельными пометами снабжаются такие слова в именных группах, которые помогают правильно распределить участников ситуации. Это, например, прилагательные - **бывший, новый, экс-, действующий.**

В следующих двух таблицах представлены слова-маркеры ситуаций и участники, наличие которых обязательно для формирования контекста ситуации (обязательные участники). Отбор таких слов проводился эмпирически на основе анализа текстов об отставках и назначениях на новостных порталах в сети Интернет. Конечно, такой список не может описывать всё языковое многообразие конструкций, описывающих целевые ситуации, но, безусловно, включает в себя самые частотные и типичные языковые выражения.

Ситуации с тремя участниками - @назначение, @увольнение

Целевое слово	**Обязательный участник**
Уволить, увольнять, увольнение	-
Снять, снимать, снятие	должность
Отправить, отправлять	отставка
прекратить	полномочия
Освободить, освобождать, освобождение	- должность
Отстранить, отстранять, отстранение	должность
Назначить, назначать, назначение, переназначить	Должность, организация или геополитическая единица
Поставить, ставить	Должность, организация или геополитическая единица
утвердить	должность

Ситуации с двумя участниками (первый никогда не выражен при данных целевых словах) – @получение_должности, @уход_в_отставку.

Целевое слово	Обязательный участник
Стать, становиться	должность
Сохранить, сохранять, сохранение	должность
Получить, получать, получение	должность
Быть	должность
назначаться	должность
Уходить, уйти, уход	Должность или отставка
Покинуть, покидать	Должность
Оставить, оставлять	Должность
Подать, подавать	Отставка
Сложить, сложение	полномочия
Лишиться, лишение	Должность или полномочия
уволиться	-

5 Описание алгоритма извлечения информации о ситуациях

Именные группы участников ситуаций собираются последовательно справа и слева от целевого слова. Работа правил организована следующим образом: правила собраны в фазы, которые выполняются последовательно одна за другой. Всего есть 9 фаз правил. Информация о работе фаз приведена в таблице ниже:

№	Описание работы фазы	Аннотации, которые строятся правилами
1	Словам, описывающим ситуацию, ставятся в соответствие специальные аннотации, в атрибуты которых в следующих фазах будет записываться информация об участниках. Эти аннотации также играют роль маркера для фаз, в которых происходит поиск потенциальных участников ситуации	**AVerb** – аннотация, маркирующая слово-ситуацию
2	Построение аннотаций для потенциальных участников, выраженных нестандартными конструкциями	**UKAZ** – аннотация для именных групп с главным словом “указ”, “решение”, “распоряжение” и т.п. **POST** – аннотация для именных групп с главным словом “пост”, “должность”, “место”, “кресло”.
3	Построение аннотаций для ключевых слов – 2-й этап	**AVerb** – аннотация, маркирующая ситуацию
4	Построение аннотаций для участников ситуаций, расположенных непосредственно справа и слева от целевого слова	**AALink** – аннотации для именных групп – потенциальных участников ситуации, которые выделяются правилами фазы непосредственно справа и слева от аннотации **AVerb**
5	Построение аннотаций для участников ситуаций, расположенных справа или слева от участников, определённых в фазе 4	**AALink** – аннотации для именных групп – потенциальных участников ситуации, которые выделяются справа и слева от аннотаций **AALink**, построенных правилами фазы 4 (эти аннотации включают в себя аннотации **AALink**, построенные ранее)
6	Построение аннотаций для участников ситуаций,	**AALink** – аннотации для именных групп – потенциальных

	расположенных справа или слева от участников, определённых в фазе 5	участников ситуации, которые выделяются справа и слева от аннотаций **AALink**, построенных правилами фазы 5 (эти аннотации включают в себя аннотации **AALink**, построенные ранее)
7	Добавление в аннотации **AVerb** атрибутов, соответствующих найденным потенциальным участникам (обязательные участники)	Аннотация **AVerb** получает новые атрибуты
8	Добавление в аннотацию AVerb атрибутов, соответствующих найденным потенциальным участникам (только для тех целевых слов, у которых на предыдущем этапе был опознан обязательный участник или обязательный участник не нужен)	Аннотация **AVerb** получает новые атрибуты
9	Определение отношений между целевым словом и найденным участником ситуации (первый участник, второй участник, третий участник)	**ALink** – аннотации, которые задают отношение между словом, маркирующим ситуацию, и участником, – приписывание номера участнику ситуации

Рассмотрим работу алгоритма на примере

Советником Шойгу стала телеведущая Мария Китаева.

Эталон разметки выглядит так:

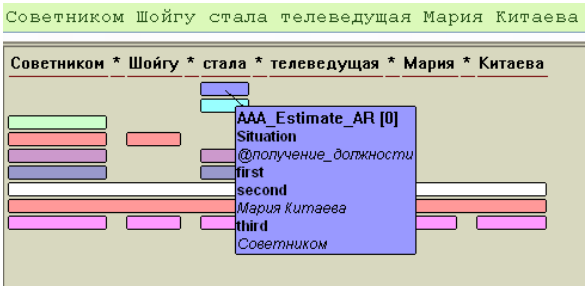


Рис. 3 Эталон разметки для разбираемой ситуации.

Результат работы фазы 1:

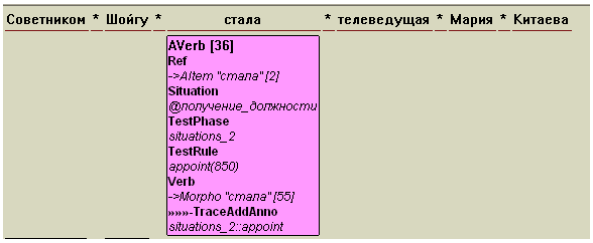


Рис.4. Построение аннотации AVerb, соответствующей слову-маркеру ситуации.

Построена аннотация AVerb, в атрибутах которой есть ссылка на ситуацию, которую маркирует эта аннотация (в данном примере - @получение должности), есть атрибут, по ссылке из которого можно попасть в аннотацию, где хранится информация обо всех морфологических характеристиках слова.

Следующий рисунок – результат работы фазы 4 (правила фаз 2 и 3 в данном контексте не работают)

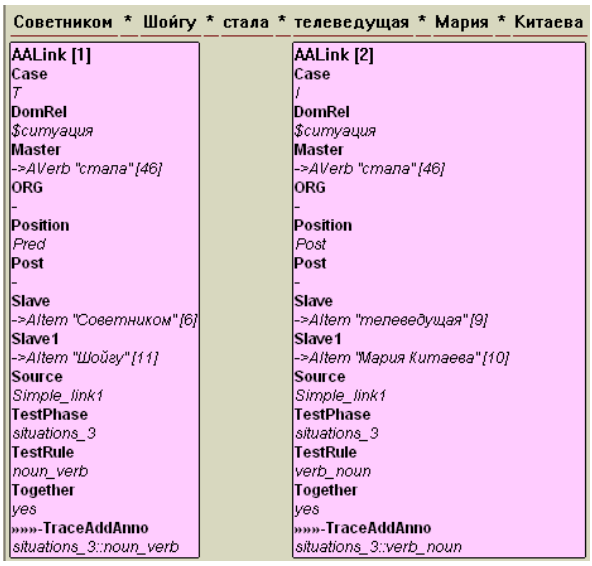


Рис.5. Построение аннотаций AALink, соответствующих именным группам участников ситуации.

В результате работы правил 4-й фазы построились две аннотации класса AALink – соответственно слева и справа от слова-маркера ситуации. Эти аннотации соотносятся с именными группами “Советником Шойгу” и “телеведущая Мария Китаева”.

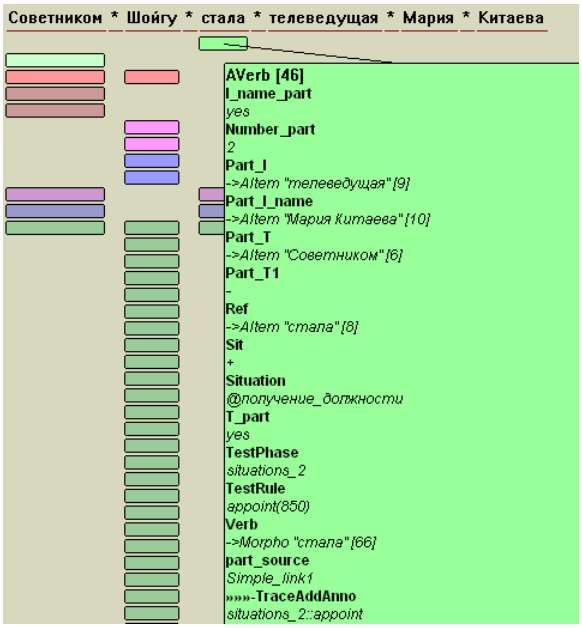


Рис.6. Добавление атрибутов в аннотацию AVerb.

Рисунок 6 иллюстрирует результат работы правил фаз 7 и 8, когда в аннотацию слова-маркера ситуации добавляются атрибуты-ссылки на найденные именные группы потенциальных участников. При этом названия атрибутов отражают падеж и некоторые другие важные для дальнейшей работы характеристики. В разбираемом примере - это Part_I (“телеведущая”), Part_I_name (“Мария Китаева”), Part_T(“Советником”).

На следующих двух рисунках показаны аннотации ALink, которые построены правилами последней, 9-й, фазы и соответствуют отношениям между словом-маркером ситуации и словом из именной группы, указывающим непосредственно на участника ситуации. Атрибут DomRel показывает номер участника ситуации, атрибут Situation – название ситуации, атрибуты Master и Slave – это ссылки на слово-маркер ситуации и на слово, обозначающее участника.

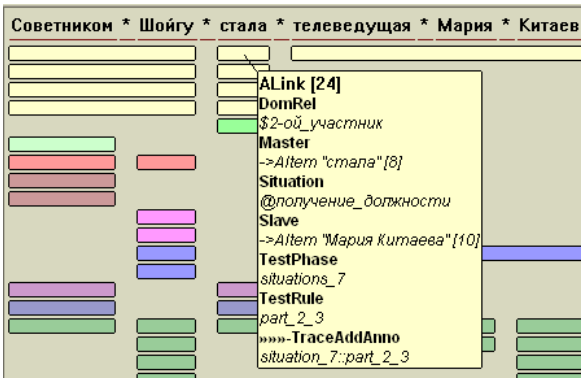


Рис.7. Построение аннотации ALink, моделирующей отношение между словом-маркером ситуации и её 2-м участником.

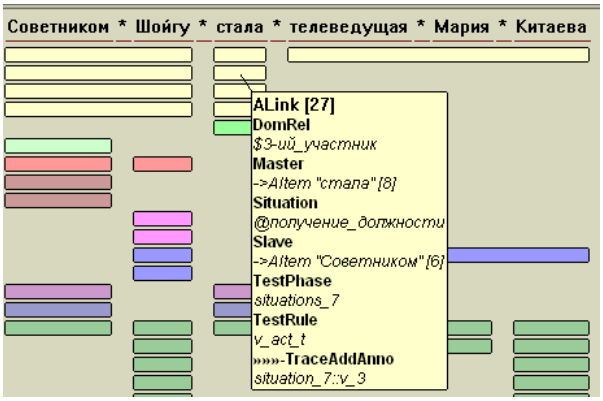


Рис.8. Построение аннотации ALink, моделирующей отношение между словом-маркером ситуации и её 3-м участником.

Наконец, на рисунке 9 приведён результат работы алгоритма извлечения информации о ситуации:

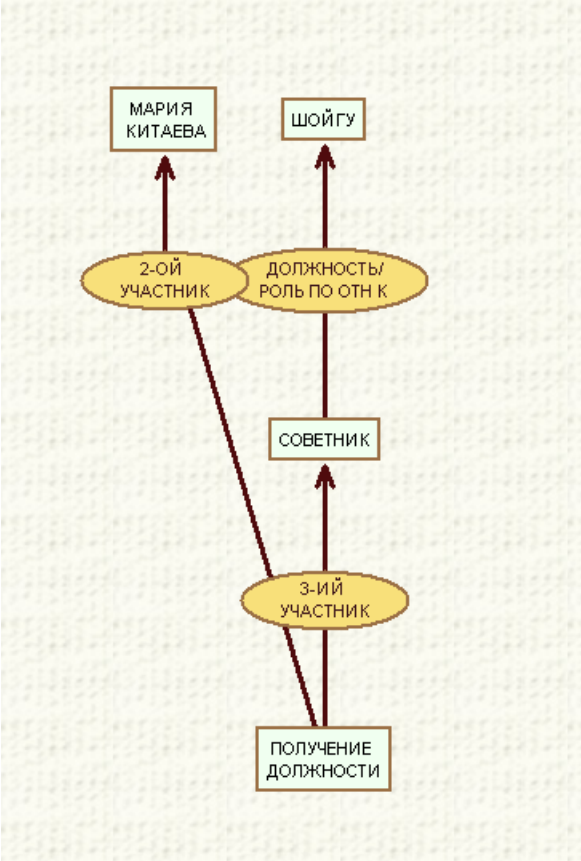


Рис.9. Графическое представление результатов извлечения.

6 Результаты тестирования работы алгоритма на размеченной коллекции.

Для проверки эффективности работы системы была размечена тестовая коллекция из 231 текста (тексты взяты на новостных порталах в сети Интернет). В текстах встретились и были размечены описанным выше способом 868 ситуаций отставки-назначения. Результаты тестирования приведены в таблице ниже:

Точность	0,76
Полнота	0,72
F-мера	0,74

При оценке результатов необходимо иметь в виду, что данные по всем трём оценкам приводятся по ситуациям, у которых при анализе атрибуты полностью совпали с атрибутами эталона.

По всей видимости, данные результаты отражают технологический предел возможностей системы ИСИДА-Т на данном этапе разработки. На результат анализа ситуаций оказывают влияние эффективность работы алгоритма выявления имён (F-мера – 0,95), а также особенности синтаксических конструкций, задающих ситуации.

Заключение

В результате работы с размеченной коллекцией были получены числовые данные, позволяющие оценить эффективность разработанного подхода к извлечению информации о ситуациях отставки-назначения. На данном этапе этот результат позволяет надеяться, что при развитии разработок в направлении анализа всего текста и разрешения кореферентности будут улучшены показатели извлечения информации о ситуациях и мы сможем в большей степени приблизить наши знания, полученные из текста с помощью автоматического извлечения, к знаниям, которые получает читающий человек.

Работа выполнена при поддержке РФФИ, грант № 13-06-00483а.

Литература

[1] Александровский Д.А., Кормалев Д.А., Кормалева М.С., Куршев Е.П., Сулейманова Е.А., Трофимов И.В. Развитие средств аналитической обработки текста в системе ИСИДА-Т // Тр. Десятой нац. конф. по искусственному интеллекту с междунар. участием КИИ-2006, Обнинск, 25-28 сентября 2006 г.: В 3 т. — М.: Физматлит, 2006. — Т. 2. — С. 555—563.

[2] Гершензон Л. М., Ножов И. М., Панкратов Д. В. Система извлечения и поиска структурированной информации из больших текстовых массивов СМИ. Архитектурные и лингвистические особенности. Компьютерная лингвистика и интеллектуальные технологии: Труды международной конференции «Диалог’2005» (Звенигород, 1–6 июня, 2005 г.)/ Под ред. И. М. Кобозевой,А. С. Нариньяни, В. П. Селегея. — М.: Наука, 2005.

[3] Ермаков А.Е. Автоматическое извлечение фактов из текстов досье: опыт установления анафорических связей. Компьютерная лингвистика и интеллектуальные технологии: труды Международной конференции "Диалог 2007". – Москва, Наука, 2007

[4] Ермаков А.Е. Извлечение знаний из текста и их обработка: состояние и перспективы. Информационные технологии 2009, № 7

[5] Киселев С.Л., Ермаков А.Е., Плешко В.В. Поиск фактов в тексте естественного языка на основе сетевых описаний. Компьютерная лингвистика и интеллектуальные технологии: труды Международной конференции Диалог’2004. — М.: Наука, 2004.

[6] Кормалев Д.А., Куршев Е.П., Сулейманова Е.А., Трофимов И.В. Архитектура

инструментальных средств систем извлечения информации из текстов. Труды международной конференции "Программные системы: теория и приложения", Переславль-Залесский, М.: Физматлит, 2004, т.2, с.49—70

[7] Кормалев Д.А., Куршев Е.П., Сулейманова Е.А., Трофимов И.В. Извлечение данных из текста. Анализ ситуаций ньюсмейкинга. Труды Восьмой национальной конференции по искусственному интеллекту с международным участием КИИ-2002. Москва, Физматлит, 2002, с. 199-206

[8] Кормалев Д.А., Куршев Е.П., Сулейманова Е.А., Трофимов И.В.. Технология извлечения информации, из текстов, основанная на знаниях. Программные продукты и системы, 2009, №2

[9] Куршев Е.П., Сулейманова Е.А. Ресурсы предметных знаний в системах интеллектуального анализа текста // Тр. междунар. конф. «Программные системы: теория и приложения», ИПС РАН, Переславль-Залесский, октябрь 2006 г.: В 2 т. — М.: Физматлит, 2006. — Т.1. — С. 379—390.

[10] http://www.mlg.ru/

Extracting information on appointments and dismissals from news texts. An experience in developing an annotated corpus. Testing results.

Natalia Vlasova

The paper describes an experiment on annotating a collection of Russian-language news texts for an information extraction task. The objective was to evaluate the efficiency of an approach to solving the appointment-dismissal task, which is implemented in the ISIDA-T software. ISIDA-T has been developed during the past decade in the Program Systems Institute, RAS. It is based on a knowledge-engineering approach to information extraction. The paper describes the implementation of information extraction method, the annotation principles, the test collection, and presents some evaluation results.

[11] Власова Н.А. Подход к автоматическому извлечению информации о назначениях и отставках лиц (на материале новостных сообщений) // Электронные библиотеки: перспективные методы и технологии, электронные коллекции. XIV Всероссийская научная конференция RCDL-2012. Труды конференции. — Переславль-Залесский : Университет города Переславля, 2012. — С. 374—378.

[12] Котельников Д.С., Лукашевич Н.В. Итерационное извлечение шаблонов описания событий по новостным кластерам Электронные библиотеки: перспективные методы и технологии, электронные коллекции. XIV Всероссийская научная конференция RCDL-2012. Труды конференции. — Переславль-Залесский : Университет города Переславля, 2012. — С. 362—373.

[13] http://projects.ldc.upenn.edu/ace/docs/ - электронный документ – принципы разметки ACE

Использование тематических моделей в извлечении однословных терминов

© М. А. Нокель
МГУ им. М. В. Ломоносова, Москва
mnokel@gmail.com

© Н. В. Лукашевич
НИВЦ МГУ им. М. В. Ломоносова, Москва
louk_nat@mail.ru

Аннотация

В статье представлены результаты экспериментов по применению тематических моделей к задаче извлечения однословных терминов. В качестве текстовых коллекций была взята подборка статей из электронных банковских журналов на русском языке и англоязычная часть корпуса параллельных текстов Europarl. Эксперименты показывают, что использование тематической информации значительно улучшает качество извлечения однословных терминов независимо от предметной области и используемого языка.

Ключевые слова

Тематические модели, Кластеризация, Извлечение однословных терминов

1 Введение

Извлечение терминов из текстов определённой предметной области играет значительную роль во многих прикладных задачах, в первую очередь – в разработке и пополнении различных терминологических ресурсов, таких как тезаурусы и онтологии [35]. Поскольку разработка таких ресурсов вручную достаточно трудоёмка, за последние годы было проведено большое количество исследований по автоматизации данного процесса.

Большинство современных методов извлечения терминов основываются на использовании различных статистических и лингвистических признаков слов. Основная цель при этом заключается в получении упорядоченного списка кандидатов в термины, в начале которого находится как можно больше слов, с наибольшей вероятностью являющихся терминами. В некоторых работах было экспериментально установлено, что использование машинного обучения для комбинирования признаков значительно улучшает результаты извлечения терминов по сравнению с методами, основанными только на одном каком-то признаке, поскольку те или иные признаки только частично отражают особенности поведения терминов в текстах [17].

На текущий момент традиционно используемые для извлечения терминов статистические признаки никак не отражают тот факт, что большинство терминов относятся к той или иной подтеме предметной области. Поэтому нами было сделано предположение, что выделение таких подтем в коллекции текстов способно улучшить качество автоматического извлечения терминов.Для проверки этого предположения в статье будут рассмотрены различные методы выделения подтем в коллекции текстов, которые часто в литературе называются статистическими тематическими моделями [4].

Некоторые виды статистических тематических моделей могут основываться на традиционных методах автоматической кластеризации текстов [12]. В последнее время предложены вероятностные механизмы выделения подтем в текстовых коллекциях такие, как методы, основанные на скрытом распределении Дирихле (Latent Dirichlet allocation [4]), которые собственно и были названы тематическими моделями и в настоящее время интенсивно исследуются в рамках различных приложениях автоматической обработки текстов ([12], [29], [3]).

Основная задача данной статьи заключается в исследовании возможности использования тематической информации для повышения качества извлечения однословных терминов. Для этой цели вначале в текстовой коллекции выделяются подтемы, затем к ним применяются некоторые модификации хорошо известных признаков, которые впоследствии используются вместе с другими статистическими и лингвистическими признаками.

Для того чтобы результаты, представленные в статье, не зависели ни от предметной области, ни от языка, были взяты две предметные области и соответствующие текстовые коллекции: банковская предметная область и тексты банковской тематики на русском языке и широкая предметная область современной общественной жизни Европы и речи с заседаний Европарламента на английском языке. При этом эксперименты будут строиться следующим образом:

1. Вначале статистические тематические модели будут исследованы с точки зрения задачи

извлечения однословных терминов с целью выбора наилучшей;

2. Затем будет осуществлено сравнение признаков, посчитанных для лучшей тематической модели, с остальными признаками с целью определения вклада, который даёт использование тематической модели в рассматриваемой задаче.

2 Близкие работы

За последние годы было предложено много различных статистических и лингвистических признаков слов, используемых для извлечения однословных терминов из коллекции текстов определённой предметной области ([6], [1], [20], [10] и др.).

Все предложенные признаки можно разделить на следующие группы:

1. *Признаки, основанные на частотности слов-кандидатов.* К этой группе относится, например, признак *TFRIDF*, предложенный в работе [6] и использующий модель Пуассона для предсказания терминологичности слов;

2. *Признаки, использующие контрастную коллекцию*, т.е. коллекцию более общей тематики. Одним из наиболее характерных представителей данной группы является широко используемый на практике признак *Относительная частотность* [1], основанный на сравнении относительных частотностей слов в рассматриваемой и в контрастной текстовой коллекциях;

3. *Контекстные признаки*, соединяющие в себе информацию о частотности слов-кандидатов с данными о контексте их употребления. Наиболее известными представителями этой группы являются признаки *C-Value* [20] и *NC-Value* [10], учитывающие частоту встречаемости объемлющего словосочетания для кандидата в термины.

Однако ни один из предложенных признаков не является определяющим [25], и фактически из текстов извлекается довольно большой список слов-кандидатов, которые затем должны быть проанализированы и подтверждены экспертом по предметной области. Важно поэтому дополнять список используемых признаков, что позволит получать в начале списка как можно больше слов, с наибольшей вероятностью являющихся терминами.

3 Статистические тематические модели

Новые признаки слов-кандидатов, которые вводятся в данной статье, используют информацию, получаемую статистическими тематическими моделями в исследуемых текстовых коллекциях.

Статистическая тематическая модель (далее – тематическая модель) коллекции текстовых документов на основе статистических методов определяет, к каким подтемам относится каждый документ и какие слова образуют каждую подтему, представляющую собой список часто встречающихся рядом друг с другом слов, упорядоченный по убыванию степени принадлежности ему [34]. Так, в таблице 1 представлены первые десять слов, наиболее полно характеризующие три случайно выбранных подтемы, выделенных из русскоязычных текстов банковской тематики рассматриваемой коллекции.

Подтема 1	**Подтема 2**	**Подтема 3**
Банкнота	*Обучение*	*Германия*
Офшорный	*Студент*	*Франция*
Счетчик	*Учебный*	*Евро*
Купюра	*Вуз*	*Европейский*
Подделка	*Семинар*	*Польша*
Обращение	*Образование*	*Европа*
Номинал	*Знание*	*Чехия*
Монета	*Специалист*	*Италия*
Подлинность	*Слушатель*	*Немецкий*
Поддельный	*Учитель*	*Французский*

Таблица 1: Примеры подтем

В тематических моделях, как правило, используется модель мешка слов, в которой каждый документ рассматривается как набор встречающихся в нём слов. При этом перед выделением подтем текстовая коллекция обычно подвергается предобработке, выделяющей только значимые слова в каждом документе. В частности, в данном исследовании для русского языка были отобраны только существительные и прилагательные, а для английского – только существительные, поскольку они покрывают большую часть терминов.

На сегодняшний день разработано достаточно много различных тематических моделей. Для выбора моделей для исследования были проанализированы предыдущие работы, в которых осуществляется сравнение моделей с точки зрения различных практических приложений. Так, в работе [29] утверждается, что каждая тематическая модель имеет свои сильные и слабые стороны. Сравнивая между собой методы NMF (неотрицательной матричной факторизации) и LDA (латентного размещения Дирихле), авторы приходят к выводу, что оба этих алгоритма дают похожее ка-

чество, хотя NMF и выдаёт немного больше бессвязных подтем. В работе [12] утверждается, что традиционные тематические модели показывают приемлемое качество выделения подтем, но имеют множество ограничений. В частности они предполагают, что каждый документ имеет только одну тематику. В действительности же документы представляют собой, как правило, смесь подтем. Кроме того, авторы отмечают, что параметры традиционных моделей достаточно сложно настраивать. В то же время в работе подчёркивается, что более сложные модели (такие как LDA) необязательно дадут лучшие результаты.

Поскольку, как следует из упомянутых выше работ, среди тематических моделей нет явного лидера и непонятно, какое качество они покажут в рассматриваемой задаче извлечения однословных терминов, было решено выбрать несколько наиболее характерных представителей, которых условно можно отнести либо к вероятностным, либо к методам кластеризации текстов, рассматриваемых с точки зрения тематических моделей. Каждая из выбранных моделей будет рассмотрена в следующих подразделах.

3.1 Тематические модели, основанные на методах кластеризации текстов

Традиционные тематические модели, как правило, основываются на методах жёсткой кластеризации, рассматривающих каждый документ как разреженный вектор в пространстве слов большой размерности [28]. После окончания работы алгоритма кластеризации каждый получившийся кластер рассматривается как один большой документ для вычисления вероятностей входящих в него слов по следующей формуле:

$$P(w|t) = \frac{TF(w|t)}{\sum\limits_{w} TF(w|t)} \tag{1}$$

где $TF(w|t)$ – частотность слова w в кластере t.

В процессе кластеризации текстовых документов можно выделить следующие общие шаги:

1. Предобработка документов (фильтрация слов);
2. Преобразование документа во внутреннее представление (в вектор слов);
3. Расчёт расстояния между документами на основе внутреннего представления;
4. Кластеризация документов на основе рассчитанного расстояния с помощью одного из алгоритмов.

Для численной оценки расстояния между документами необходим способ определения значимости каждого слова в обособлении одного документа относительно другого. Для этого были предложены различные схемы взвешивания отдельных слов, наиболее распространённой из которых является схема TFIDF [19], которая также была включена в данное исследование. В ней каждому слову в документе ставится в соответствие величина, вычисляемая по следующей формуле:

$$TFIDF(w|d) = TF(w|d) * \max\left(0, \log\frac{N - DF(w)}{DF(w)}\right) \tag{2}$$

где N – общее число документов в коллекции, $DF(w)$ – число документов в коллекции, в которых встречается слово w.

В следующих разделах будут описаны выбранные нами методы построения традиционных тематических моделей.

3.1.1 K-Средних и Сферический K-Средних

Алгоритм K-Средних [18] начинает свою работу со случайной инициализации центров масс каждого кластера. Далее он итеративно повторяет следующие шаги:

1. Все документы разбиваются на кластеры в соответствии с тем, какой из центров масс оказался ближе по выбранной метрике;
2. Для каждого полученного кластера пересчитывается центр масс.

В качестве метрики близости между двумя документами исследовались следующие:

- Евклидово расстояние (*K-Means*) [18]:

$$sim(A, B) = \sqrt{\sum_i (A_i - B_i)^2} \tag{3}$$

- Косинусная мера близости *(сферический k-средних – SPK-Means)*. При этом все векторы, представляющие документы, нормализуются к единичной гиперсфере [33]:

$$sim(A, B) = \frac{\sum\limits_i (A_i \times B_i)}{\sqrt{\sum\limits_i A_i} \times \sqrt{\sum\limits_i B_i}} \tag{4}$$

3.1.2 Иерархическая агломеративная кластеризация

Алгоритм иерархической агломеративной кластеризации [14] изначально рассматривает каждый документ как отдельный кластер. Затем он итеративно повторяет следующие шаги:

1. Находятся и объединяются в новый кластер два наиболее близких кластера;
2. Пересчитываются расстояния между новым кластером и всеми остальными.

Процесс повторяется до тех пор, пока не останется заданное число кластеров.

В качестве способов определения наиболее близких кластеров исследовались следующие наиболее распространённые [14]:

- *Complete-link* ("полное связывание"). Наиболее близкие кластеры – это кластеры с наименьшим максимальным парным расстоянием между документами;
- *Single-link* ("одиночное связывание"). Наиболее близкие кластеры – это кластеры с наименьшим минимальным парным расстоянием между документами;
- *Average-link* ("среднее связывание"). Это компромисс между двумя предыдущими способами. Наиболее близкие кластеры – это кластеры с наименьшим средним парным расстоянием между документами.

3.1.3 Неотрицательная матричная факторизация (NMF)

Алгоритм NMF, изначально разработанный для уменьшения размерности, зарекомендовал себя для решения задач кластеризации [32]. Данный алгоритм осуществляет нечёткую кластеризацию, которая относит один и тот же документ к разным кластерам с разными вероятностями.

Принимая на входе неотрицательную разреженную матрицу V, которая получается записыванием векторов, представляющих документы, по столбцам, алгоритм ищет такие матрицы W и H меньшей размерности, что $V \approx WH$ по некоторой метрике. В качестве такой метрики исследовались следующие [16]:

- Евклидово расстояние (*NMF Euc*):

$$||A - B||^2 = \sum_{i,j}(A_{ij} - B_{ij})^2 \qquad (5)$$

- Расстояние Кульбака-Лейблера для неотрицательных матриц (*NMF KL*):

$$D(A||B) = \sum_{i,j}(A_{ij} \log \frac{A_{ij}}{B_{ij}} - A_{ij} + B_{ij}) \quad (6)$$

В результате работы алгоритма в матрице W получается распределение слов по кластерам, а в матрице H – распределение документов по кластерам. Нормируя соответствующие величины для каждого слова/документа, можно получить вероятности принадлежности этого слова/документа кластеру.

3.2 Вероятностные тематические модели

Вероятностные тематические модели представляют каждый документ в виде смеси подтем, в которой каждая подтема представляет собой некоторое вероятностное распределение над словами. Вероятностные модели порождают слова по следующему правилу:

$$P(w|d) = \sum_t P(w|t)P(t|d) \qquad (7)$$

где $P(t|d)$ и $P(w|t)$ – распределения подтем по документам и слов по подтемам, а $P(w|d)$ – наблюдаемое распределение слов по документам.

Порождение слов происходит следующим образом. Для каждого документа d и для каждого слова $w \in d$ выбирается тема t из распределения $P(t|d)$, и затем генерируется слово w из распределения $P(w|t)$.

Самыми известными представителями данной категории являются метод вероятностного латентного семантического индексирования (PLSI) и латентное размещение Дирихле (LDA).

3.2.1 PLSI

Метод PLSI, также известный как PLSA, был предложен в работе [13]. Данный метод моделирует матрицу V, в которой V_{ij} обозначает число вхождений слова w_i в документ d_j, получающуюся из модели с k подтемами:

$$P(w_i, d_j) = \sum_{t=1}^{k} P(t)P(d_j|t)P(w_i|t) \qquad (8)$$

Параметры модели настраиваются с помощью максимизации правдоподобия наблюдаемых данных из матрицы M, т.е. максимизируя следующий функционал:

$$\sum_{i,j} TF(w_i|d_j) \log P(w_i, d_j) \to \max \qquad (9)$$

Поскольку в статье [7] теоретически обосновано, что алгоритм NMF, минимизирующий расстояние Кульбака-Лейблера и рассмотренный в прошлом разделе, эквивалентен алгоритму PLSA, в данном исследовании метод PLSA не рассматривается отдельно.

3.2.2 LDA

Метод латентного размещения Дирихле был предложен в работе [4]. LDA расширяет модель PLSI, добавляя туда априорное распределение параметров модели ($P(w|t)$ и $P(t|d)$), считая их распределёнными по закону Дирихле.

Для настройки параметров модели необходим Байесовский вывод. Однако, поскольку он алгоритмически неразрешим [4], исследовались следующие два применяемых на практике приближённых способа Байесовского вывода:

- *LDA VB* – вариационный Байесовский вывод, описанный в статье [4];
- *LDA Gibbs* – метод Монте-Карло с марковскими цепями, использующий сэмплирование Гиббса [27].

3.3 Базовая тематическая модель

В качестве baseline была взята "тематическую" модель, которая не выделяет никаких подтем, а просто рассматривает каждый документ как отдельно взятую подтему. Данная модель будет использоваться нами в экспериментах для сравнения с другими методами.

4 Коллекции текстов для экспериментов

Во всех экспериментах, описываемых в данной статье, слова-кандидаты извлекались из двух различных коллекций:

- Коллекция банковских русскоязычных текстов (10422 документа, примерно 15.5 млн слов), взятых из различных электронных банковских журналов: Аудитор, Банки и Технологии, РБК и др.;
- Английская часть корпуса параллельных текстов Europarl [8] из заседаний Европарламента (9673 документа, примерно 54 млн слов).

Для подтверждения терминологичности слов-кандидатов использовались следующие "золотые стандарты":

- Для русского языка – тезаурус, разработанный вручную для Центрального Банка Российской Федерации и включающий в себя порядка 15000 терминов, относящихся к сфере банковской активности, денежной политики и макроэкономики;
- Для английского языка – официальный многопрофильный тезаурус Европейского Союза Eurovoc [9], предназначенный для ручного индексирования заседаний Европарламента. Его английская версия включает в себя 15161 термин.

При этом слово-кандидат считается термином, если оно содержится в тезаурусе.

Все признаки слов-кандидатов рассчитывались для 5000 самых частотных слов. В качестве метрики оценки качества была выбрана Средняя Точность (AvP) [19], определяемая для множества D всех слов-кандидатов и его подмножества $D_q \subseteq D$, представляющего действительно термины (т.е. подтверждённые тезаурусом):

$$AvP(n) = \frac{1}{|D_q|} \sum_{1 \le k \le |D_q|} \left(r_k \times \left(\frac{1}{k} \sum_{1 \le i \le k} r_i \right) \right) \quad (10)$$

где $r_i = 1$, если i-е слово-кандидат $\in D_q$, и $r_i = 0$ иначе. Данная формула отражает тот факт, что чем больше терминов сосредоточено в вершине итогового списка слов-кандидатов, тем выше мера средней точности.

Эксперименты проводились с разным числом выделяемых подтем: 50, 100 и 150 соответственно. Визуально результаты получались разными, но на качестве извлечения терминов это никак не отразилось. Поэтому все дальнейшие эксперименты проводилось с числом подтем, равным 100.

5 Выбор лучшей тематической модели

Как уже было сказано выше, вначале будут представлены результаты экспериментов по определению наилучшей тематической модели. Для этого будут предложены и посчитаны для каждой из рассмотренных выше тематических моделей некоторые модификации известных признаков слов.

5.1 Признаки, использующие тематическую информацию

Основной идеей всех признаков, использующих полученную с помощью какой-либо тематической модели информацию, является тот факт, что в начале списков, образующих подтемы, с большой вероятностью находятся термины. Для экспериментов мы предложили некоторые модификации известных признаков (см. таблицу 2). В таблице 2 используются следующие обозначения:

- $TF(w)$ – частотность слова w
- $DF(w)$ – документная частотность слова w
- $P(w|t)$ – условная вероятность принадлежности слова w подтеме t
- k – число топиков

5.2 Результаты экспериментов

В таблицах 3 и 4 представлены результаты экспериментов для исследуемых русского и английского корпуса соответственно.

Как видно из приведённых выше таблиц, лучшее качество независимо от языка и предметной области даёт тематическая модель **NMF**, минимизирующая расстояние Кульбака-Лейблера. Так, лучшим признаком для обоих языков является *Term Score* с 16% (соответственно 21%) прироста

Признак	Формула
Частотность (TF)	$\sum_t P(w\|t)$
TFIDF	$TF(w) \times \log \frac{k}{DF(w)}$
Domain Consensus (DC) [22]	$-\sum_t (P(w\|t) \times \log P(w\|t))$
Maximum TF	$\max_t P(w\|t)$
Term Score (TS) [3]	$\sum_t TS(w\|t)$ $TS(w\|t) = P(w\|t) \log \frac{P(w\|t)}{(\prod_t P(w\|t))^{\frac{1}{k}}}$
TS-IDF	$TS(w) \times \log \frac{k}{DF(w)}$
Maximum TS (MTS)	$\max_t TS(w\|t)$

Таблица 2: Признаки, использующие тематическую информацию

Модель	TF	TFIDF	DC	MTF	TS	TSIDF	MTS
K-Means	33.3	25.5	32.7	34.4	35.7	28.7	34.3
SPK-Means	35.5	27.2	35	33.9	36.3	30.1	33.6
Single-link	34.8	39.9	33.6	38.9	38.4	40.5	39
Comp-link	35.6	41	34.5	39.2	38.4	41	39.5
Average-link	35.8	40.7	34.5	39.5	39	40.9	39.6
NMF Euc	40.8	42.5	40.3	40.8	42	43.1	41.9
NMF KL	42.3	40.3	37.5	47.1	**48.9**	42.9	47.9
LDA VB	35.8	42.7	32.8	42.8	42.5	45.1	46.5
LDA Gibbs	37.7	38.4	35	46.2	42.6	42.8	47.2
Baseline	34	37.6	32.8	38.5	38.1	42	38.1

Таблица 3: Средняя точность признаков на русском корпусе

Model	TF	TFIDF	DC	MTF	TS	TSIDF	MTS
K-Means	29.3	32.3	28.9	30.3	30.1	31.8	30.4
SPK-Means	28.1	29.8	27.9	28.7	28.6	29.7	28.7
Single-link	30.3	38.9	29.8	37.3	36.5	38.8	39.9
Comp-link	31.1	39.6	30.4	37.2	34.6	38.9	39
Average-link	30.5	38.9	29.9	37.1	35.4	38.3	39.3
NMF Euc	34.4	31.6	32.3	41.1	43.7	31.6	40.5
NMF KL	33.3	37.7	31.2	44.3	**44.4**	37.3	44.1
LDA VB	32.3	30.3	30.5	37.1	36.3	30.3	38.5
LDA Gibbs	35.2	41.8	33.3	42.6	37.8	43.7	43.5
Baseline	31.5	32.8	30	36	33.6	35	36.7

Таблица 4: Средняя точность признаков на английском корпусе

качества относительно лучших признаков базовой модели (*TFIDF* для русского корпуса и *Maximum Term Score* для английского корпуса).

Помимо вычисления средней точности отдельных признаков было также осуществлено их комбинирование для каждой исследуемой тематической модели в отдельности с помощью метода логистической регрессии, реализованного в библиотеке Weka [30]. При этом проводилась четырёхкратная кросс-проверка, означающая, что вся исходная выборка разбивалась случайным образом на четыре равные непересекающиеся части, и каждая часть по очереди становилась контрольной подвыборкой, а обучение проводилось по остальным трём. Результаты комбинирования признаков для русского и английского корпусов представлены в таблице 5.

Как видно из приведённых выше таблиц, тематическая модель **NMF**, минимизирующая расстояние Кульбака-Лейблера, снова даёт наилучшее качество с 10% прироста для русского и с 23% прироста для английского корпусов относительно базовой тематической модели.

Модель	Средняя точность	
	Для русского корпуса	Для английского корпуса
Baseline	44.9	36.2
K-Means	36.2	33.7
SPK-Means	38.1	33.3
Single-link	42.1	41.4
Complete-link	41.9	41.3
Average-link	42.7	41.3
NMF Euc	43.4	43.8
NMF KL	**49.5**	**44.5**
LDA VB	46.1	36.7
LDA Gibbs	47.9	44.4

Таблица 5: Средняя точность комбинирования признаков, использующих тематическую информацию

Таким образом, наилучшей тематической моделью оказалась модель **NMF**, минимизирующая расстояние Кульбака-Лейблера.

6 Сравнение с другими признаками

Для изучения вклада тематической информации в задачу автоматического извлечения однословных терминов было решено сравнить результаты предложенных признаков, использующих тематическую информацию, с остальными статистическими и лингвистическими признаками для обоих исследуемых корпусов для 5000 самых частотных слов.

В качестве признаков, не использующих тематическую информацию, были взяты характерные представители групп, описанных в разделе 2.

6.1 Признаки, основанные на частотности

Признаки из данной группы опираются на предположение о том, что термины, как правило, встречаются в коллекции гораздо чаще остальных слов. В исследование были включены следующие признаки: *Частотность*, *Документная частотность*, *TFIDF* [19], *TFRIDF* [6], *Domain Consensus* [22].

6.2 Признаки, использующие контрастную коллекцию

Для вычисления признаков этой категории помимо целевой коллекции текстов предметной области использовалась контрастная коллекция текстов более общей тематики. Для русского языка в качестве таковой была взята подборка из примерно 1 миллиона новостных текстов, а для англий-

ского – n-граммные статистики из Британского Национального Корпуса [5].

Основная идея таких признаков заключается в том, что частотности терминов в целевой и контрастной коллекциях существенно различаются. В данном исследовании рассматривались следующие признаки: *Относительная частотность* [1], *Релевантность* [26], *TFIDF* [19] с вычислением документной частотности по контрастной коллекции, *Contrastive Weight* [2], *Discriminative Weight* [31], *KF-IDF* [15], *Lexical Cohesion* [24] и *Логарифм правдоподобия* [11].

6.3 Контекстные признаки

Контекстные признаки соединяют в себе информацию о частотности слов-кандидатов с данными о контексте их употребления в коллекции. В данном исследовании рассматривались следующие признаки: *C-Value* [20], *NC-Value*, *MNC-Value* [10], *Token-LR*, *Token-FLR*, *Type-LR*, *Type-FLR* [21], *Sum3*, *Sum10*, *Sum50*, *Insideness* [17].

6.4 Прочие признаки

В качестве остальных признаков, не использующих тематическую информацию, рассматривались номер позиции первого вхождения в документы, типы слов-кандидатов (существительное или прилагательное), слова-кандидаты, начинающиеся с заглавной буквы, и существительные в именительном падеже ("подлежащие") и слова из контекстного окна с некоторыми самыми частотными предопределёнными терминами [23].

Кроме этого, также рассматривались и комбинации данных признаков с некоторыми статистическими величинами (такими, как частотность в целевом корпусе). Всего было взято 28 таких признаков.

6.5 Результаты экспериментов

Лучшие признаки каждой из упомянутых выше групп для русского и английского корпусов приведены в таблицах 6 и 7.

Группа признаков	**Лучший признак**	**AvP**
Основанные на частотности	*TFRIDF*	41.1
Использующие контрастную коллекцию	*Логарифм правдоподобия*	36.9
Контекстные	*Sum3*	37.4
Тематические	*Term Score*	**48.9**

Таблица 6: Средняя точность лучших признаков для русского корпуса

Как видно из приведённых выше таблиц, независимо от языка и предметной области лучшими индивидуальными признаками оказались тематические, превзойдя остальные на 19% и 15% средней точности для русского и английского корпусов соответственно.

Группа признаков	**Лучший признак**	**AvP**
Основанные на частотности	*TFRIDF* для подлежащих	38.5
Использующие контрастную коллекцию	*TFIDF* для подлежащих	34.2
Контекстные	*C-Value*	31.3
Тематические	*Term Score*	**44.5**

Таблица 7: Средняя точность лучших признаков для английского корпуса

Для оценки же вклада тематических признаков в общую модель извлечения однословных терминов мы сравнили модель извлечения, учитывающую тематические признаки (7 baseline признаков и 7 признаков, посчитанных для наилучшей тематической модели NMF KL), и модель, не использующую их. Результаты сравнения для обоих рассматриваемых корпусов приведены в табл. 8 (комбинирование признаков осуществлялось с помощью логистической регрессии из библиотеки Weka [30]).

Корпус	**Средняя точность**	
	Без тематических признаков	С тематическими признаками
Русский	54.6	**56.3**
Английский	50.4	**51.4**

Таблица 8: Результаты сравнения моделей с тематическими признаками и без них

Мы считаем, что данные результаты, показанные на двух разных коллекциях, подтверждают, что тематические модели действительно вносят дополнительную информацию в процесс автоматического извлечения терминов.

В заключение в таблице 9 представлены первые 10 элементов из списков извлечённых слов-кандидатов, полученных с помощью моделей, учитывающих тематические признаки (при этом термины выделены курсивом).

7 Заключение

В статье представлены результаты экспериментального исследования возможности применения тематических моделей для улучшения качества автоматического извлечения однословных терминов.

Были исследованы различные тематические модели (как вероятностные, так и традиционные методы кластеризации) и предложены несколько модификаций известных признаков для упорядочивания слов-кандидатов по убыванию их терминологичности. В качестве текстовых коллекций бы-

№	Русский корпус	Английский корпус
1	*Банковский*	Member
2	*Банк*	Minute
3	Год	*Amendment*
4	*РФ*	*Document*
5	*Кредитный*	EU
6	*Налоговый*	President
7	*Кредит*	*People*
8	*Пенсионный*	*Directive*
9	Средство	Year
10	Клиент	Question

Таблица 9: Примеры извлечённых слов-кандидатов

ли взяты два различных корпуса: электронные банковские статьи на русском языке и речи с заседаний Европарламента на английском языке.

Эксперименты показали, что независимо от предметной области и языка использование тематической информации способно значительно улучшить качество автоматического извлечения однословных терминов.

Список литературы

[1] K. Ahmad, L. Gillam, L. Tostevin. University of Survey Participation in Trec8. Weirdness indexing for logical document extrapolation and retrieval. In the Proceedings of TREC 1999, 1999.

[2] R. Basili, A. Moschitti, M. Pazienza, F. Zanzotto. A Contrastive Approach to Term Extraction. In the Proceedings of the 4th Terminology and Artificial Intelligence Conference, 2001.

[3] D. Blei and J. Lafferty. Topic Models. Text Mining: Classification, Clustering and Applications, Chapman & Hall, pp. 71–89, 2009.

[4] D. Blei, A. Ng and M. Jordan. Latent Dirichlet Allocation. Journal of Machine Learning Research, No 3, pp. 993–1022, 2003.

[5] British National Corpus. `http://www.natcorp.ox.ac.uk/`

[6] K. Church and W. Gale. Inverse Document Frequency IDF. A Measure of Deviation from Poisson. In the Proceedings of the Third Workshop on Very Large Corpora. MIT Press, pp. 121–130, 1995.

[7] Chris Ding, Tao Li, Wei Peng. On the equivalence between Non-negative Matrix Factorization and Probabilistic Latent Semantic Indexing. Computational Statistics and Data Analysis, No 52, pp. 3913–3927, 2008.

[8] European Parliament Proceedings Parallel Corpus 1996–2011. `http://www.statmt.org/europarl/`

[9] EuroVoc. Multilingual Thesaurus of the European Union. `http://eurovoc.europa.eu/drupal/`

[10] K. Frantzi and S. Ananiadou. Automatic Term Recognition Using Contextual Cues. In the Proceedings of the IJCAI Workshop on Computational Terminology, pp. 29–35, 2002.

[11] A. Gelbukh, G. Sidorov, E. Lavin-Villa, L. Chanona-Hernandez. Automatic Term Extraction using Log-likelihood based Comparison with General Reference Corpora. In the Proceedings of the Natural Language Processing and Information Systems, pp. 248–255, 2010.

[12] Q. He, K. Chang, E. Lim, A. Banerjee. Keep It Smile with Time: A Reeximanation of Probabilistic Topic Detection Models. In the Proceedings of IEEE Transactions Pattern Analysis and Machine Intelligence. Volume 32, Issue 10, pp. 1795–1808, 2010.

[13] Thomas Hofmann. Probabilistic Latent Semantic Indexing. In the Proceedings of the 22nd Annual International SIGIR Conference on Research and Development in Information Retrieval, ACM New York, USA, pp. 50–57, 1999.

[14] S. C. Johnson. Hierarchical Clustering Schemes. Psychometrica, No 2, pp. 241–254, 1967.

[15] D. Kurz and F. Xu. Text Mining for the Extraction of Domain Retrieval Terms and Term Collocations. In the Proceedings of the International Workshop on Computational Approaches to Collocations, 2002.

[16] Daniel D. Lee and H. Sebastian Seung. Algorithms for Non-negative Matrix Factorization. In the Proceedings of NIPS, pp. 556–562, 2000.

[17] N. Loukachevitch. Automatic Term Recognition Needs Multiple Evidence. In the Proceedings of the 8th International Conference on LREC, 2012.

[18] J. B. MacQueen. Some Methods for classification and Analysis of Multivariate Observations. In the Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability. University of California Press, pp. 281–297, 1967.

[19] Christopher D. Manning, Prabhakar Raghavan and Hinrich Schutze. Introduction to Information Retrieval. Cambridge University Press, 2008.

[20] H. Nakagawa and T. Mori. A Simple but Powerful Automatic Term Extraction Mehod.

In the Proceedings of the Second International Workshop on Computational Terminology, pp. 29–35, 2002.

[21] H. Nakagawa and T. Mori. Automatic Term Recognition based on Statistics of Compound Nouns and their Components. Terminology, vol. 9, no. 2, pp. 201–219, 2003.

[22] R. Navigli and P. Velardi. Semantic Interpretation of Terminological Strings. In the Proceedings of the 6th International Conference on Terminology and Knowledge Engineering, Springer, pp. 95–100, 2002.

[23] M. A. Nokel, E. I. Bolshakova, N. V. Loukachevitch. Combining Multiple Features for Single-Word Term Extraction. Компьютерная лингвистика и интеллектуальные технологии. По материалам конференции Диалог-2012, Бекасово, pp. 490–501.

[24] Y. Park, R. J. Bird, B. Boguraev. Automatic glossary extraction beyond terminology identification. In the Proceedings of the 19th International Conference on Computational Linguistics, 2002.

[25] P. Pecina and P. Schlesinger. Combining Association Measures for Collocation Extraction. In the Proceedings of the COLING/ACL, ACL Press, pp. 651–658, 2006.

[26] A. Peñas, V. Verdejo, J. Gonzalo. Corbus-based Terminology Extraction Applied to Information Access. In the Proceedings of the Corpus Linguistics 2001 Conference, pp. 458–465, 2001.

[27] X.-H. Phan, C.-T. Nguyen. GibbsLDA++: A C/C++ implementation of latent Dirichlet Allocation (LDA), 2007.

[28] G. Salton. Automatic text processing: the transformation, analysis, and retrieval of information by computer. Addison-Wesley, 1989.

[29] K. Stevens, P. Kegelmeyer, D. Andrzejewski, D. Buttler. Exploring Topic Coherence over many models and many topics. In the Proceedings of EMNLP-CoNLL, pp. 952–961, 2012.

[30] Weka 3. Data Mining Software in Java. `http://www.cs.waikato.ac.nz/ml/weka`

[31] W. Wong, W. Liu, M. Bennamoun. Determining Termhood for Learning Domain Ontologies using Domain Prevalence and Tendency. In the Proceedings of the 6th Australasian Conference on Data Mining, pp. 47–54, 2007.

[32] W. Xu, X. Liu, Y. Gong. Document Clustering Based On Non-negative Matrix Factorization. In the Proceedings of SIRGIR, pp. 267–273, 2003.

[33] Shi Zhong. Efficient Online Spherical K-means Clustering. In the Proceedings of IEEE-IJCNN, Monreal, Canada, July 31 – August 4, pp. 3180–3185, 2005.

[34] К. В. Воронцов и А. А. Потапенко. Регуляризация, робастность и разреженность вероятностных тематических моделей. Журнал “Компьютерные исследования и моделирование”, т. 4, №12, с. 693–706, 2012.

[35] Н. В. Лукашевич. Тезаурусы в задачах информационного поиска. Москва: Издательство Московского университета, 2011.

Application of Topic Models to the Task of Single-Word Term Extraction

Michael Nokel, Natalia Loukachevitch

The paper describes the results of an experimental study of statistical topic models applied to the task of single-word term extraction. The English part of the Europarl corpus and the Russian articles taken from online banking magazines were used as target text collections. The experiments demonstrate that topic information significantly improves the quality of single-word term extraction, regardless of the subject area and the language used.

Поддержка повторного использования спецификаций потоков работ за счет обеспечения их независимости от конкретных коллекций данных и сервисов

© Брюхов Д.О. © Вовченко А.Е. © Калиниченко Л.А.
ИПИ РАН,
Москва
brd@ipi.ac.ru itsnein@gmail.com leonidk@synth.ipi.ac.ru

Аннотация

Статья рассматривает вопросы организации исследований в науках с интенсивным использованием данных (НИИД). Конкретно в ней изучается проблема повторного использования потоков работ в научных исследованиях. В статье представлен подход к встраиванию предметных посредников в среду для совместных исследований в НИИД. Этот подход позволяет создавать методы и алгоритмы решения задач независимо от конкретных реализаций ресурсов (данных и сервисов). За счет обеспечения независимости потоков работ от конкретных коллекций данных и сервисов существенно упрощается возможность повторного использования потоков работ.

1 Введение

Науки с интенсивным использованием данных (НИИД) развиваются в рамках новой парадигмы научных исследований (так называемой 4-й парадигмы [14]), согласно которой новые знания образуются в результате анализа разнообразных данных, накопленных в результате проведения измерений, наблюдений, моделирования, вычислений. Формулирование этой парадигмы явилось результатом осознания все возрастающей роли данных для развития науки, научных открытий практически во всех научных областях. Данные становятся ключевым источником получения знаний в НИИД. При этом объем, разнообразие и качество накапливаемых данных быстро растет отчасти благодаря быстрому развитию техники наблюдений и измерений различных природных явлений и процессов, введению в практику новых методов и инструментов наблюдения. Поэтому системы с интенсивным использованием данных имеют существенное пересечение с быстро развиваемой областью, именуемой «Big Data».

Вместе с тем, в НИИД «ученые, вместо того, чтобы заниматься исследованиями, затрачивают большую часть своего времени на поиск данных, манипулирование, обмен данными. И такое положение все время усугубляется» (наблюдение DoE Office of Science Data Management Challenge в USA).

Наиболее заметны следующие проблемы организации исследований в НИИД:

1) Создаваемые в НИИД методы анализа данных и алгоритмы решения задач как правило ориентированы на конкретные коллекции данных, находящиеся в поле зрения конкретных ученых в конкретный момент. Из-за этого отсутствует возможность повторного использования таких методов, алгоритмов и их реализаций над другими данными, в других коллективах НИИД.

2) Отсутствует практика накопления и повторного использования методов анализа данных, алгоритмов решения задач и их реализаций в научном сообществе НИИД. Фактически опыт проведения исследований, методы решения задач анализа данных в НИИД не накапливаются.

3) В НИИД отсутствует практика формирования ИТ-базированных, согласованных в сообществах концептуальных определений научных областей (включающих их структуру, понятия, спецификации методов, задач, техник проведения измерений и экспериментов, и пр.).

Данная статья подготовлена в рамках проекта[1], ориентированного на преодоление названных проблем. Для преодоления проблемы (2) предлагается использовать потоки работ как

[1] Проект “Обеспечение повторного использования реализаций методов анализа информации и алгоритмов решения задач в научных областях с интенсивным использованием данных" в рамках программы фундаментальных исследований Президиума РАН № 16 «Фундаментальные проблемы системного программирования»

Труды 15-й Всероссийской научной конференции «Электронные библиотеки: перспективные методы и технологии, электронные коллекции» — RCDL-2013, Ярославль, Россия, 14-17 октября 2013 г.

универсальное средство определения и реализации методов анализа данных, алгоритмов решения задач и их композиций. Опыт проведения исследований с интенсивным использованием данных в научном сообществе НИИД предлагается накапливать в виде потоков работ и их метаописаний. Средства накопления спецификаций потоков работ реализованы при этом на основе обоснованного выбора одного из существующих международных проектов подобных систем (таких как myExperiment [4], Wf4Ever [11], VisTrails [10], Trident [9], и др.). Одним из существенных недостатков таких проектов является отсутствие возможности использования в них концептуальных определений коллекций данных, обрабатываемых потоками работ (проблема 3), и, как следствие этого, ориентированность потоков работ на конкретные коллекции данных, что препятствует возможности повторного использования спецификаций потоков работ и их реализаций над другими данными в других исследованиях НИИД (проблема 1). В статье показано, как преодолеть названные недостатки за счет введения концептуальных спецификаций в практику определения потоков работ и задания отображений в них конкретных коллекций данных на основе техники предметных посредников. Тем самым удается обеспечить независимость накапливаемых для повторного использования спецификаций потоков работ от конкретных коллекций данных, а также при необходимости применить интеграцию конкретных коллекций данных для образования адекватных концептуальных коллекций.

2 Среды для публикации и повторного использования потоков работ

В настоящем разделе дан краткий обзор систем, обеспечивающих публикацию и повторное использование спецификаций потоков работ.

Особо стоит выделить среду для совместных исследований myExperiment [4], в которой ученые могут публиковать потоки работ для решения задач. Среда myExperiment была введена в 2007 году и в настоящее время является одной из самых больших репозиториев потоков работ (в ней содержится более 2000 потоков работ), используется тысячами ученых в различных областях науки. Среда myExperiment позволяет публиковать потоки работ в различных системах управления потоками работ. Для ряда систем управления потоками работ (таких, как Taverna [6], Galaxy [8], Trident [9]) поддерживаются дополнительные возможности такие, как управление метаданными, извлечение информации об используемых сервисах, визуализация потоков работ.

Другим примером репозитория потоков работ является проект ER-flow [5] (проект FP7 "Building a European Research Community through Interoperable Workflows and Data"), являющийся продолжением проекта SHIWA. Проект ER-flow предоставляет ученым программную поддержку для создания, обмена и запуска потоков работ в различных системах управления потоками работ (ASKALON, Galaxy, GWES, Kepler, LONI Pipeline, MOTEUR, Pegasus, P-GRADE, ProActive, Triana, Taverna, WS-PGRADE).

Системы управления потоками работ в науке поддерживают доступ к широкому набору уже существующих баз данных и сервисов анализа данных в различных областях науки (в биологии, астрономии, социальных науках, и др.), использование которых позволяет упростить процесс создания потоков работ.

Репозитории потоков работ позволяют ученым находить интересующие их потоки работ, воспроизводить результаты этих потоков работ, повторно использовать существующие потоки работ для решения задач в рамках названных выше ограничений.

Для конкретизации рассмотрения в данной статье предполагается использовать myExperiment с ориентацией на систему управления потоками работ Taverna [6]. Taverna – это система управления потоками работ, которая может быть использована в различных областях науки. Она предоставляет набор сервисов для создания и выполнения разнообразных потоков работ. Taverna была создана в рамках проекта myGrid [7].

3 Проблемы повторного использования потоков работ

Taverna предоставляет средства для поиска (по тегам) потоков работ в среде myExperiment. Найденные потоки работ можно запускать как с исходными значениями входных параметров, предоставленными разработчиками, так и с произвольными значениями. Это позволяет воспроизвести результаты исследования других ученых с целью возможного повторного использования разработанных потоков работ. Тем не менее зачастую повторное использование может оказаться невозможным.

Спецификация потока работ в Taverna задается в виде направленного графа. Потоки работ в Taverna реализуют модель потоков данных (data flow model). Таким образом,.поток работ состоит из сервисов, представляющих собой программные компоненты (такие как веб-сервисы), и направленных связей между ними, выражающих зависимости по данным. Taverna поддерживает широкий набор как локальных, так и удаленных сервисов в различных областях науки. В частности, Taverna обеспечивает доступ к произвольным WSDL и REST сервисам; к конкретным веб сервисам, таким как BioMoby [15], BioMart [12] и SoapLab [16]; к локальным Java сервисам (BeanShell скрипты); к базам данных посредством JDBC. Taverna поддерживает использование вложенных потоков работ. Это позволяет встраивать уже существующие потоки

работ (возможно разработанные другими учеными) при создании новых потоков работ.

Одной из главных проблем повторного использования потоков работ в Taverna является зависимость спецификаций потоков работ от конкретных коллекций данных и/или сервисов. В Taverna каждый сервис настраивается на доступ к конкретным сервисам и базам данных. Это не позволяет повторно использовать такие потоки работ, если необходимо, например, обрабатывать другие коллекции данных. Также, если какой-либо из сервисов или база данных в настоящий момент недоступны, то весь поток работ не сможет быть выполнен.

Данная статья нацелена прежде всего на решение проблемы повторного использования потоков работ в Taverna над базами данных. Taverna поддерживает ряд способов доступа к базам данных из потока работ:

1. Создание веб сервиса, реализующего доступ к базе данных. Доступ к этому веб сервису из потока работ осуществляется по протоколу SOAP;
2. Полная реализация интерфейса расширения (extension point) Taverna, включающего поддержку языка запросов к базе данных и графический интерфейс для конструирования запросов и предоставления пользователю метаданных подключаемой базы данных. В Taverna этот подход реализован для сервиса BioMart [12] и в плагине AstroTaverna [13];
3. Использование существующих сервисов BioMart для доступа к подключаемой базе данных;
4. Использование JDBC сервиса для доступа к базам данных.

Возможность подключения нового ресурса через BioMart заслуживает отдельного рассмотрения. BioMart (а точнее BioMart портал) представляет собой систему управления данными, ориентированную на выполнение разнообразных запросов над биологическими данными. В портале системы можно найти нужные ресурсы по метаданным, а также задать к ним запрос и получить результат. Также запросы могут быть заданы над несколькими конкретными базами данных, зарегистрированными в портале. Данные из BioMart могут быть получены посредством веб страницы, графического или консольного инструментария, или из программ посредством веб-сервисов либо напрямую через perl или java АПИ.

С другой стороны, BioMart (а точнее BioMart сервис) представляет собой адаптер, унифицирующий интерфейс различных баз данных, таких как MS SQL Server, PostgreSQL, MySQL, DB2, Oracle. По сути, любая (из поддерживаемых) база данных может быть оформлена как BioMart сервис, после чего полученный сервис подключается к порталу. С точки зрения схемы ресурса, при создании BioMart сервиса возможно определение взглядов (SQL views) над исходной схемой для ее модификации (удалить атрибуты, убрать какие-то таблицы, добавить ключи, и др.). Также, для повышения производительности взгляды можно материализовать. BioMart автоматически обновляет материализованные взгляды в случае изменения исходных данных в ресурсе. Кроме того, можно устанавливать связи между различными базами данных (по ключам), образуя их федерацию.

С концептуальной точки зрения схемы BioMart сервисов определяются на основе схем ресурсов. Это подход известен в литературе как GAV [2] и обладает рядом недостатков, основным из которых является слабая масштабируемость, т.к. добавление (удаление) одного из ресурсов влечет за собой изменение федеративной схемы. Инструментарий Taverna предоставляет доступ не к BioMart порталу, а к отдельным BioMart сервисам. Чтобы добавить новую операцию в поток работ, выбирается конкретный BioMart сервис, с конкретной схемой, и формулируется конкретный запрос, что также затрудняет повторное использование потока этого работ.

Основное отличие предлагаемого в настоящей работе подхода заключается в поддержке концептуальной схемы предметной области для спецификации потоков работ и введении промежуточного слоя предметных посредников, обеспечивающего отображение схем произвольных конкретных ресурсов (баз данных и сервисов) в концептуальную схему, интеграцию ресурсов. Благодаря этому спецификация потоков работ не требует изменения при изменении ресурсов, что является необходимым условием обеспечения повторного использования потоков работ.

4 Инфраструктура предметных посредников как средство решения проблем повторного использования

4.1 Концепции инфраструктур предметных посредников

Основной идеей инфраструктуры решения задач над неоднородными информационным ресурсами является введение промежуточного слоя между ресурсами и потребителями информации, образуемого предметными посредниками [1]. Каждый предметный посредник поддерживает спецификацию предметной области для решения некоторого класса задач.

Посредники реализуют подход к решению задач, ориентированный на проблему. В рамках подхода, ориентированного на проблему (подхода, «движимого приложением»), формулируется концептуальная спецификация задачи, включающая базовые сущности и понятия предметной области, функции, процессы и пр. Такое определение предметной области, представляет собой спецификацию предметного посредника для решения класса задач. Сущности и понятия предметной области, определенные таким образом, не зависят от существующих информационных

ресурсов. В терминах предметной области формулируются программы для решения задачи на языке правил посредника и на языках программирования. Для решения конкретной задачи выявляются инфраструктуры, содержащие ресурсы, необходимые для ее решения (например, гриды, облачные инфраструктуры, репозитории данных, и др.). Далее, идентифицируются ресурсы, релевантные задаче, используя реестры доступных инфраструктур. Релевантные задаче ресурсы регистрируются в предметных посредниках, задающих отображение схем ресурсов в концептуальную спецификацию.

Таким образом, при изменении набора ресурсов, спецификация алгоритма решения задачи остается неизменной, и может быть повторно использована на другом наборе коллекций данных.

4.2 Обеспечению независимости потоков работ от данных на основе предметных посредников

Как было отмечено выше, все сервисы в потоках работ Taverna определены в терминах конкретных сервисов и баз данных, что не позволяет задавать спецификации потоков работ независимо от конкретных ресурсов.

По сути, посредники представляют собой виртуальные базы данных, и в потоках работ Taverna их можно подключать аналогично обычным базам данных. Возможны 2 способа подключения посредников к Taverna: посредством веб сервиса и посредством разработанного плагина (соответствующие 1-му и 2-му способам, рассмотренным в разделе 3). При первом способе над посредником создается веб сервис, реализующий интерфейс посредника. Доступ к посреднику из потоков работ Taverna осуществляется посредством этого веб сервиса по протоколу SOAP. Вторым способом подключения предметных посредников к Taverna может являться разработка специального плагина под средство разработки потоков работ Taverna Workbench. Taverna предоставляет возможность создания подобных плагинов, посредством интерфейса расширения (extension point), для добавления и расширения функциональности Taverna Workbench. Этот плагин сможет предоставлять графический интерфейс для помощи в конструировании запросов к предметным посредникам и интерфейс для доступа к метаданным предметного посредника.

Все доступные в Taverna ресурсы, используемые в качестве узлов в потоках работ, могут быть использованы также посредством посредников. В частности, предметные посредники поддерживают использование WSDL сервисов в виде функций. Конкретные веб-сервисы (например, BioMoby, BioMart и SoapLab) также могут быть использованы из посредника. BeanShell скрипты могут быть оформлены в виде программ на Java над предметным посредником, либо в виде функции предметного посредника. Базы данных подключаются к посреднику посредством адаптеров.

Концептуальные коллекции с технической точки зрения могут быть использованы точно также как обычные базы данных в Taverna. С помощью предметных посредников в виде концептуальных коллекций могут быть оформлены любые базы данных. Главное отличие концептуальных коллекций от обычных заключается в том, что их схема остается неизменной независимо от набора фактически используемых ресурсов. В результате, запросы к концептуальной коллекции, и следовательно, поток работ остаются неизменными при изменении набора конкретных ресурсов. Таким образом может быть получена спецификация потока работ, определяемая в терминах предметной области предметного посредника и не зависящая от конкретных ресурсов. Это решает одну из основных проблем повторного использования потоков работ.

5 Пример применения подхода к обеспечению независимости спецификации потоков работ на основе задачи определении вторичных стандартов

В этом разделе мы рассмотрим предлагаемый нами подход на задаче определения вторичных стандартов для фотометрической калибровки оптических компонентов космических гамма-всплесков [3], поставленной Институтом Космических Исследований РАН. Задача заключается в том, что по координатам площадки, требуется найти в ней звезды, удовлетворяющие ряду условий (не переменные, точечные, с хорошими изученными параметры). Такие звезды называется «стандартами» и могут быть использованы для калибровки новых поступающих данных.

5.1 Описание схемы посредника для задачи определения вторичных стандартов

На Рис. 1 представлена схема посредника, разработанная для решения этой задачи. Она включает в себя описание концептов, необходимых для решения задачи, таких как: экваториальные координаты (CoordEQJ); фотометрическую систему (PhotometricSystem); фотометрическую полосу (Passband); магнитуду в некоторой фотометрической системе (Magnitude); абстрактный астрономический объект (Astronomical Object); звезду (Star); стандарт (Standard); изображение (Image). Также схема посредника содержит функции, необходимые для решения задачи, включая: метод кросс-идентификации (matchObjects); метод вычисления цветового индекса (colorIndex); метод проверки типа объекта по некоторому эталонному каталогу (каталогам) (checkType); метод проверки, является ли звезда переменной на основе данных из многих других ресурсов (isVariable).

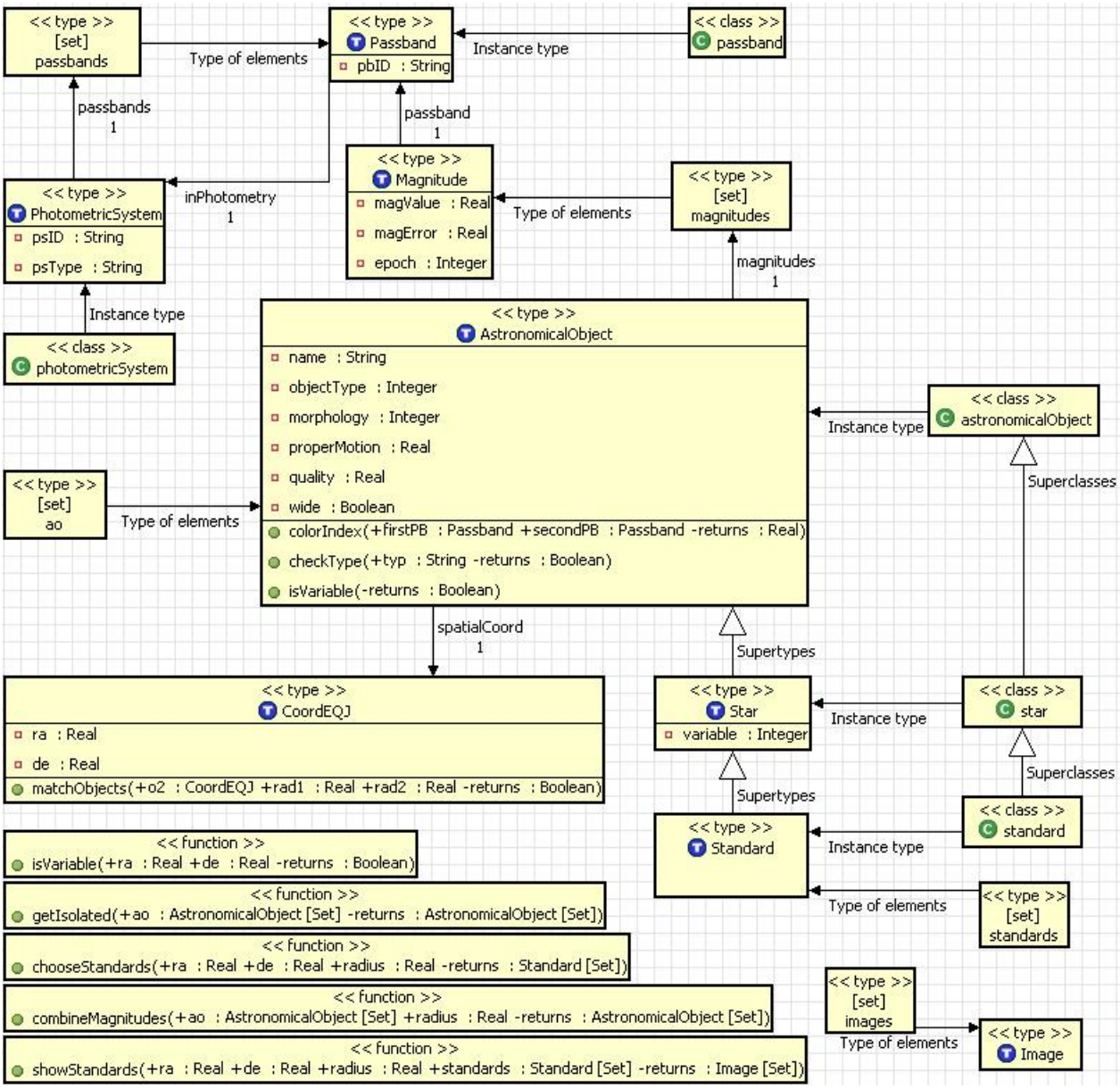


Рис. 1 Схема посредника для задачи определения вторичных стандартов

Представленная схема не зависит от конкретных ресурсов, используемых для решения задачи. Каталоги SDSS, USNOB-1, 2MASS, GSC, UCAC – основные ресурсы, используемые для извлечения стандартов. Именно среди этих каталогов отбираются все звезды, удовлетворяющие параметрам. Каталоги VSX, ASAS, GCVS, NSVS используются для проверки факта переменности выбранных стандартов. Список ресурсов может со временем меняться, но при этом схема посредника останется неизменной и методы решения задач определения вторичных стандартов также останутся неизменными.

5.2 Программа решения задачи определения вторичных стандартов

Задача определения стандартов была сформулирована в виде программы (последовательности правил) над схемой, рассмотренной выше. Параметром программы является площадка на небесной сфере, в которой произошел гамма-всплеск. Площадка характеризуется центром с координатами queryRA, queryDE и радиусом radius. Программа посредника состоит из восьми последовательных правил.

Правило 1 – В первом правиле среди всех астрономических объектов выбираются те, что попадают в указанную площадку. При этом нас интересуют только координаты (ra, de), звездные величины в различных полосах (magnitudes), тип объекта (objectType), собственное движение (properMotion) и качество данных (quality). Это правило на языке правил посредников (язык СИНТЕЗ [17]) выглядит следующим образом:

```
r(x/[ra, de, name, magnitudes, objectType,
properMotion, quality])
:- astronomicalObject(x1/[ra: spatialCoord.ra, de:
spatialCoord.de, name, objectType, properMotion,
quality, magnitudes])
& ra < queryRA + radius & ra > queryRA - radius
```

```
& de < queryDE + radius & de > queryDE - radius
```

Правило продуцирует коллекцию r, состоящую из астрономических объектов (astronomicalObject), содержащих необходимые атрибуты и удовлетворяющих ограничениям на координаты, указанные в теле правила.

Правило 2 – Во втором правиле отсеиваются неизолированные объекты. Изолированные объекты – это объекты, в некоторой окрестности которых на небесной сфере не наблюдается других объектов:

```
getIsolated(r1, r2);
```

Правило 3 – В третьем правиле среди ранее выбранных объектов отсеиваются галактики, и выбираются звезды с очень малым собственным движением и качественными фотометрическими данными:

```
r3(x/[ra, de, name, magnitudes])
:- r2(x1/[ra, de, name, objectType, properMotion,
quality, magnitudes])
& checkType(ra, de, 'Galaxy', nType) & nType = false
& objectType = Star
& properMotion < 0.01
& quality  < 0.01
```

Правило 4 - В четвертом правиле используются объекты, полученные в первом правиле. Среди объектов этого класса выбираются только те, для которых верно, что они переменные. Переменность определяется с помощью функции isVariableByMagnitude.

```
r4(x/[ra, de, name])
:- r1(x1/[ra, de, name, magnitudes])
& isVariablebyMagnitudes(ra, de, isVar) & isVar = true
```

Правило 5 - В пятом правиле выбираются переменные звезды из каталогов переменных звезд: GCVS, VSX, NSVS, ASAS.

```
r4(x/[ra, de, name])
:- variableStar(x1/[ra: spatialCoord.ra, de:
spatialCoord.de, name])
```

Правило 6 - В шестом правиле, производится кросс-идентификация объектов из класса кандидатов в стандарты (результат правила 3), и класса переменных звезд, посредством вызова функции xmatch.

```
xmatch(r3, r4, r5);
```

Правило 7 - В седьмом правиле из класса кандидатов в стандарты, полученного после кросс-идентификации, выбираются только те объекты, для которых не нашлось близко расположенного переменного объекта (distance > 0.01). На практике, это означает что кандидат в стандарты – не переменный объект.

```
r6(x/[ra, de, name magnitudes])
:- r5(x1/[ra, de, name, magnitudes, distance])
& distance > 0.01
```

Правило 8 – В предыдущем правиле построена коллекция r6, содержащая стандартные звезды. В заключительном правиле стандарты маркируются на изображение площадки гамма-всплеска, и предоставляются пользователю для утверждения.

```
r7(im/Image)
:- r6(x/ra, de, name, magnitudes])
& showStadards(ra, de, radius, magnitudes, im)
```

5.3 Описание Веб сервиса для доступа к посреднику для задачи определения вторичных стандартов

Для доступа к предметному посреднику решения задачи определения стандартов был разработан Веб сервис. Этот веб сервис включает в себя следующие методы, реализующие описанные выше правила:

executeQuery – выполняет правило посредника [17]. Этим правилом достаются кандидаты в стандарты. В качестве правила используется комбинация из описанных выше правил 1-3 (раздел 5.2). Данные возвращаются в формате SynthClass[2].

getVariableStarsFromCatalogues - получает из посредника коллекцию переменных звезд в заданной области из каталогов переменных звезд (правило 5). Данные возвращаются в формате SynthClass.

getVariableStarsByMagnitudes - получает из посредника коллекцию переменных звезд в заданной области, определяя переменная ли она по магнитудам (правило 1 и 4). Данные возвращаются в формате SynthClass.

removeVariableStars - получает коллекцию стандартов, и коллекцию переменных (аналог правил 6 и 7 реализованных одной функцией). Из первой удаляются те объекты, которые содержатся во второй.

removeStarsWithAnomalyMagnitudes - отсеивает аномальные звезды из входной коллекции объектов. Это дополнительный метод, не описанный выше в правилах. Был добавлен по настоянию астрономов для обеспечения большей точности результата.

getAladinCandidates – по полученной коллекции объектов возвращает изображение (аналоги правила 8), которое может быть открыто специалистом из программы Aladin [19], популярной среди астрономов.

[2] Формат представляет собой расширение стандартного для виртуальной обсерватории представления таблиц VOTable [18]. Расширения обеспечивают возможность представления коллекций объектов сложной структуры.

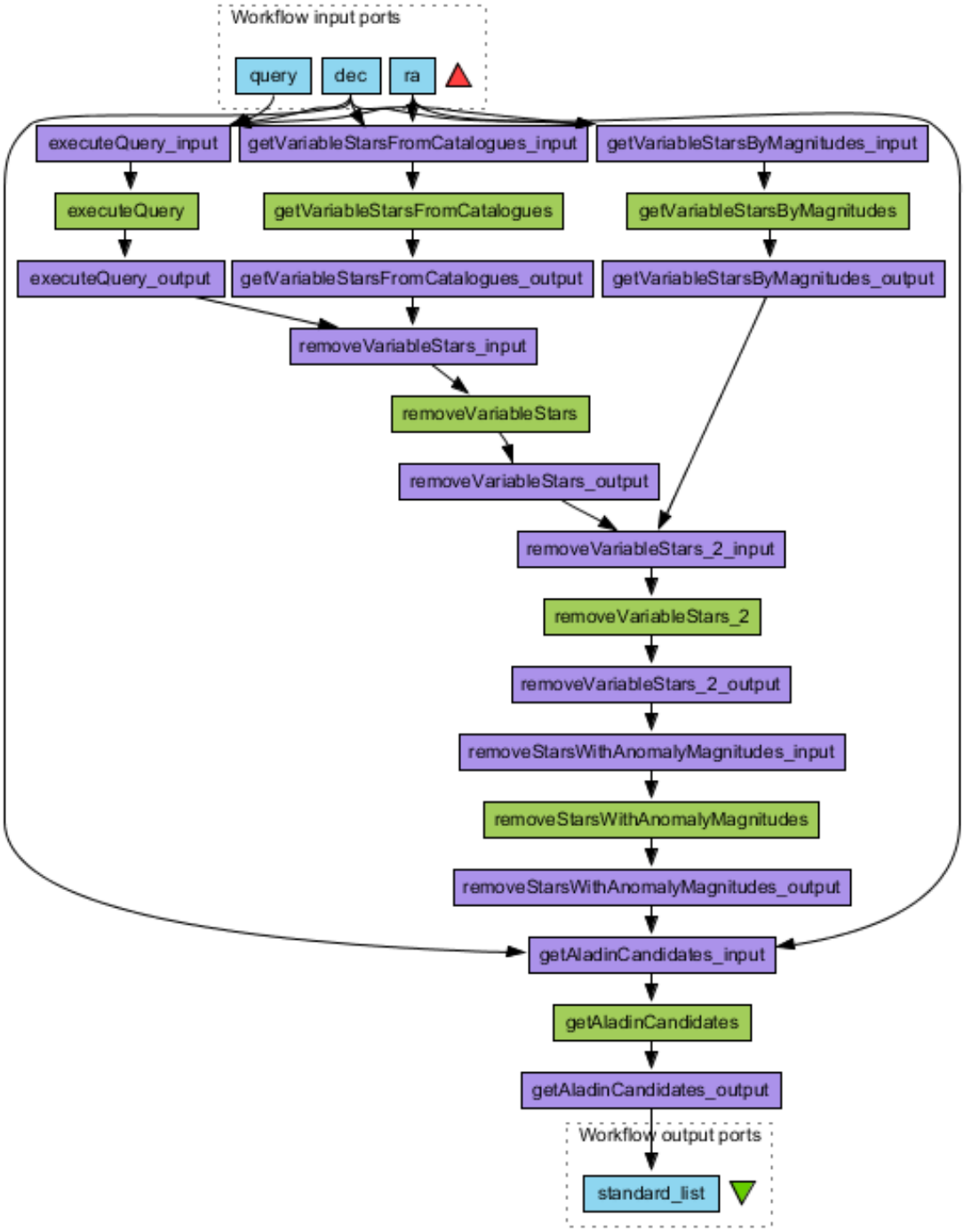


Рис. 2 Поток работ решения задачи вторичных стандартов в среде Taverna

5.4 Описание потока работ решения задачи определения вторичных стандартов в среде Taverna

На Рис. 2 представлен поток работ решения задачи вторичных стандартов в среде Taverna. Входными параметрами его являются координаты площадки на небесной сфере, в которой произошел гамма-всплеск.

Поток работ представляет собой набор вызовов методов Веб сервиса, описанного выше. Также в потоке работ присутствуют вспомогательные функции преобразования входных и выходных параметров методов в формат XML.

Результатом выполнения этого потока работ является изображение Aladin [19] с наложенным на него списком стандартов. На Рис. 3 показан пример результата, получаемого специалистом. Результат включает в себя изображение, а также отмеченные на изображении объекты – кандидаты в стандарты, удовлетворяющие всем требованиям.

5 Заключение

Предлагаемый подход по встраиванию предметных посредников в среду организации исследований в НИИД позволяет упростить

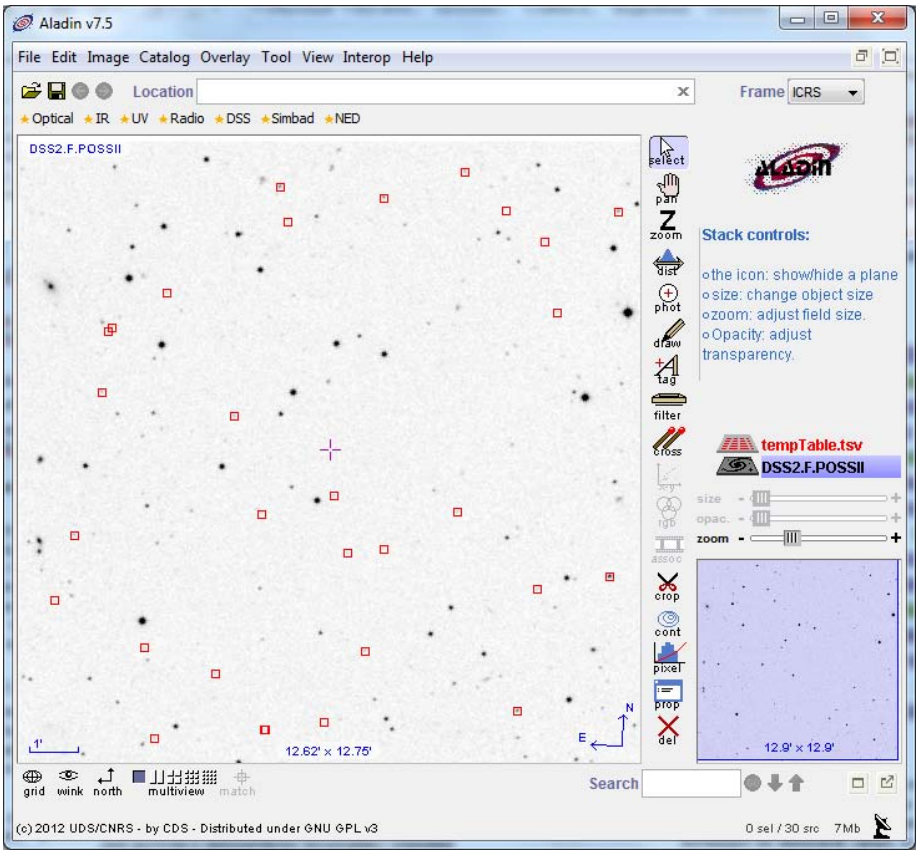


Рис. 3 Изображение найденных кандидатов в стандарты

решение ряда проблем таких, как: накопление методов анализа данных, алгоритмов решения задач и их реализаций в научном сообществе; воспроизведение и повторное использование таких алгоритмов и методов; формирование ИТ-базированных концептуальных определений научных областей; использование методов и средств высокоуровневых декларативных определений методов анализа данных и алгоритмов решения задач в НИИД. Хотя статья рассматривает предлагаемый подход применительно к конкретной среде myExperiment и системе управления потоками работ Taverna, предлагаемый подход может быть аналогично использован в других средах с другими системами управления потоками работ.

Литература

[1] Брюхов Д.О., Вовченко А. Е., Захаров В.Н., Желенкова О.П., Калиниченко Л.А., Мартынов Д.О., Скворцов Н.А., Ступников С.А. Архитектура промежуточного слоя предметных посредников для решения задач над множеством интегрируемых неоднородных распределенных информационных ресурсов в гибридной грид-инфраструктуре виртуальных обсерваторий // Информатика и ее применения. – М., 2008. – Т. 2, Вып. 1. – С. 2-34.

[2] Alon Y. Halevy. Answering Queries Using Views: A Survey. VLDB Journal, 10(4), 2001.

[3] Вовченко А.Е., Вольнова А.А., Денисенко Д.В., Калиниченко Л.А., Куприянов В.В., Позаненко А.С., Скворцов Н.А., Ступников С.А. Применение средств виртуальной обсерватории для выбора вторичных стандартов поля при фотометрии оптического послесвечения гамма-всплесков // Труды Всероссийской астрономической конференции ВАК-2010 «От эпохи Галилея до наших дней». – САО РАН: Нижний Архыз. – 2010.

[4] De Roure, D., Goble, C. and Stevens, R. (2009) The Design and Realisation of the myExperiment Virtual Research Environment for Social Sharing of Workflows. Future Generation Computer Systems 25, pp. 561-567

[5] Mark Santcroos. Experiences from workflow sharing using the SHIWA Workflow Repository for application porting to DCI. EGI Community Forum Book of Abstracts, EGI, Manchester, UK, 2013.

[6] Katherine Wolstencroft, Robert Haines, Donal Fellows, Alan Williams, David Withers, Stuart Owen, Stian Soiland-Reyes, Ian Dunlop, Aleksandra Nenadic, Paul Fisher, Jiten Bhagat, Khalid Belhajjame, Finn Bacall, Alex Hardisty, Abraham Nieva de la Hidalga, Maria P. Balcazar Vargas, Shoaib Sufi, and Carole Goble. The Taverna workflow suite: designing and executing workflows of Web Services on the desktop, web or in the cloud. Nucleic Acids Research, First published online May 2, 2013.

[7] myGrid project http://www.mygrid.org.uk/

[8] Goecks, J, Nekrutenko, A, Taylor, J and The Galaxy Team. Galaxy: a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences. Genome Biol. 2010 Aug 25;11(8):R86.

[9] Roger Barga, Jared Jackson, Nelson Araujo, Dean Guo, Nitin Gautam, Yogesh Simmhan. The Trident Scientific Workflow Workbench. Proceeding of the 2008 Fourth IEEE International Conference on eScience, Pages 317-318, December 07-12, 2008.

[10] Steven P. Callahan, Juliana Freire, Emanuele Santos, Carlos E. Scheidegger, Claudio T. Silva and Huy T. Vo. VisTrails: Visualization meets Data Management. Proceedings of ACM SIGMOD 2006.

[11] Wf4Ever project http://www.wf4ever-project.org/

[12] Kasprzyk A. BioMart: driving a paradigm change in biological data management. Database (Oxford) 2011.

[13] Walton N. A., Witherwick D. K., Oinn T., Benson K. M. Taverna and workflows in the virtual observatory, Astronomical Data Analysis Software and Systems ASP Conference Series, Vol. 394, Proceedings of the conference held 23-26 September, 2007, p 309.

[14] The Fourth Paradigm: Data-Intensive Scientific Discovery. Tony Hey, Stewart Tansley, and Kristin Tolle, Eds. Microsoft Research, Redmond, WA, 2009. 286 pp.

[15] M. D. Wilkinson, D. Gessler, A. Farmer, L. Stein. The BioMOBY Project Explores Open-Source, Simple, Extensible Protocols for Enabling Biological Database Interoperability. In Proceedings of the Virtual Conference on Genomics and Bioinformatics (2003).

[16] Martin Senger, Peter Rice, Tom Oinn. Soaplab - a unified Sesame door to analysis tools, Proceedings, UK e-Science, All Hands Meeting 2003, Editors - Simon J Cox, p.509-513, ISBN - 1-904425-11-9, September 2003.

[17] Kalinichenko L.A., Stupnikov S.A., Martynov D.O. SYNTHESIS: a Language for Canonical Information Modeling and Mediator Definition for Problem Solving in Heterogeneous Information Resource Environments. Moscow: IPI RAN, 2007.

[18] VOTable Format Definition http://www.ivoa.net/documents/VOTable/

[19] Aladin Sky Atlas http://aladin.u-strasbg.fr/

Support of the workflow specifications reuse by ensuring its independence of the specific data collections and services

© Briukhov D.O., Vovchenko A.E., Kalinichenko L.A.

Institute of Informatics Problems (IPI RAN)

The paper is devoted to the problem of organization of the research process in the data-intensive sciences (DIS). It is focused on the problem of the workflow reuse. The paper presents an approach of embedding the subject mediators into the environment for collaborative research in DIS. This approach provides independence of problem solving methods and algorithms of the source data and services. It is shown that the independence of workflow from particular data collections and services constitutes a necessary requirement for the workflows re-use.

Метаданные о научных методах для обеспечения их повторного использования и воспроизводимости результатов

© Н. А. Скворцов, Д. О. Брюхов, Л. А. Калиниченко, Д. Ковалёв, С. А. Ступников
Институт проблем информатики РАН
Москва
nskv@ipi.ac.ru

Аннотация

В науках с интенсивным использованием данных предъявляются высокие требования к обработке больших объёмов данных набором научных методов для получения вторичной информации и новых знаний об исследуемых объектах. При этом важной оказывается доступность реализаций научных методов, применяемых в предметной области для организации обработки данных и решения задач. Обеспечение электронного хранения, повторного использования и воспроизводимости результатов экспериментов становятся неотъемлемыми атрибутами реализаций научных методов. В статье исследуется состав метаданных, которыми должны сопровождаться процессы, специфицирующие или реализующие научные методы, для обеспечения их повторного использования и воспроизводимости результатов. Компоненты процессов и данные сопоставляются с понятиями предметной области, сопровождаются информацией об их происхождении и качестве, системы тестов описывают разновидности ситуаций, в которых методы должны работать определённым образом. На примере открытой среды MyExperiment, организующей и предоставляющей доступ к коллекции научных потоков работ, показано, как расширение состава метаданных потоков работ позволяет организовать в коллекции семантический поиск релевантных решаемой задаче научных методов, проверить найденные реализации методов на интероперабельность, возможность повторного использования и обеспечить воспроизводимость результатов, полученных при их применении.

Работа выполнена при поддержке РФФИ (гранты 11-07-00402-а, 13-07-00579-а) и Президиума РАН (программа 16П, проект 4.2).

Труды 15-й Всероссийской научной конференции «Электронные библиотеки: перспективные методы и технологии, электронные коллекции» — RCDL-2013, Ярославль, Россия, 14-17 октября 2013 г.

1 Введение

Получение колоссальных объёмов данных, подлежащих анализу научным сообществом, рождает качественное изменение в подходах к построению информационных систем для обработки данных и поддержки научных исследований. Науки с интенсивным использованием данных [1] призваны выявить полезные знания из объёма накопившихся ранее данных и потока появляющихся данных. Это требует постоянного автоматического применения широкого ассортимента известных методов, включая оценку существенных свойств и параметров объектов, проверку научных гипотез, выявление результатов, подтверждающих или опровергающих экспериментальные модели и так далее. Результаты применения научных методов сохраняются и становятся источником данных для работы других методов в данной области и сопряжённых проблемных областях.

Информационные системы в науках с интенсивным использованием данных комбинируют организацию информации в исследуемой области и организацию цифрового хранения и применения научных методов, используемых в данной предметной области. Научные методы могут представлять собой описание процессов обработки данных. Реализации методов разрабатываются в виде сервисов и потоков работ над доступными данными. Спецификации определяют, какие входные данные необходимы для работы методов, что и по каким алгоритмам они реализуют и какие результаты выдают. Потоки работ могут быть вложенными, то есть, вызывать друг друга в качестве подпроцессов.

Коллекции научных методов разрабатываются и занимают своё место в инструментах научного сообщества. В качестве примеров можно привести системы поддержки исследований в области астрономии. Проект VizieR [3] собирает всевозможные каталоги, организует их поиск и поиск в них, предоставляет набор сервисов, которые наиболее востребованы астрономическим сообществом. Однако до сих пор набор сервисов, реализующих какие-либо астрономические методы, достаточно ограничен. Этой информационной системой благодаря доступности данных пользуются практически все, кто работает с астрономическими данными. Среда виртуальной обсерватории Astrogrid [10], поддерживает удалённый доступ не только к данным, но и к сервисам различного назначения. Расширение Astrogrid средствами предметных посредников [12] позволило описывать в grid-среде спецификации предметных областей для формулирования и решения классов научных задач. Открытая коллекция научных потоков работ MyExperiment [6] объединяет тысячи пользователей и потоков работ и десятки проектов, в том числе в области астрономии, предоставляющих или использующих накопленные потоки работ.

Для того, чтобы подобные коллекции методов развивались и использовались научным сообществом, должна выйти на новый уровень вся инфраструктура поддержки научных исследований. Необходимо развитие и повсеместное использование сообществами общедоступных спецификаций предметных областей исследований и развитие семантических подходов решения задач с их использованием. Источники данных и реализации научных методов должны систематизироваться и связываться со спецификациями предметной области. Это позволяет упростить интеграцию информационных и методических ресурсов, автоматизировать многие шаги в обработке данных, которые до сих пор решались посредством ручных манипуляций всякий раз при решении новых задач. Реализации научных методов требуют разработки таким образом, чтобы упростить или даже автоматизировать их семантический поиск и использование в согласии со спецификациями предметной области. Данные и методы необходимо сопровождать информацией об их происхождении, точности, полноте. В цели разработки и реализации методов должны изначально закладываться возможность их повторного использования в данной и смежных областях, возможности воспроизведения результатов при одинаковых исходных данных. Создание инфраструктуры научных исследований, позволяющей использовать методы повторно, освободит исследователей от усилий, прилагаемых сегодня для интеграции неоднородных информационных ресурсов и реализации локально методов их обработки. Вместо этого само накопление методической базы, доступной, надёжной, согласованной со спецификациями предметной области и удобной в использовании, будет являться вкладом в развитие науке.

Данное исследование имеет целью разработку метаданных и методов работы с ними, которые должны сопровождать научные данные и реализации научных методов для достижения их повторного использования и воспроизводимости результатов научных экспериментов. В разделе 2 обсуждаются требования к доступным реализациям научных методов, исходным данным и получаемым результатам исследований в свете наук с интенсивным использованием данных. Раздел 3 посвящён связанным проектам и решениям. В разделе 4 более подробно описаны некоторые аспекты реализации проекта MyExperiment, выбранного для демонстрации возможностей инфраструктуры поддержки научных исследований с расширенным набором метаданных спецификаций научных потоков работ. Раздел 5 описывает собственно предлагаемый набор метаданных, сопровождающих доступные научные данные и реализации научных методов. В разделе 6 демонстрируется использование предложенных метаданных для организации поиска и повторного использования научных методов в инфраструктуре поддержки научных исследований.

2 Требования к реализации научных методов в среде поддержки научных исследований

Вначале необходимо представить требования, которые предъявляются к научным данным и методам для создания инфраструктуры поддержки научных исследований, позволяющей развивать спецификации предметных областей и коллекции научных данных и методов и использовать их реализации в исследованиях.

1. Под спецификацией предметной области, доступной и принимаемой сообществом исследователей, можно понимать набор связанных формальных онтологий предметной области исследования и смежных с ней областей. В соответствии с онтологиями могут создаваться концептуальные схемы предметной области, необходимые для организации информационных структур и спецификации методов, используемых в обработке данных.

Для развития семантических подходов к решению научных задач данные, информационные ресурсы и реализации научных методов необходимо связывать со спецификациями предметной области.

Агентами научного сообщества могут выступать как исследователи, так и информационные системы. Поэтому спецификации, описывающие методы и данные, должны обеспечивать понимание человеком и возможность машинной обработки. В этой связи необходимо использовать разработки, связанные с семантическим вебом [9].

2. Научные методы и данные должны быть отрыты и доступны для использования научным сообществом, работающим и решающим задачи в данной предметной области. Результаты работы методов также должны быть доступны для использования. Для этого они должны быть надлежащим образом специфицированы и опубликованы в общедоступных коллекциях. Коллекции собирают и систематизируют информацию и обеспечиваются средствами семантического поиска.

3. Важным принципом реализации научных методов является их независимость от источников данных. Подмена источников данных другими релевантными источниками надлежащего качества должна быть проста и не должна сказываться на работоспособности методов.

4. Для обеспечения повторного использования и данные, и методы необходимо сопровождать информацией об их происхождении. Она включает аутентификацию методов и данных, их источники, историю их развития и трансформации от создания до момента использования. С другой стороны, реализации методов должны сохранять информацию о происхождении обрабатываемых данных и обеспечивать дополнение этой информации в соответствии с манипуляциями, производимыми ими над данными.

5. Для оценки возможности повторного использования данных, методов и результатов расчётов или экспериментов необходима информация об их качестве: точности и полноте открытых данных, точности и полноте результатов, обеспечиваемых научными методами.

6. Обеспечение повторного использования также предполагает необходимость достаточно подробных спецификаций требований к их входным и выходным данным.

7. Обеспечение воспроизводимости результатов работы методов подразумевает под собой средства описания среды, необходимой для исполнения предоставляемых методов, спецификации поддерживаемых стандартов, а также наборы тестов, обеспечивающих проверку работы методов в различных ситуациях.

3 Связанные работы

Интересной разработкой с точки зрения накопления научных методов является среда разработки и сбора научных потоков работ MyExperiment [6]. Она организована как социальная сеть, позволяющая регистрировать исследователей, включать их в различные тематические группы, публиковать потоки работ, реализованные в различных сторонних системах, описывать эксперименты, связанные с вызовом потоков работ, составлять объекты исследования (фактически проекты), состоящие из потоков работ, документов, файлов данных, ссылок. Среда MyExperiment обеспечивает поиск потоков работ по метаданным, предоставляет их описание, позволяет их запускать. Интерфейсы среды соответствуют стандарту связанных открытых данных [11] и имеют соответствующие интерфейсы для этого. Тем временем, у данной среды есть ряд недостатков, препятствующих возможности повторного использования и воспроизведения результатов исполнения потоков работ.

То, что спецификации потоков работ публикуются в виде файлов, сгенерированных в форматах сторонних редакторов потоков работ, с одной стороны, позволяет использовать различные средства для их создания, с другой стороны, является причиной неоднородности и невозможности автоматизации использования опубликованных реализаций. В частности, спецификации потоков работ, созданные в наиболее используемом в данной среде внешнем редакторе Taverna [7], разбираются средой для выделения входных и выходных данных, визуализации структуры потоков работ, однако не имеет интерфейсов доступа к внутренней структуре потоков работ.

Данные для экспериментов и результаты, связанные с потоками работ, в MyExperiment также отданы на откуп внешним редакторам. В частности, Taverna поддерживает включение в спецификацию потока работ тестового примера для исполнения. Для подтверждения воспроизводимости результатов этого недостаточно, так как невозможна спецификация различных случаев и альтернативных путей прохождения потока работ.

В среде MyExperiment нет требования независимости методов от источников данных или возможности подмены источников, и в коллекции есть множество потоков работ, которые по своей сути являются не реализациями методов, а сервисами, предоставляющими данные из специфических источников данных по некоторым входным параметрам.

Хотя MyExperiment декларирует расширяемость онтологии, на которой построена схема информационной системы, на деле связи спецификаций потоков работ с какими-либо описаниями предметной области исследования сделать посредством существующих интерфейсов невозможно. В среде поддерживаются только вербальные пояснения к потокам работ и теги, и обеспечивается возможность поиска по ним.

В Taverna поддерживаются спецификации происхождения данных. Однако предназначены метаданные о происхождении только для записи пути прохождения данных внутри исполненного потока работ. Для достоверной проверки возможности повторного использования данных этого явно недостаточно, так как невозможно отследить историю их получения и преобразования от момента создания. К тому же доступа через интерфейсы MyExperiment к имеющимся данным о пути преобразования данных в потоке работ нет.

Проект wf4ever [4] предоставляет набор средств для поддержки повторного использования, проверки применимости, воспроизводимости и других свойств потоков работ. Среди описаний в проекте возможно специфицировать происхождение, внутреннюю структуру потоков работ, возможности доступа, жизненный цикл, развитие, многоверсионность и другие аспекты. Потоки работ могут проверяться на полноту, непротиворечивость, доступность и совместимость источников данных. Для этого предоставляются необходимые структуры данных и интерфейсы пользователя. В данном проекте в качестве экспериментальной базы взята коллекция потоков работ MyExperiment. Спецификации предметов исследования и потоков работ можно импортировать из MyExperiment, дополнить спецификациями, предоставляемыми проектом, и использовать набор сервисов для поддержки жизненного цикла потоков работ. Проект не предполагает больших продвижений в сторону семантических подходов к обеспечению доступа к потокам работ, а направлен больше на анализ самих потоков. В частности, одной из целей экспериментов ставится анализ того, почему многие из потоков работ в среде MyExperiment на сегодняшний момент попросту не запускаются.

4 Среда поддержки коллекции научных потоков работ MyExperiment

На примере среды разработки и публикации научных потоков работ MyExperiment мы будем показывать, какие метаданные необходимо добавлять к спецификациям потоков работ для обеспечения их повторного использования и воспроизводимости результатов. Поэтому более подробно остановимся на реализации сред MyExperiment

Для хранения метаинформации о потоках работ в среде MyExperiment используется база данных, схема которой специфицирована набором модулей онтологии. В этих модулях определены средства описания внутренней структуры накапливаемых потоков работ, спецификации пользователей, групп, аннотаций и других необходимых метаобъектов. Рассмотрим часть из них, представляющую интерес для данного исследования.

Для хранения метаобъектов о различных видах компонентов потоков работ создано базовое понятие WorkflowComponent. Его подпонятие NodeComponent описывает узлы потоков работ. Разновидности узлов представлены понятиями: Source – узлы-источники, приносящий в поток работ данные на обработку, Sink – узлы окончания потока работ, в которые приходят данные результатов исполнения потока работ., и Processor – узлы, исполняющие сервисы обработки данных. В свою очередь, типы исполнительных узлов описываются подпонятиями. В частности, WSDLProcessor соответствует вызову веб-сервиса. DataflowProcessor специфицирует вложенный поток работ, также состоящий из компонентов. Данные, Входы, выходы и соединения каждого узла в потоке работ описываются понятиями Input, Output и Link соответственно и объединяются базовым понятием IOComponent.

Объект исследования в MyExperiment представляет собой контейнер, содержащий файлы (например, данные, документы), внешние ссылки и потоки работ. Для хранения потоков работ как целостного объекта служат понятие AbstractWorkflow и его подпонятия Workflow и WorkflowVersion. Аналогично спецификациям файлов соответствуют понятия AbstractFile с подпонятиями File и FileVersion. Такая организация позволяет создавать многоверсионные объекты.

Понятия файлов и потоков работ объявляются имеющими суперпонятия Annotatable. С помощью этого понятия с ними могут быть связаны несколько видов аннотаций, среди которых комментарии, цитирования, теги и другие. Теги используются в качестве описания потоков работ и файлов для поиска в коллекции MyExperiment.

Сами метаобъекты, описывающие потоки работ, хранятся в реляционной базе, но реализована генерация их представления в модели RDF как экземпляров онтологии MyExperiment. Каждый метаобъект имеет в системе свой уникальный идентификатор URI. Например, идентификатор конкретного потока работ выглядит так: http://www.myexperiment.org/workflows/3514/.

Для разработчиков приложений над MyExperiment доступны несколько интерфейсов. К метаинформации MyExperiment можно задавать http-запросы через REST-интерфейс. Java-интерфейс MyJPI представляет собой REST-интерфейс, обёрнутый в классы языка Java. Наконец, реализован интерфейс точки доступа SPARQL, позволяющий задавать запросы к метаинформации MyExperiment и получать RDF-данные в соответствии со схемой, заданной онтологией, в нескольких форматах с учётом или без учёта автоматического вывода по правилам RDF Schema.

Однако все упомянутые интерфейсы имеют ограничение: в них не реализован доступ к внутренней структуре потоков работ, несмотря на то, что она определяется онтологией как компоненты потоков работ. Посредством программных интерфейсов можно получить ссылку на поток работ как файл Taverna. Этот файл подлежит разбору уже средствами Taverna для получения данных о внутренней структуре потоков работ. Это означает, что в рамках запроса на получить внутреннюю структуру потока работ не удастся.

В составе объектов исследования, помимо файлов (документации, данных), ссылок, потоков работ и аннотаций, поддерживаемых в MyExperiment, для обеспечения требований, изложенных в разделе 2, должны содержать также исчерпывающие наборы тестов, учитывающие

различные ситуации, и соответствующие данные результатов тестов при разных входных условия.

Таким образом, для создания среды исследований, обеспечивающей семантический поиск методов, повторное использование и воспроизводимость, в MyExperiment требуется расширение интерфейсов доступа к структуре потоков работ и поддержка систем тестов с результатами. В целом, это возможно, так как MyExperiment является проектом с окрытым кодом. Однако на данном этапе исследование проводилось с использованием оригинального сервера MyExperiment, соответственно, средства со стороны MyExperiment не менялись.

5 Расширение состава метаданных, сопровождающих публикуемые данные и научные методы

Для поиска объектов исследования, релевантных решаемой задаче, в MyExperiment предназначены только их текстовые описания и аннотации тегами. Причём связаны они, могут быть только с потоками работ в целом или файлами, исходя из их суперпонятия Taggable. Для коллекции методов и потоков работ, обеспечивающей их повторное использование, этого, безусловно, недостаточно.

Мы производим расширение состава хранимых метаданных об объектах исследования, потоках работ и их компонентах, для реализации семантических подходов работы с методами предметной области. Спецификации расширенного состава метаданных оформляются в виде набора онтологий разного назначения. Описанные онтологические модули находятся в открытом доступе по адресу: http://ontology.ipi.ac.ru/ontologies/astront, – и могут использоваться для накопления метаданных в соответствии с их определениями. Для хранения метаданных, связанных с конкретными метаобъектами MyExperiment, используется отдельная база экземпляров RDF.

Для реализации семантических подходов к поиску потоков работ, релевантных решаемой задаче, их повторному использованию и обеспечению воспроизводимости, в первую очередь, необходимо развивать спецификации предметной области, в которой собирается коллекция методов. Поиск потоков работ, отвечающих требованиям задачи, необходимо связывать с онтологией предметной области, которой принадлежит коллекция и в которой решается задача. Для этого метаобъекты, описывающие потоки работ, объявляются экземплярами классов понятий онтологии предметной области. Отнесение метаобъекта к классу понятия в терминах онтологий реализуется посредством отношения rdf:type. Для более сложных описаний в терминах онтологии метаобъекты могут становиться экземплярами неименованных классов, определённых как подпонятия понятий онтологии, но без введения новых понятий и свойств в онтологию.

Мы рассматриваем предметную область звёздной астрономии, включающую понятия одиночных звёзд, кратных систем звёзд. С ними связаны модули с описанием понятий астрометрии, фотометрии, астрофизики как понятий смежных областей. Эти модули используются в большинстве задач в области астрономии вне зависимости от того, какие задачи они решают.

В частности, в модуле астрометрии определены следующие понятия:

- Coordinate
- CoordinateSystem
- EquatorialCoordinateSystem
- CoordinateSystemComponent
- Epoch
- RightAscension
- Declination
- и другие.

Понятия имеют иерархию, описание структуры с помощью связей и ограничений.

В онтологию предметной области включены также более специфические модули, определяющие знания о парах и компонентах кратных звёзд, параметрах орбит двойных звёзд, параметрах кривой светимости затменных звёзд и других. Такие модули используются в более узких классах задач, в частности, связанных с определёнными видами астрономических объектов.

В качестве примера отнесения данных или компонентов потоков работ к понятиям онтологии предметной области, метаобъект с данными о координате прямого восхождения (RA_J2000) астрономического объекта может быть связан с понятием онтологии RightAscension, но для более точного описания такой метаобъект должен стать экземпляром выражения (подпонятия) в терминах онтологии, ограничивающего класс множеством экземпляров *x* таких, что x принадлежит Right Ascension, и существует координата *y*, система координат у которой экваториальная, и у которой есть компоненты: *x* и эпоха, равная J2000. Выбор простого или более точного стиля описания метаданных в дальнейшем влияет на качество поиска метаобъектов в терминах онтологии.

Наряду с модулями онтологии предметной области в нашем подходе спецификации метаданных пополняются также специализированными онтологиями, описывающими требования к происхождению данных, их качеству и среде исполнения.

В качестве онтологии происхождения данных используется в соответствии с рекомендацией W3C онтология PROV-O [2]. В её основе лежат понятия агента (Agent), деятельности (Activity) и сущности (Entity). Агентами могут быть человек (Person),

организация (Organization) или программа (SoftwareAgent). Вариации отношений их экземпляров друг с другом описывают различные события и ситуации, которые необходимо фиксировать при преобразовании, перемещении, изменении статуса данных. Например, метаданные об исходных данных, которые использовались процессом, выражается отношением used, связывающего агента и деятельность; информация об инструменте, который был использован для генерации результата, выражается отношением wasAttributedTo, связывающего сущность и программу и так далее. Посредством такой онтологии можно задавать метаданные об авторстве и принадлежности данных и методов, прослеживать историю преобразования данных от первоначального источника до текущего состояния, сопровождать реальные данные и методы другой подобной информацией.

Приведём пример спецификации происхождения данных для потока работ wf3514, обращающегося к внешнему сервису resolve_coordinates (Sesame Name Resolver) для локализации астрономического объекта на небе по его имени. Результирующие данные потока resolve_coordinates_outputTable могут содержать информацию в виде триплетов об инструменте, которым созданы данные и о потоке работ:

```
wf3514:resolve_coordinates
    rdf:type prov:SoftwareAgent .
wf3514:resolve_coordinates_outputTable
    rdf:type prov:Entity;
    prov:wasAttributedTo
        wf3514:resolve_coordinates;
    prov:wasGeneratedBy wf3514:wf3514 .
```

Ещё одна часть спецификации необходимых метаданных, онтология качества данных DQ [5], содержит набор факторов качества данных, определяемых измерениями в многомерном пространстве значений и метриками качества в этих измерениях. В качестве примера взяты измерения полноты данных (Completeness), объёма данных (Data Volume), возраста данных (Timeliness), точности (Accuracy), целостности (Consistency), меры доверия (Confidence). Состав измерений и метрики для их реализации сильно зависят от предметной области исследования. С одним объектом может одновременно быть связано несколько значений качества в разных измерениях. Экземпляры понятий данной онтологии связываются с потоками работ и файлами в целом, любыми компонентами потоков работ, сервисами и их параметрами, а также с самими данными. Метрики оценки качества также могут различными, но они согласовываются и специфицируются сообществом, работающим в предметной области.

Спецификации сред воспроизведения также могут требовать определения некоторой структуры метаданных. Однако, данные, необходимые для обеспечения воспроизводимости экспериментов, в многом выразимы средствами онтологии происхождения данных.

Также в среде MyExperiment требуется разработка поддержки систем тестов. До сих пор они описываются только некоторыми исследователями и неформально, в поле описания потока работ, либо в файлах, включённых в коллекцию объекта исследования. После реализации такой поддержки входные и выходные данные тестов, должны связываться

Для соответствия разработанным требованиям к публикации научных методов необходимо обеспечение определённых метаобъектов MyExperiment метаданными в терминах упомянутых онтологий.

Метаданными в терминах онтологии предметной области должны сопровождаться:

- файлы, потоки работ как целостные объекты;
- входные узлы в качестве предусловий;
- выходные узлы в качестве спецификаций их постусловий;
- узлы обработки данных;
- их входы и выходы.

Таким образом, производится описание семантики компонентов потоков работ в онтологии, на основе которого появится возможность поиска потоков работ, релевантных задачам, по понятиям, соответствующим потокам в целом, по соответствию семантики входных и выходных узлов, по семантике узлов обработки, по семантике блоков и потоков данных внутри потоков работ. Помимо поиска появляется возможность верификации потоков работ и их использования.

Метаданными в терминах онтологии происхождения сопровождаются:

- сами потоки работ как описания научных методов, требующих прояснения происхождения;
- обрабатывающие компоненты потоков работ как определённые научные сервисы;
- данные, направляемые на обработку в потоке работ, находящиеся в процессе обработки и результирующие.

Любые данные, входящие в объект исследования в виде файлов или участвующие в потоках работ, должны быть соотнесены с онтологиями предметной области, происхождения, качества данных.

Некоторые аспекты качества данных могут быть связаны с методами и потоками работ в целом как спецификациями качества, ожидаемого от работы методов.

Тесты и их результаты снабжаются связями с онтологией предметной областью, причём особенности различных ситуаций, представляемых разными тестами, желательно отражать в

ограничениях понятий. Результаты тестов должны иметь метаданные происхождения, связанные с историей выполнения тестов в потоках работ.

6 Применение метаданных для обеспечения повторного использования и воспроизводимости результатов работы научных методов

Онтологии предметной области исследования, происхождения данных, качества данных, сред исполнения фактически определяют разные ракурсы взгляда на описываемые объекты исследования и научные методы. Метаданные в терминах определённых онтологий – не зависимые друг от друга проекции на объект исследованияв контектсе знаний данной онтлогии. Запросы в терминах каждой из этих онтологий, могут выдать потоки работ или их компоненты, соответствующие определённым требованиям с точки зрения конкретной онтологии.

Для хранения метаданных используется база RDF-триплетов на основе Jena. В ней хранятся экземпляры в соответствии со структурой, определённой описанными выше онтологиями. Для работы с базой экземпляров используется язык запросов SPARQL.

При решении научных задач и поиске релевантных задаче реализаций научных методов возникнет необходимость предъявления требований одновременно с нескольких ракурсов. Таким образом, понадобится обрабатывать запросы, включающие конъюнктивно требования одновременно в терминах нескольких онтологий.

Пример запроса.

```
prefix rdf:
<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
prefix mecomp:
<http://rdf.myexperiment.org/ontologies/components/>
prefix astrobjects:
<http://ontology.ipi.ac.ru/ontologies/astrobjects.owl>
prefix prov:
<http://www.w3c.org/ns/prov#>
SELECT ?workflow WHERE
{ ?output rdf:type astrobjects:AstrObject .
  ?output prov:wasGeneratedBy ?workflow .
  ?output prov:wasAttributedTo :resolve_coordinates.
  SERVICE <http://rdf.myexperiment.org/sparql>
  { ?output mecomp:belongs-to-workflow ?workflow .
    ?output rdf:type mecomp:Sink }
}
```

Такой запрос к базе RDF-экземпляров выясняет, какие потоки работ из коллекции MyExperiment возвращают астрономические объекты, обращаясь за ними в сервис resolve_coordinates (с точки зрения онтологии происхождения данных). Соответственно, он включает в себя требования к выборке из точки доступа MyExperiment метаобъектов класса Workflow, к которым относятся метаобъекты класса Sink. В языке запросов SPARQL для обращения к распределённым точкам доступа используются средства федеративных запросов с помощью конструкции SERVICE. и прямого указания адреса точки доступа MyExperiment. Остальные требования относятся к тем же RDF-ресурсам, но опрашивается база RDF-экземпляров с метаданными. Одно из них относится к метаданным в терминах онтологии астрономии, а именно, принадлежность выходных данных потока работ понятию AstrObject,. А другое – к метаданным в терминах онтологии происхождения данных, а именно, какой инструмент используется для генерации данных. Таким образом, один запрос использует термины MyExperiment, термины онтологии предметной области и термины происхождения данных, а результатом запроса являются найденные в коллекции научных методов потоки работ, релевантные сформулированным в запросе требованиям.

Подобное использование метаданных позволяет решать многие задачи, связанные с семантическим подходом к обеспечению интероперабельности научных методов, их повторным использованием и обеспечением.

На основе метаданных о связи с предметной областью можно решать задачи поиска релевантных методов:

- по понятиям, связанным с потоками работ в целом;
- по соответствию требованиям задачи понятий, связанных с входными и выходными данными потоков работ, то есть, по спецификации в терминах онтологии того, что мы имеем и того, какие данные мы имеем, и того, что хотим получить в результате работы метода;
- по присутствию в потоке работ компонентов-стадий, которые необходимы для решения задач;
- по другим возможным критериям, формулируемым с использованием понятий предметной области.

Возможно производить семантический контроль используемых методов и принятых решений:

- проверку семантики данных между всеми компонентами потока работ;
- проверку корректности использования подпроцессов по их входным и выходным параметрам;
- соответствие семантики входного компонента семантике входных данных, либо выходных данных выходным компонентам;
- соответствие семантики данных, проходящих из выхода одного компонента на вход другой, по принципу спецификаций пред- и постусловий:

постусловие выхода предыдущего компонента должно быть строже предусловия входа последующего компонента.

Видно, что обеспечение семантической интероперабельности за счёт соотнесения задач, данных и методов со знаниями предметной области является основой для обеспечения повторного использования научных методов.

Обеспечение качества данных, достоверности, полноты и других аспектов, связанных с надёжностью данных и методов, реализуется с помощью использования онтологиями качества данных и их происхождения.

Возможности метаданных происхождения данных также сложно переоценить. С их помощью осуществляется:

- контроль реальных источников данных и их качества в соответствии с требованиями задачи;
- контроль за соответствием требованиям решения задачи используемых открытых реализаций научных методов
- контроль прохождения тестов по определённому пути в потоках работ и соответствия качества получаемых данных требованиям задачи
- проверка требований воспроизводимых экспериментов к исполняемой среде.

Таким образом, воспроизводимости результатов способствует ведение метаданных происхождения для каждой манипуляции, производимой при прохождении экспериментов. При воспроизведении результатов возможно отследить обратную цепочку манипуляций и повторить её.

Спецификации требований к исполняемой среде, необходимой для проведения эксперимента, формулируются в терминах происхождения данных.

7 Заключение

В статье проанализированы требования к средам поддержки научных исследований для обеспечения повторного использования научных методов и воспроизводимости результатов их работы. Предложен набор метаданных, которые должны сопровождать данные и методы с этой целью. Метаданные определяются в терминах онтологий и включают привязку описаний научных методов и потоков работ к знаниям предметной области и также снабжение информацией о происхождении и качестве данных. Показан путь использования этих метаданных.

Литература

[1] The Fourth Paradigm: Data-Intensive Scientific Discovery. Tony Hey, Stewart Tansley, and Kristin Tolle, Eds. Microsoft Research, Redmond, WA, 2009. 286 pp.

[2] The PROV Ontology. W3C Recommendation. – W3C, 2013. – URL: http://www.w3.org/TR/prov-o/.

[3] VizieR. – URL: http://vizier.u-strasbg.fr/cgi-bin/VizieR

[4] Wf4Ever project. – URL: http://www.wf4ever-project.org/

[5] S. Geisler, S. Weber, Ch. Quix. Ontology-based data quality framework for data stream applications. // Proc. of the 16th International Conference on Information Quality (ICIQ-11). – 2011.

[6] Goble C. A., De Roure D. C. myExperiment: social networking for workflow-using e-scientists // Proceedings of the 2nd workshop on Workflows in support of large-scale science. – ACM, 2007. – С. 1-2.

[7] D. Hull, K. Wolstencroft, R. Stevens, C.A. Goble, M.R. Pocock, P. Li, T. Oinn. Taverna: A tool for building and running workflows of services, Nucleic Acids Research, 34 (Web-Server-Issue), 2006, pp. 729–732.

[8] L. Moreau. Provenance-Based Reproducibility in the Semantic Web. // Web Semantics: Science Services and Agents on the World Wide Web. – 9, (2). – 2011. – P. 202-221.

[9] Shadbolt N., Hall W., Berners-Lee T. The semantic web revisited //Intelligent Systems, IEEE. – 2006. – Т. 21. – №. 3. – С. 96-101.

[10] Walton N. A. et al. AstroGrid: A place for your science //Astronomy & Geophysics. – 2006. – Т. 47. – №. 3. – С. 3.22-3.24.

[11] Yu L. Linked open data //A Developer's Guide to the Semantic Web. – Springer Berlin Heidelberg, 2011. – С. 409-466.

[12] А. Е. Вовченко, Л. А. Калиниченко, С. А. Ступников Семантический грид, основанный на концепции предметных посредников. // Труды четвертой международная конференция "Распределённые вычисления и Грид-технологии в науке и образовании" Grid2010, Дубна, ОИЯИ, 2010. – с. 309-318.

Scientific Methods Metadata for Provision of the Methods Reuse and Result Reproducibility

N. A. Skvortsov, D. O. Briukhov, L. A. Kalinichenko, D. Kovalev, S. A. Stupnikov

Data-intensive sciences are characterized by the constantly growing needs for specific data analysis methods intended for producing new knowledge related to the investigated areas. Development of new data analysis methods becomes a significant, inseparable part of research. Digital preservation, reuse and reproducibility of computer experiment results become inherent attributes of scientific discovery. The paper investigates metadata structure to be attached to the processes specifying or implementing scientific data analysis methods for their reuse and result

reproducibility. Process components and data are referred to the domain concepts and need to be supplied with the information about data provenance and quality. Specific test collections are needed to describe kinds of cases in which methods must behave in an anticipated way. Using the open myExperiment environment organizing and providing access to the collection of scientific workflows as an illustration, we demonstrate how the extension of its metadata could have allowed to organize the semantic search for methods relevant to a problem, to verify interoperability, reusability and reproducibility of processes implementing the methods.

.

Core semantic model for generic research activity[€]

© Vasily Bunakov
Scientific Computing Department, Science and Technology Facilities Council,
Harwell OX11 0QX, United Kingdom
vasily.bunakov@stfc.ac.uk

Abstract

A simple research activity model is suggested that is agnostic to research domain and allows independent curation of the research information lifecycle by a variety of its stakeholders with a potential to further link individual activities into meaningful research provenance or research value chains. We consider the drivers for conceiving the model, its main aspects, an RDF manifestation of it, a particular business case for its application, and discuss its potential for future applications.

1 Introduction

Different stages of the research lifecycle in natural sciences as well as in social and economic research produce multiple data artefacts under control of different data management solutions and software platforms. (We use the term "data" here and there in a broad sense: not necessarily numeric data resulting from measurements but research proposals, software components, configuration files, electronic publications, etc.) Data curators working in a particular research domain tend to develop a specific metadata model that aims to cover the entire research lifecycle from the research inception to the research outputs dissemination. Such a metadata model quite often serves as a foundation for the design of the actual information systems and services. The example of a comprehensive metadata model for the research performed at large facilities like synchrotrons, powerful lasers or neutron sources is the Core Scientific MetaData model [5]; the example in social research is DDI-Lifecycle [7].

Proceedings of the 15th All-Russian Conference "Digital Libraries: Advanced Methods and Technologies, Digital Collections" — RCDL-2013, Yaroslavl, Russia, October 14-18 2013.

[€] This work is related to the ENGAGE project www.engage-project.eu and the projects of PaNdata collaboration www.pan-data.eu supported by the EU 7th Framework Programme for Research and Technological Development. The author would like to thank his colleagues in ENGAGE and PaNdata for their input for this paper although the views expressed are the views of the author and not necessarily of the projects.

Substantial effort of renown information experts has been spent in order to extend some established metadata models with new semantic features; the example in social research will be DDI semantic modelling ([8], [9]). The richness and the expressivity of metadata model that has evolved through decades can be considered a limitation that makes it harder to agree on what should constitute the "true" semantic representation, or what format of it should be a "canonical" one. Also the attempts to transform the entire domain-specific metadata model into semantic representation, and then offer it for common adoption and data linkage may contradict the social nature of Linked Data as its curation can be reasonably considered an incremental and opportunistic effort of multiple parties (as brilliantly illustrated by [1]).

This is not to say that semantic modelling of the entire research domain is not sensible or do not have a potential for implementation. Collaborative projects of a multinational scale such as PaNdata-ODI ([2], also see under [16]) consider semantic representation of the popular domain-specific metadata model [5] with the purpose of system integration. The motive for this consideration is that, despite the actual information systems in different research centres may be based on the implementations of the same generic metadata model and even on the same software platform for data catalogue [14], the practices of the catalogue configuration, the interpretation and the use of the model elements, and hence the actual semantics of these elements may vary dramatically. A common semantic layer, probably in the form of ontology, is considered then a viable architecture solution that should allow retaining the existing local practices of data cataloguing and at the same time, should give the IT teams an ability to meaningfully integrate distributed data and services.

That semantic layer, however, will require an inclusion into a certain best practices framework to sustain it through time [4], otherwise divergent business needs and business practices of the collaboration participants can make a thoroughly designed semantic model obsolete the next day after its implementation in a real IT solution. Keeping a comprehensive semantic model actual can be quite an expensive endeavour with substantial overheads on continuous business analysis and communication with multiple parties.

Another concern about the attempts of semantic representation of comprehensive metadata models is a tendency for them to reflect the information needs of

only a few types of the research lifecycle stakeholders: this is commonly Researchers and Data Archivists. The information needs of other stakeholders from Funding, Industry, or Education are often under-represented. To resolve this issue, one can take two approaches:

A) As a responsible information curator, conduct thorough business analysis of the research lifecycle stakeholders' types and their information needs then incorporate the knowledge acquired into a comprehensive model that, in order to be effective, should be validated by the stakeholders themselves (then, ideally, permanently amended).
B) Give different stakeholders a reasonable modeling means to express their role in the research lifecycle so that each of them becomes an information curator who cares about the quality and the actuality of her contribution into the shared pool of information.

The latter approach seems more adequate in the present situation when the advance of Linked Data principles allows various stakeholders to meaningfully model their part of information universe, also re-use the results of similar modeling effort made elsewhere.

We suggest a small but quite universal "core" model in the spirit of Linked Data principles [1] with low barriers for its adoption and use for semantic annotation of the research activity in different local information contexts, with their further inclusion into a global information context. We think that such a model should not focus on data but on common patterns of research activity observed in different research domains (for which we give examples further in this paper); various data then can be considered artefacts or "footprint" of different types of research activity.

2 Research activity model

2.1 Types and common patterns of research activity

Research lifecycles analyzed and structured by digital curators in the respective research domains can be a good source for discovering granular research activities and their interrelations. In this work, we consider two lifecycles: in facilities science[1] and in social research; they are most relevant to the projects which contributed to the development of our model ([11], [16]) and their respective research domains stay quite far apart so may help us with testing our model universality.

Lifecycle in facilities science that underpins CSMD model [5] includes the submission of a research proposal to the facility user office in order to get the facility resource for research (e.g. beam time on synchrotron); the further approval of the proposal by the facility's user office; experiment scheduling; conduct of the actual experiment with data collection; data storage; data analysis; and eventually publishing research results with record keeping for them. Beyond this lifecycle that is supported by facility itself, there is research funding activity, or research policy making, or the researchers' social communication that all can be considered elements of a larger "research value chain".

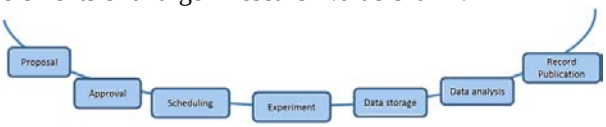


Figure 1. Research lifecycle in facilities science (as captured by CSMD model).

The lifecycle of social research that underpins DDI-Lifecycle model [7] includes the formulation of the study concept, further data collection, its processing, archiving, distribution, discovery, analysis, and repurposing. Funding, or policy making, or social communication, despite there are some placeholders for references to these types of activity – are again beyond the immediate scope of DDI.

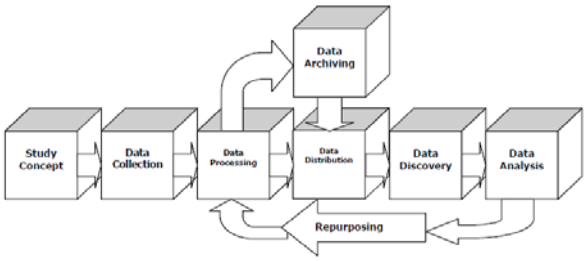


Figure 2. Research lifecycle in social science (as captured by DDI-L model).

Each activity yields certain outputs, e.g. in facilities science, the research proposal preparation results in the investigation (experiment) description, data analysis yields derived data etc. Previous activity may provide an input for other activity or give it a context, e.g. it is quite common for researchers to refer to the previous investigations (experiments) when they apply for a new investigation to be conducted at the same facility.

Despite there are similarities between the two aforementioned lifecycles and between the roles of stakeholders involved in them, there are differences, too. Even more differences come up if we consider context or scope of each research activity, or means for their description that are present in each model. As an example, in facilities science, the scope of experiment can be understood by considering what samples or chemical substances have been under investigation; in social research, it can be meaningful parameters describing the human audience which the study has been aimed upon. Not these details that may be different but the very presence of Context and Scope, as well as the Inputs and Outputs for the research activity, or Actors who perform it, or Effects of the research do represent a common pattern – very generic but universal

[1] For the sake of clarity, we use the term "facilities science" for the research performed on large-scale scientific instruments (synchrotrons, powerful lasers and alike) by visitor teams or individual researchers who obtain, via the application process, access to the common facility resource in order to conduct their experiments or observations, and to collect the resulting data.

across research fields.

These patterns are common not only across different research domains for the similar types of research activity (when we draw parallels e.g. between facility science Experiment and social research Study); this is also the case for different types of research activity within the same lifecycle, e.g. funding or data analysis or record publication have their Inputs and Outputs, their Actors, Effects, Context (Conditions) and Scope.

These basic patterns contribute to a reasonable model that should not be too burdensome for the respective stakeholders (or information specialists working for them) to apply, yet is expressive enough to promote the principles and best practices of Linked Data in various research domains. We consider a potential for such an application below in the section devoted to a particular business case; in the meanwhile, we are going to formally introduce the major aspects of a generic research activity, and suggest a practical RDF-based manifestation for them.

2.2 Generic research activity (research activity "cell")

We deem important the following aspects of a generic research activity:

Aspect	Description	Examples	
		Research per se	Research data analysis
Input	Something that is taken in or operated on by Activity	Previous research	Raw data
Output	Something that is intentionally produced by Activity	Raw data	Derived (analyzed) data
Scope	Something that Activity is aimed at or deals with	Sample properties	One or more experiments
Condition	Something that affects or supports Activity, or gives it a specific context	Scientific instrument	IT environment
Actor	Something or somebody who participates in Activity	Investigator	Data analyst
Effect	Something that is a consequence of Activity	Environment pollution	New software module

Schematically, the granular research activity can be represented by the following diagram:

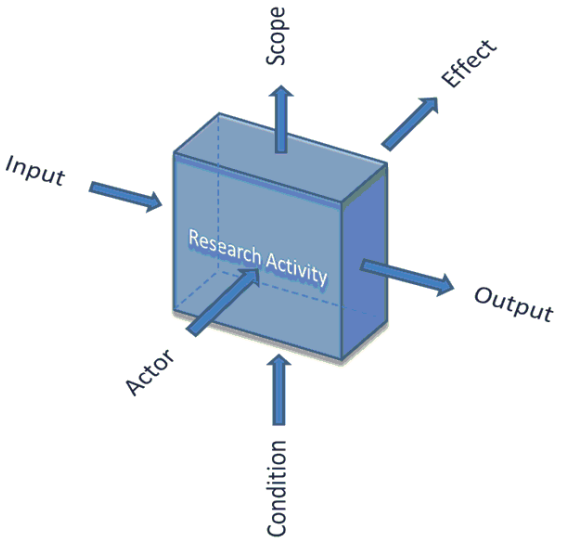


Figure 3. Research activity "cell".

Research activities can be combined as "cells" in chains where Output of one can be an Input to another but in fact, the model allows other sorts of links between activities. As an example, a piece of regulation such as data management policy can be an Output of one activity (policy making), and a Condition that affects another activity (research per se); a new software module that is a side Effect of a certain activity (data analysis) can be a non-human Actor that participates in other activity (e.g. automated indexing of experimental data). This shows that activity aspects in fact do not have "types": a modeler can use and combine them as dictated by the semantics of the respective subject area.

This view is inspired, to some extent, by SADT activity model [17] with its idea of combining activities into the hierarchy or a grid but is quite different by introducing some other activity aspects and not imposing their typization. Also SADT promotes a top-down approach to structured analysis and systems design when we suggest a bottom-up approach that allows combining the granular activities in more complex information structures.

Compared to other project-driven attempts to model research activity ([10], [15]) our model is going to be simpler, more universal, and deliberately aimed at semantic modeling of a granular activity rather than of the entire research lifecycle thus providing a "building block" for a more sophisticated information modeling as and when required.

2.3 RDF manifestation of activity model

The outlined model may imply different manifestations; we feel that one expressed in RDFS Plus (RDF Schema with a few OWL terms) has a good potential for adoption by information curators and implementation in real IT solutions. This paper Appendix suggests the

RDFS Plus manifestation of the activity model that can be extended by domain specific entities and properties. As an example, an information modeler in facilities science might want to extend the model as follows:

```
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.
@prefix am: <http://example.org/stuff/ActivityModel#>.
@prefix rm: <http://example.org/stuff/ResearchModel#>.
# For Activities
    rm:Research rdfs:subClassOf am:Activity .
    rm:Experiment rdfs:subClassOf rm:Research .
# For Conditions
    rm:Condition rdfs:subClassOf am:Condition .
    rm:Regulation rdfs:subClassOf rm:Condition .
    rm:DataManagementPolicy rdfs:subClassOf rm:Regulation .
# For Output
    rm:Output rdfs:subClassOf am:Output .
    rm:Publication rdfs:subClassOf rm:Output .
    rm:Dataset rdfs:subClassOf rm:Output .
# For Scope
    rm:Scope rdfs:subClassOf am:Scope .
    rm:ExperimentalTechnique rdfs:subClassOf rm:Scope .
    rm:SubjectCoverage rdfs:subClassOf rm:Scope .
# For properties
    rm:activity_location rdfs:subPropertyOf am:hasScope .
    rm:activity_subject rdfs:subPropertyOf am:hasScope .
```

The user of the information system where the RDF data prepared according to our model is published can then use reasonable SPARQL requests to inquire for different aspects of research activities, e.g. trying to realize first how much research output, and how much of each type is out there:

```
SELECT ?output_type (COUNT(?output) as ?total)
WHERE {?output_type rdfs:subClassOf am:Output .
   ?output a ?output_type .
     }
GROUP BY ?output_type
```

or try to discover the chains of interrelated activities:

```
SELECT ?previous_activity ?current_activity
WHERE {?previous_activity am:hasOutput ?output .
     ?output am:inputFor ?current_activity .}
```

User may be familiar with just our activity model knowing very little about a certain research domain at start, then accumulating more and more knowledge through sensible incremental requests. In case the information modeler, in addition to our basic activity model, has followed good practices of data curation so that e.g. instances of Scope or Condition subclasses are not literals but dereferenceable URIs, the User will have even more opportunities of getting familiarized with the semantics of a particular research domain. When we tell of "User" we of course mean the software agents, too, as the prospect of employing them is a strong incentive for any semantic modeling.

2.4 Business case for semantic categorization and annotation of existing metadata

As we mentioned, it may not be easy to give birth to the semantic representation of a comprehensive metadata model because of its richness and complexity, and because of substantial overheads for communication among information curators who apply the model in different contexts. Another observation is that detailed metadata records may in fact represent different activities performed by different stakeholders of the research information lifecycle – while the records that in fact circulate in the information management solutions are focused on particular types of stakeholders only and support their specific roles in the first place. A certain stakeholder, e.g. Data Librarian or Data Archivist may claim that Her information management solution is focused on *data* in pursuit of some common interest when, in fact, the information management solution primarily supports this particular stakeholder specific *role* in the information lifecycle with only some types of other stakeholders well served.

As an example, DDI [7] suggests some means to model information about funding but European funding bodies are likely to use their own information systems, many of them based on CERIF standard [6]. So the richness and expressivity of DDI, as well as the actual information systems based on it are in fact aimed at researchers in social science and data archivists, not at funders who are likely to have their own information systems based on other metadata standards, and not at other types of stakeholders in Business, Education, or researchers in other research domains.

We feel that it will be more productive to admit this natural attitude of the information management solutions and their owners to cater for only one or a few roles; it may be better to provide a reasonable means to model different roles and their activities on a granular level than try to capture an elusive information context in more and more complex versions of a comprehensive semantic model. If we take the existing records in a certain rich metadata format, this approach results in categorization and annotation of the entire metadata records with other metadata based on a smaller but semantically meaningful and universal information model – like our activity model.

Let us see how our core semantic model may serve DDI metadata categorization and annotation.[2] The analysis shows that one DDI record typically represents different types of research activity:

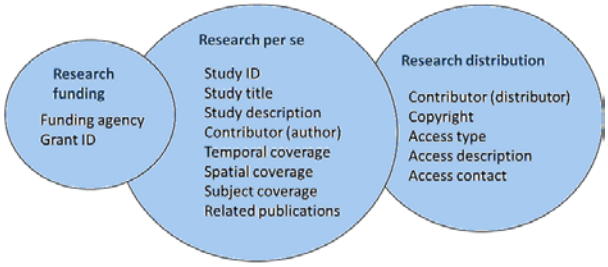


Figure 4. Research activities represented by a DDI record.

As we have identified different types of research activity, we can model them accordingly; we can also

[2] This approach was applied to DDI records harvested from the UK Data Archive and GESIS archive ([18], [13]) in the interests of the ENGAGE project [11] and was communicated in [3] as a prolegomenon to the generic model that we are presenting now.

identify specific Actors (Funding Agency, Author, Distributor), activity Outputs (Publication, Dataset), Scopes (Spatial Coverage, Subject Coverage) and Conditions (Copyright, Access Terms). Different granular activities will be modeled then with different amount of detail but we can enrich them with data from other information systems: for research funding – through funding agency portals, for research – through the project and the individual investigators' Web pages. This information enrichment should ideally be done by the Actors of the respective Activities (Funding, Research per se, Distribution) as they best understand the information context and the semantics of their business.

Our activity model then should allow curating the data and data context (metadata) in a distributed manner, and the combination of granular activities in sensible information context chains. This should eventually give us a more dispersed but a more complete description of the research discourse for a particular Study – more complete if compared to what the Data Archivist deemed valuable to capture and describe in a DDI record for the same. Our core model then serves as a "glue" to support the common information context and facilitate the interoperability of different digital curation frameworks that are operated by different Actors in support of their own Activities.

The existing well curated archives of DDI records can be considered then a valuable "fuel" to support the launch of the research discourse "Web" or "grid". The role-centric nodes of it will be performing their part of digital curation, with sharing its results via simple and commonly understandable semantic model that can be interpreted not only by data archivists or researchers in social science but by various stakeholders from other research domains, or business, or education, or policy making.

2.5 Conclusion

We outlined the motivation for why a simple model would be valuable for the semantic representation of a generic research lifecycle. We introduced the major aspects of the model, suggested an RDF manifestation for them and showed how the domain-agnostic requests might work for information discovery. We then considered a particular business case of applying the model to the existing rich metadata records in social science but there are more promising cases to consider.

One of the immediate candidates is facilities science with its CSMD metadata [5] that we already mentioned. The diverse business practices for using the existing mature data management solutions based on CSMD model [14] may become a barrier to the meaningful sharing of facilities science data as Linked Data. Our model then may be of help for the re-engineering of the existing data archives in spirit of Linked Data and Semantic Web principles, through semantic annotation of the CSMD metadata records (which may involve some decomposition, too, similarly to what we demonstrated for DDI metadata).

Another prospective area where we think our model may prove to be valuable is long-term digital preservation with its two well-known problems of the accountable data provenance and of the meaningful data representation for the future (and changing) community of data consumers. The ability of our model to combine individual data curation activities into the traceable chains of them, as well as its very focus on the Activity (with data being an artefact or footprint of it) may contribute to the satisfactory resolution of the data provenance problem. The model's data discovery capabilities based on standard information requests and profiles of them when it is enough for the User to be familiar with our basic semantic model in order to start the incremental knowledge discovery – may contribute to the meaningful data representation.

Also we find the multi-disciplinary and *distributed* curation, discovery and re-use of the research information to be in high demand; it is already in the agenda of a few actual European projects (see under [11], [12], [16]) and it is reasonable to expect more of them to come. The domain-agnostic nature of our model, as well as its very manageable core size and expandability where required let us hope for its application in some of the existing and future e-infrastructure initiatives.

3 Appendix: RDFS Plus manifestation of the activity model

```
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix am: <http://example.org/stuff/ActivityModel#> .

############### Core entities of Activity model ###############

# Comments are based on the Oxford dictionary, with some generalization or amendment where appropriate

am:Activity rdf:type rdfs:Class ;
        rdfs:label "Activity" ;
        rdfs:comment "Something that Actor does, or has done,
            or is going to do, or can do" .
am:Input rdf:type rdfs:Class ;
        rdfs:label "Activity Input" ;
        rdfs:comment "Something that is taken in or operated on
            by Activity" .
am:Output rdf:type rdfs:Class ;
        rdfs:label "Activity Output" ;
        rdfs:comment "Something that is intentially produced
            by Activity" .
am:Actor rdf:type rdfs:Class ;
        rdfs:label "Activity Actor" ;
        rdfs:comment "Something or somebody who participates
            in Activity" .
am:Effect rdf:type rdfs:Class ;
        rdfs:label "Activity Effect" ;
        rdfs:comment "Something that is a consequence
            of Activity" .
am:Condition rdf:type rdfs:Class ;
        rdfs:label "Activity Condition" ;
        rdfs:comment "Something that affects or supports
            Activity, or gives it a specific context" .
am:Scope rdf:type rdfs:Class ;
        rdfs:label "Activity Scope" ;
        rdfs:comment "Something that Activity is aimed at
            or deals with" .
```

```
########## Core properties of Activity model ##########

# am:hasInput or am:inputFor
# links Activity to its Input
am:hasInput owl:inverseOf am:inputFor .

# am:hasOutput or am:outputOf
# links Acttivity to its Output
am:hasOutput owl:inverseOf am:outputOf .

# am:hasActor or am:actorFor
# links Activity to its Actor
am:hasActor owl:inverseOf am:actorFor .

# am:hasEffect or am:effectOf
# links Activity to its Effect
am:hasEffect owl:inverseOf am:effectOf .

# am:hasCondition or am:ConditionFor
# links activity to its Condition
am:hasCondition owl:inverseOf am:ConditionFor .

# am:hasScope or am:ScopeOf
# links Activity to its Scope
am:hasScope owl:inverseOf am:scopeOf .
```

References

[1] Tim Berners-Lee. Open, Linked Data for a Global Community. A talk given on Gov 2.0 Expo, Washington, DC, 26 May 2010. http://www.gov2expo.com/gov2expo2010/public/schedule/detail/14247

[2] Juan Bicarregui, Vasily Bunakov, and Michael Wilson. PANdata international information infrastructure for synchrotrons: opportunity for collaboration. Presentation on the 19th Russian Synchrotron Radiation Conference (SR-2012), Novosibirsk, Russia, 25-28 June 2012. http://epubs.stfc.ac.uk/work-details?w=63074

[3] Vasily Bunakov. Semantic categorization of DDI metadata. Presentation on the 4th Annual European DDI User Conference (EDDI12), Bergen, Norway, 03-04 Dec 2012. http://epubs.stfc.ac.uk/work-details?w=64315

[4] Vasily Bunakov and Brian Matthews. Data curation framework for facilities science. In Proceedings of DATA 2013: the 2nd International Conference on Data Management Technologies and Applications, p.211-216, Reykjavík, Iceland, 29-31 July 2013.

[5] Brian Matthews et al., 2012. Model of the data continuum in Photon and Neutron Facilities. PaNdata ODI, Deliverable D6.1. http://pan-data.eu/sites/pan-data.eu/files/PaNdataODI-D6.1.pdf

[6] Common European Research Information Format. See under www.eurocris.org

[7] Data Documentation Initiative – Lifecycle Specification. http://www.ddialliance.org/Specification/DDI-Lifecycle/

[8] Semantic Statistics for Social, Behavioural, and Economic Sciences: Leveraging the DDI Model for the Web. Schloss Dagstuhl, September 11 – 16, 2011. http://www.dagstuhl.de/en/program/calendar/evhp/?semnr=11372

[9] DDI Lifecycle: Moving Forward. Schloss Dagstuhl, October 21 – 26, 2012. http://www.dagstuhl.de/en/program/calendar/evhp/?semnr=12432

[10] DARIAH-EU: Digital Research Infrastructure for the Arts and Humanities. http://www.dariah.eu/

[11] ENGAGE: An Infrastructure for Open, Linked Governmental Data Provision towards Research Communities and Citizens. http://www.engage-project.eu/

[12] EUDAT: European Data Infrastructure. http://www.eudat.eu/

[13] GESIS - Leibniz-Institut für Sozialwissenschaften. http://www.gesis.org/

[14] ICAT project. http://www.icatproject.org/

[15] Infrastructure for Integration in Structural Sciences (I2S2) Project. http://www.ukoln.ac.uk/projects/I2S2/

[16] PaNdata: Photon and Neutron Data Infrastructure. http://pan-data.eu/

[17] Structured Analysis and Design Technique. http://en.wikipedia.org/wiki/Structured_Analysis_and_Design_Technique

[18] UK Data Archive (for social sciences and humanities). http://data-archive.ac.uk/

Отображение графовой модели данных в каноническую объектно-фреймовую информационную модель при создании систем интеграции неоднородных информационных ресурсов*

© С. А. Ступников
ИПИ РАН,
Москва
ssa@ipi.ac.ru

Аннотация

В работе рассматривается отображение модели данных атрибутированных графов в каноническую информационную модель данных для создания систем виртуальной или материализованной интеграции неоднородных информационных ресурсов – СУБД, Веб-сервисов и т.д.

Целью работы является создание обоснованной теоретической базы для интеграции ресурсов, основанных на графовых моделях.

Работа выполнена при поддержке РФФИ (гранты 11-07-00402-а, 13-07-00579-а) и Президиума РАН (программа 16П, проект 4.2).

1 Введение

Роль данных в различных областях деятельности человека – научных исследованиях, здравоохранении, образовании, промышленности и т.д. – непрерывно растет в последние годы. Укрепляется новая парадигма в науке и информационных технологиях, связанная с интенсивным использованием данных – так называемая *четвертая парадигма* [16]. Развиваются новые информационные технологии, в которых данные становятся доминирующим фактором, новые подходы к концептуализации, организации и реализации информационных систем. При этом требуется не только создание методов и средств оперирования данными, объемы которых выходят за рамки возможностей современных технологий баз данных, но и разработка новых подходов, позволяющих справляться с разнообразием массово и хаотично развивающихся языков и моделей данных.

Данная статья продолжает исследования по унификации моделей, применяемых в системах с интенсивным использованием данных, для виртуальной или материализованной интеграции ресурсов при создании федеративных баз данных или хранилищ данных. К таким моделям относятся разнообразные NoSQL-модели; онтологические и семантические модели; графовые модели; модели, основанные на многомерных массивах и т.д.

Материализованная интеграция предполагает создание хранилища данных, в которое загружаются интегрируемые ресурсы. При этом данные из схемы ресурса преобразуются в общую схему хранилища.

Виртуальная интеграция обычно предполагает создание *предметных посредников,* образующих промежуточный слой между пользователем (приложением) и неоднородными информационными ресурсами. Данные из ресурсов не материализуются в посредниках: доступ к данным осуществляется при помощи запросов к посреднику в терминах федеративной схемы посредника. Эти запросы переписываются в частичные запросы над информационными ресурсами, затем исполняются на ресурсах. Результаты частичных запросов объединяются и выдаются пользователю также в терминах федеративной схемы [8].

Унификацией модели данных ресурса называется ее отображение в *каноническую информационную модель* (служащую общим языком в среде разнообразных моделей ресурсов), сохраняющее информацию и семантику операций языка манипулирования данными (ЯМД) [20]. Унификация моделей ресурсов является необходимым предусловием материализованной или виртуальной интеграции ресурсов, т.к. семантические отображения, связывающие федеративную схему и схемы ресурсов, нужно проводить в единой (канонической) модели [9].

Труды 15-й Всероссийской научной конференции «Электронные библиотеки: перспективные методы и технологии, электронные коллекции» — RCDL-2013, Ярославль, Россия, 14-17 октября 2013 г.

В ранее проведенных исследованиях изучались вопросы унификации NoSQL-моделей [21] и моделей, основанных на многомерных массивах [22].

В данной статье рассматривается еще один важный класс моделей данных – *графовые модели*. Исследования графовых моделей начались в середине 1980-х годов. Математическими основаниями для них послужила теория графов, а наибольшее влияние оказали так называемые *семантические модели* (например, модель «сущность-связь») [3]. Целью графовых моделей было преодоление ограничений, налагаемых традиционными моделями данных, связанных с представлением исходных графовых структур данных.

Основными отличительными чертами графовых моделей данных являются следующие [3]:

- данные и/или схема данных представляются в виде графов или структур данных, обобщающих понятие графа (гиперграфы или гипервершины);
- манипулирование данными выражается в виде трансформаций графов или при помощи операций, основными параметрами которых являются такие характерные графовые структуры и свойства, как пути, подграфы, связность и т.д.;
- ограничения целостности тесно связаны с графами как структурой данных. Так, ограничениями могут быть уникальность меток ребер и вершин, типизация ребер и вершин, ограничения на область определения и область значений свойств ребер и вершин.

Графовые модели данных применяются в тех случаях, когда информация о взаимосвязях между данными или их топологии является более важной (или настолько же важной), как сами данные. Поводом к использованию графовых моделей может быть также недостаточная выразительная сила языков запросов традиционных моделей. Наиболее распространенными примерами применения графовых моделей являются системы управления и анализа сложных сетей – социальных, биологических, информационных, транспортных, телекоммуникационных и других.

Наибольшего разнообразия в своем развитии графовые модели достигли в 1990-х годах. Наряду с обычными графами, представляющими собой множества вершин (помеченных или непомеченных), соединенных ребрами (направленными или ненаправленными, помеченными или непомеченными), развитие в графовых моделях получили такие структуры, как *гипервершины* и *гиперграфы*. Гипервершина представляет собой вложенный граф, а ребра гиперграфа могут соединять не две, а произвольное количество вершин [7].

Однако, обзоры состояния современных графовых СУБД [3] показывают, что большинство существующих баз данных основаны на простых или атрибутированных графах (*attributed graph* или *property graph*), в которых атрибуты (свойства) приписываются ребрам и/или вершинам графа. Именно такие модели и были выбраны в данной статье в качестве исходных, подлежащих унификации моделей.

В качестве канонической модели в работе рассматривается объектно-фреймовая модель данных, а именно - язык СИНТЕЗ [10], нацеленный на разработку предметных посредников для решения задач в средах неоднородных ресурсов, и поддержанный программными средствами исполнительной среды предметных посредников.

Статья организована следующим образом.

В разделе 2 рассмотрена и проиллюстрирована на примере модель данных атрибутированных графов.

В разделе 3 рассмотрены и проиллюстрированы основные принципы отображения модели данных атрибутированных графов в язык СИНТЕЗ.

В разделе 4 рассмотрены вопросы доказательства сохранения информации и семантики операций при отображении графовых моделей в объектные с использованием формального языка спецификаций AMN [1].

В разделе 5 рассмотрены родственные исследования и направления дальнейшей работы.

2 Модель данных атрибутированных графов

В настоящее время существует большое количество СУБД, модели данных которых основаны на простых или атрибутированных графах. Языки определения данных (ЯОД), языки манипулирования данными (ЯМД), прикладные интерфейсы пользователя (API) различаются в этих системах весьма существенно. Для того, чтобы обеспечить общность подхода по унификации графовых моделей, в данной статье рассматривается синтетическая модель данных атрибутированных графов. Структуры данных модели покрывают возможности моделей таких известных систем, как, например, Neo4j [11], Dex [15], InfiniteGraph, OrientDB, VertexDB, Filament, OQGraph, Horton, InfoGrid.

В качестве ЯМД синтетической модели рассматривается декларативный язык Cypher [17], развиваемый в системе Neo4j. Поэтому, фактически, рассматриваемая модель является расширением графовой модели Neo4j. С точки зрения общности по отношению к другим графовым языкам запросов, язык Cypher покрывает такие классы возможностей, как смежность (adjacency) вершин и ребер, достижимость по путям фиксированной длины (fixed-length paths reachability), достижимость по простым регулярным путям (regular simple paths), поиск кратчайших путей, поиск подграфов по образцу (pattern matching) [3]. Это также означает,

что язык Cypher покрывает возможности языков запросов основных современных графовых баз данных (в том числе, перечисленных в предыдущем параграфе).

Итак, синтетическая модель выбрана таким образом, чтобы рассматриваемые методы отображения ее в каноническую (раздел 3) можно было применить для унификации различных реальных графовых моделей систем, упомянутых выше.

Заметим, что в данной работе не рассматривается важный класс СУБД, основанных на модели RDF [12], включающий такие системы, как AllegroGraph, G-Store, BrightstarDB. Часто RDF относят к графовым моделям. Однако, ввиду обширности области применения и развития приложений RDF, а также специфики ЯОД, ЯМД и семантики RDF по сравнению с основной массой графовых моделей, вопросы унификации RDF следует рассматривать отдельно.

Рассмотрим сначала вопросы определения данных в модели данных атрибутированных графов.

База данных в модели есть граф, вершины и ребра которого *типизированы*. Тип вершины или ребра представляет собой, фактически, совокупность атрибутов (свойств), приписываемых вершине или ребру. Определим формально множество всевозможных типов вершин *VertexTypes* и множество типов ребер *EdgeTypes*.

Так, *VertexTypes* представляет собой множество троек вида <*id, name, A*>, где *id* – идентификатор типа (например, целое число), *name* – имя типа (строка символов), *A* – подмножество множества всевозможных атрибутов *Attributes*.

Множество атрибутов *Attributes* есть множество кортежей вида <*id, name, type*>, где *id* и *name* – идентификатор и имя атрибута соответственно, *type* $\in$ *B* – тип атрибута, *B* – множество встроенных типов (например, *boolean, int, float, string,*типы массивов и т.д.)

Множество *EdgeTypes* есть набор кортежей вида <*id, name, A, directed, restricted, head, tail*>, где *id* – идентификатор типа; *name* – имя типа; *A* $\subseteq$ *Attributes*; *directed* $\in$ *{true, false}* – флаг направленности ребра; *restricted* $\in$ *{true, false}* – булевский флаг определенности типов вершин, которые связывает ребро; *head* $\in$ *VertexTypes* – тип исходящей вершины ребра; *tail* $\in$ *VertexTypes* – тип входящей вершины ребра.

Для любого типа *T* $\in$ *EdgeTypes* если *restricted(T) = true*, то значения *head(T), tail(T)* определены; если же *restricted(T) = false*, то значения *head(T), tail(T)* не определены.

Произвольная *схема S = <VT(S), ET(S)>* обобщенной графовой модели включает два множества: множество типов вершин *VT(S)* $\subseteq$ *VertexTypes* и множество типов ребер *ET(S)* $\subseteq$ *EdgeTypes.*

База данных (граф) *G*, удовлетворяющий схеме *S* имеет вид *G = <V, E>*, где

- *V = {v | ∃T.(T ∈ VT(S) & v: T)}* – множество вершин графа такое, что любая вершина имеет тип из *VT(S)*;
- *E = {<e, t, h> | ∃T.(T ∈ ET(S) & e: T & t ∈ V & t: tail(T) & h ∈ V & h: head(T))}* – множество ребер графа такое, что любое ребро имеет тип из *ET(S)* и соединяет вершины из *V*.

Типизация *x: T* означает, что для вершины (ребра) *x* могут быть определены атрибуты из *A(T)* (атрибуты типа *T*).

Рассмотрим пример схемы *Cinema* базы данных фильмов [5] в обобщенной модели. Для упрощения будем опускать в примерах идентификаторы типов и атрибутов, считая, что имена однозначно идентифицируют типы:

VT(Cinema) = { people, movie }
ET(Cinema) = { cast, directs }
A(movie) = {<id, long>, <title, string>, <year, integer>}
A(people) = {<id, long>, <name, string>}
cast = <{<character, string>}, false, false, undefined, undefined>
directs = <∅, true, true, people, movie>

Схема включает два типа вершин (*people*, *movie*) и два типа ребер (*cast*, *directs*). Тип *movie* включает три атрибута (*id*, *title*, *year*), *people* – два (*id*, *name*), *cast* – один (*character*), *directs* – ни одного. Ребра типа *cast* являются ненаправленными и типы вершин, которые они связывают, не определены; ребра типа *directs* – направлены от вершины типа *people* к вершине типа *movie*.

Пример простого графа *g*, удовлетворяющего схеме *Cinema* выглядит следующим образом:

g = <{m, p}, {e}>
m: movie = <id: 1, title: “Lost in Translation”, year: 2003>
p: people = <id: 1, name: “Scarlett Johansson”>
e: cast = <<character: “Charlotte”>, m, p>

Вопрос выбора ЯМД для обобщенной графовой модели достаточно сложен. Графовые языки запросов развивались в течении многих лет вместе с самими графовыми моделями [2]. Существуют работы по анализу и сравнению выразительной силы и вычислительной сложности графовых ЯМД [19].

Однако, в современных популярных графовых СУБД языки манипулирования представлены в большинстве случаев просто прикладным интерфейсом пользователя (API), предоставляющим доступ к структуре графа, методы обхода графа и инкапсулирующим основные алгоритмы на графах. Не существует стандарта графового языка запросов.

В данной работе в качестве ЯМД модели атрибутированных графов выбран язык Cypher [17]. С одной стороны, этот язык поддерживается и развивается в одной из самых популярных графовых СУБД с открытым кодом – Neo4j. С другой стороны, язык является декларативным, в отличие от существующих графовых API или скриптовых языков (как Gremlin). Язык основывается на различных подходах и сложившихся техниках выразительных запросов. Основные конструкции и ключевые слова имеют сходство с такими широко распространенными языками, как SQL и SPARQL.

3 Отображение модели атрибутированных графов в каноническую информационную модель

В качестве канонической модели в данной статье рассматривается объектная модель языка СИНТЕЗ [10]. Объектные модели хорошо зарекомендовали себя при унификации различных классов моделей – структурированных, онтологических, сервисных, процессных [8]. Поэтому есть основания выбирать канонические объектные модели при интеграции информационных ресурсов, представленных в моделях различных классов. При этом графовые модели выступают как один из классов моделей ресурсов, подлежащих интеграции.

3.1 Отображение языка определения данных

Схема в обобщенной графовой модели представляется в языке СИНТЕЗ в виде одноименного модуля (например, *Cinema*), включающего классы, содержащие вершины и ребра графа (например, *vertices* и *edges* соответственно):

```
{ Cinema; in: module;
  { vertices; in: class; … },
  { edges: in: class; … };
  …
}
```

Тип вершины (например, *movie* – см. раздел 2) представляется в языке СИНТЕЗ одноименным классом (который объявляется подклассом класса всех вершин *vertices*), также входящим в модуль, соответствующий схеме:

```
{ Movie; in: class; superclass: vertices;
  instance_type: {
    id: long;
    title: string;
    year: integer; };
}
```

Атрибуты типа вершины, представляются в языке СИНТЕЗ атрибутами типа экземпляров (*instance_type*) соответствующего класса. Между встроенными типами графовой модели (long, string, int и т.д.) и встроенными типами языка СИНТЕЗ (long, string, integer) устанавливается взаимно-однозначное соответствие.

Тип ребра (например, *directs* – см. раздел 2) также представляется одноименным классом (который объявляется подклассом класса всех вершин *edges*):

```
{ directs; in: class; superclass: edges;
  instance_type: {
    metaframe
      directed: true;
      restricted: true;
      startVertexType: people;
      endVertexType: movie;
    end
    edgeConstr: {in: invariant;
      {{ all e/directs.inst (directs(e) ->
           people(e.startVertex) & movie(e.endVertex)) }}
    }; };
}
```

Атрибуты типа ребра, аналогично типу вершины, представляются в языке СИНТЕЗ атрибутами типа экземпляров соответствующего класса.

Заметим, что информация о направленности (*directed*), определенности ребра (*restricted*), типах его исходящей (*startVertexType*) и входящей (*endVertexType*) вершин представляется специальной конструкцией - метафреймом [9], связанным с типом экземпляра класса. Метафреймы в языке СИНТЕЗ предназначены для выражения дополнительной метаинформации, связанной с такими сущностями, как модули, типы, классы, функции.

Кроме того, ограничение на типы исходящей и входящей вершин представляется инвариантом *edgeConstr*, заданным формулой в типизированной логике первого порядка. Знак *all* означает квантор всеобщности, знак -> - логическую импликацию, & - конъюнкцию, выражение *x/T* – типизацию переменной *x* типом *T*, *C.inst* – тип экземпляров (instance) класса *C*. Предикат *C(x)*, где *C* – имя класса, обращается в истину на экземплярах класса *C*.

Заметим также, что на переменной *e* типа *directs.inst* определены атрибуты исходящей вершины ребра *startVertex* и входящей вершины ребра *endVertex*, хотя их нет непосредственно в типе *directs.inst*. Эти атрибуты являются общими для всех типов вершин и принадлежат типу экземпляров класса *edges*:

```
{ edges; in: class;
  instance_section: {
    startVertex: vertices.inst;
    endVertex: vertices.inst;
    isValidEdge: { in: predicate;
      params: {+stVtx/vertices.inst,
               +endVtx/vertices.inst
               returns/Boolean };
      {{ (stVtx = this.startVertex &
          endVtx = this.endVertex -> returns = true) &
         (stVtx <> this.startVertex |
          endVtx <> this.endVertex -> returns = false) }}
    };
}
```

Кроме упомянутых атрибутов, тип *edges.inst* включает метод-предикат *isValidEdge*. Предикат

e.isValidEdge(v1, v2) обращается в истину, если исходящая вершина ребра *e* (*e.startVertex*) совпадает с *v1* и входящая вершина ребра *e* (*e.endVertex*) совпадает с *v2*. Спецификация метода задается формулой первого порядка, связывающей входные и выходные параметры метода. Знак | означает дизъюнкцию, <> - неравенство, *this* – объект, для которого вызывается метод.

3.2 Отображение языка манипулирования данными

При интеграции неоднородных ресурсов (баз данных, сервисов и т.д.) необходимо отображение ЯОД модели ресурса в каноническую. ЯМД канонической модели, напротив, необходимо отображать в ЯМД модели ресурса, т.к. запросы к посреднику в канонической модели нужно отображать в запросы к ресурсам.

Язык запросов (программ) модели СИНТЕЗ представляет собой Datalog-подобный язык в объектной среде. Программа представляет собой набор конъюнктивных запросов (правил) вида

$q(x/T) :- C_1(x_1/T_1), \dots, C_n(x_n/T_n),$
$F_1(X_1, Y_1), \dots F_m(X_m, Y_m), B.$

Тело запроса представляет собой конъюнкцию предикатов-коллекций, функциональных предикатов и ограничения. Здесь C_i - имена коллекций (классов), F_j – имена функций, x_i – имена переменных, значения которых пробегают по классам, T_i – типы переменных, X_j и Y_j – входные и выходные параметры функций, B – ограничение, налагаемое на x_i, X_j, Y_j.

В дальнейшем будет использоваться запись предиката-коллекции вида *movie([title, year])*. Неформально это означает, что нас не интересуют объекты класса *movie* целиком, а лишь их атрибуты *title*, *year*. Формально запись означает сокращение от *movie(_/movie.inst[title, year])*. Здесь знак _ обозначает анонимную переменную, *movie.inst* – анонимный тип экземпляров (instance) класса *movie*, *title, year* – необходимые атрибуты типа экземпляров класса.

Будет также использоваться запись *source([i, j, val1/val])*, означающая переименование атрибута *val* в *val1*.

Ввиду ограниченного объема статьи, отображение основных конструкций ЯМД будет продемонстрировано на нескольких примерах.

Пример 1 (Конъюнктивный запрос с использованием предиката смежности вершин и ребер). Рассмотрим запрос, возвращающий имена актеров по фамилии Круз, игравших в фильмах вместе со Скарлетт Йохансон:

```
q([colleague_name]) :-
people(scarlett/[name]),
movies(m),
people(colleague/[collegue_name: name]),
cast(c1), cast(c2),
c1.isValidEdge(m, scarlett),
c2.isValidEdge(m, colleague),
scarlett.name = “Scarlett Johansson”,
colleague.name.like(“*Cruz*”).
```

Запрос вернет непустой результат, если в графе базы данных существуют такие вершины-фильмы *m*, и такие ребра *c1, c2* типа *cast*, что *c1* соединяет *m* с вершиной *scarlett*, и *c2* соединяет *m* с вершиной *colleague*.

В языке Cypher такой запрос имеет вид

```
START scarlett =
  node:node_auto_index(name = 'Scarlett Johansson')
MATCH m-[c1:cast]-scarlett, m-[c2:cast]-collegue
WHERE collegue.name =~ /*Cruz*/
RETURN collegue.name
```

Каждый запрос языка Cypher представляет собой образец (pattern), по которому производится поиск в графе базы данных.

В секции START запроса указываются вершины или ребра, с которых следует начинать поиск. В данном случае это вершина *scarlett*, поскольку для нее указано значение атрибута *name*, а значит, возможен поиск по индексу этого атрибута.

В секции MATCH указывается образец поиска в графе, привязанный к стартовым вершинам. В данном случае это указание, что следует искать фильмы, в которых играла (*Cast*) *scarlett*, а также других актеров, играющих в том же фильме.

В секции WHERE указывается фильтр поиска. В данном случае это фамилия коллеги-актрисы.

В секции RESULT указываются возвращаемые значения. В данном случае это полное имя коллеги-актера.

Основные принципы отображения конъюнктивных запросов объектной модели в язык Cypher, проиллюстрированные на данном примере, состоят в следующем:

- конъюнктивный запрос представляется в языке Cypher запросом, возвращающим результат (секция RETURN);
- предикаты-коллекции и предикат смежности вершин и ребер представляются образцами секции MATCH. Каждому предикату смежности соответствует свой образец. Переменные, типизированные в предикатах-коллекциях, представляются одноименными переменными, использующимися в образцах;
- предикаты-условия представляются соответствующими предикатами секции WHERE или START;
- атрибуты результирующего предиката конъюнктивного запроса представляются одноименными атрибутами в секции RETURN.

Пример 2 (Удаление вершин). Рассмотрим запрос, удаляющий из базы данных фильм «Отчаянный»:

-movie(m) :- movie(m), m.year = “Desperado”.

В правилах со знаком «–» в голове осуществляется удаление объектов из коллекции.

В языке Cypher такой запрос представляется запросом с секцией DELETE:

```
START m = node:node_auto_index(title = 'Desperado')
DELETE m
```

Пример 3 (Обновление значения атрибута). Рассмотрим запрос, устанавливающий год создания фильма «Васаби»:

```
movie(m/[year]) :-
movie(m/[title, year1/year]), m.title = “Vasabi”,
year = 2001.
```

В языке Cypher такой запрос такой запрос представляется запросом с секцией SET:

```
START m = node:node_auto_index(title = 'Vasabi')
SET m.year = 2001
RETURN year
```

4 Сохранение информации и семантики операций ЯМД при отображении

В данном разделе рассматриваются вопросы доказательства сохранения информации и семантики операций при отображении графовых моделей в объектные с использованием формального языка спецификаций AMN [1, 4]. Применяется метод, предложенный и опробованный при унификации модели, основанной на многомерных массивах, в работе [22].

Язык AMN основан на теории множеств и типизированном языке первого порядка. Спецификации AMN называются *абстрактными машинами* и сочетают в себе пространства состояний и поведения машины, определенного операциями на состояниях. В языке AMN формализуется специальное отношение между спецификациями – *уточнение*.

Идея метода заключается в следующем. Рассмотрим исходную модель *S* и целевую модель *T*. Построим отображение θ модели *S* в модель *T*. Выразим семантику моделей в виде абстрактных машин AMN, построив при этом машины M_S и M_T соответственно. При этом структуры данных моделей представляются переменными машин, свойства структур данных представляются инвариантами машин, характерные операции моделей данных представляются операциями машин. *Операциями* в данном случае называются характерные родовые запросы в языках СИНТЕЗ и Cypher соответственно.

Рассматриваемые операции исходной и целевой модели должны быть связаны отображением ЯМД. Отображение ЯОД представляется в виде специального *склеивающего инварианта* – замкнутой формулы, связывающей состояния машин M_S и M_T.

Отображение θ считается *сохраняющим информацию и семантику операций*, если машина M_S, соответствующая исходной модели, уточняет машину M_T, соответствующую целевой модели [22]. Уточнение доказывается интерактивно при помощи специальных программных средств [4].

В качестве иллюстрации основных принципов выражения семантики синтетической графовой модели и языка СИНТЕЗ в AMN рассмотрим частичные (в связи с ограниченным объемом статьи) AMN-спецификации, соответствующие данным моделям. Нижеследующий текст организован следующим образом: приводятся последовательные части спецификации на языке AMN и сопровождаются комментариями.

Основные идеи представления семантики объектной модели языка СИНТЕЗ в языке AMN изложены в работе [22]. В настоящей статье рассматривается семантика специфических конструкций, необходимых для унификации графовых моделей.

Итак, *спецификация*, выражающая семантику объектной модели языка СИНТЕЗ, представляется в языке AMN конструкцией REFINEMENT:

```
REFINEMENT ObjectDM
```

Константы, необходимые для унификации графовой модели, объявлены в разделе CONSTANTS машины *ObjectDM* и типизируются в разделе PROPERTIES:

```
CONSTANTS
    c_edges, c_vertices,
    a_startVertex, a_endVertex,
    c_edges_instance_type
PROPERTIES …
```

Раздел PROPERTIES содержит формулу, которая состоит из предикатов, типизирующих константы Предикаты соединяются операцией конъюнкции. Так, имена классов ребер и вершин представлены константами *c_edges* и *c_vertices*, тип которых – подмножество множества строк (*STRING_Type*):

```
    c_edges: STRING_Type &
    c_vertices: STRING_Type
```

Знак типизации «:» формально означает принадлежность элемента множеству.

Имя типа экземпляров класса ребер представлено константой *c_edges_instance_type*:

```
    c_edges_instance_type: STRING_Type
```

Идентификаторы атрибутов этого типа, соответствующих исходящей и входящей вершинам ребра, представляются константами *a_startVertex, a_endVertex*, тип которых – натуральное число (*NAT*):

```
    a_startVertex: NAT &
    a_endVertex: NAT
```

Переменные, составляющие пространство состояний объектной модели, объявлены в разделе ABSTRACT_VARIABLES машины *ObjectDM* и типизируются в разделе INVARIANT:

```
ABSTRACT_VARIABLES
    m_directed, m_restricted,
    m_startVertexType,  m_endVertexType,
    isValidEdge
INVARIANT …
```

Раздел INVARIANT содержит формулу, которая состоит из предикатов, типизирующих переменные состояния, и налагающих различные совместные ограничения на переменные и константы. Предикаты соединяются операцией конъюнкции.

Так, декларируется, что *c_edges* и *c_vertices* действительно являются именами классов:

```
c_edges: classNames &
c_vertices: classNames
```

Здесь *classNames* – множество, содержащее имена всех классов базы данных [22].

Метаинформация, связанная с типом экземпляров класса ребер, представлена переменными *m_directed* (направленность ребра), *m_restricted* (определенность типов вершин ребра), *m_startVertexType* (тип исходящей вершины), *m_endVertexType* (тип входящей вершины):

```
m_directed: subclasses(c_edges) --> BOOL &
m_restricted: subclasses(c_edges) --> BOOL &
m_startVertexType:
    subclasses(c_edges) --> subclasses(c_vertices) &
m_endVertexType:
    subclasses(c_edges) --> subclasses(c_vertices)
```

Переменные типизированы полными функциями (знак -->), определенными на множестве всех классов ребер (которые являются подклассами класса всех вершин *c_edges*). Функция *subclasses* ставит в соответствие классу множество его подклассов.

Декларируется, что *c_edges_instance_type* действительно является именем типа, а атрибуты *a_startVertex* и *a_endVertex* являются атрибутами этого типа. Декларируется также, что тип значений данных атрибутов – абстрактный тип данных (*ADT*):

```
c_edges_instance_type: typeNames &
a_startVertex: typeAttributes(c_edges_instance_type) &
a_endVertex: typeAttributes(c_edges_instance_type) &
attributeType(a_startVertex) = ADT &
attributeType(a_endVertex) = ADT
```

Здесь функция *typeAttributes* возвращает множество атрибутов типа, функция *attributeType* – тип значений атрибута [22].

Предикат смежности вершин и ребер представляется функцией *isValidEdge*, сопоставляющей ребру *edg* и двум вершинам v_1, v_2 значение *истина*, если вершины v_1, v_2 соединены ребром *edg*:

```
isValidEdge: objectsOfClass(c_edges)*
    objectsOfClass(c_vertices) *
    objectsOfClass(c_vertices) --> BOOL
!(edg, v1, v2).(edg: objectsOfClass(c_edges) &
    v1: objectsOfClass(c_vertices) &
    v2: objectsOfClass(c_vertices) =>
    ((isValidEdge(edg, v1, v2) = TRUE) <=>
    (adtAttributeValue(a_startVertex)(edg) = v1 &
    adtAttributeValue(a_endVertex)(edg) = v2) ) )
```

Здесь * - знак декартова произведения множеств. Функция *adtAttributeValue(a)(o)* возвращает значение атрибута *a* объекта *o* [22].

Дополнительные необходимые свойства переменных состояния представлены конъюнктивными компонентами инварианта. Так, ребро обязательно связывает два объекта из класса *c_vertices* (вершины):

```
!edg.(edg: objectsOfClass(c_edges) =>
    adtAttributeValue(a_startVertex)(edg):
      objectsOfClass(c_vertices) &
    adtAttributeValue(a_endVertex)(edg):
      objectsOfClass(c_vertices) )
```

Здесь «!» – знак квантора всеобщности, «=>» – логическая импликация. Функция *objectsOfClass* возвращает множество объектов – экземпляров класса [22].

Если типы вершин, соединяемых ребром, определены, то они должны принадлежать классам, задаваемым метаатрибутами *startVertexType* и *endVertexType* класса ребра:

```
!(cls, edg).(cls: subclasses(c_edges) &
    edg: objectsOfClass(cls) =>
    (m_restricted(cls) = TRUE =>
    adtAttributeValue(a_startVertex)(edg):
      objectsOfClass(m_startVertexType(cls)) &
    adtAttributeValue(a_endVertex)(edg):
      objectsOfClass(m_endVertexType(cls)) ) ) &
```

Из всего ЯМД в спецификации рассмотрена единственная операция *deleteVertex* удаления вершины:

```
OPERATIONS
deleteVertex(attr, cond) =
PRE attr : dom(attributeNames) &
  cond : INT --> BOOL &
  attributeType(attr) = Integer
THEN
  objectsOfClass(c_vertices) :=
  objectsOfClass(c_vertices) -
  { vert | vert: objectsOfClass(c_vertices) &
    vert: dom(adtAttributeValue(attr)) &
    cond(integerAttributeValue(attr)(vert)) = TRUE }
END
```

Параметрами операции являются идентификатор целочисленного атрибута *attr* и функция *cond*, отвечающая условию на значение атрибута. Операция *deleteVertex* удаляет из класса *c_vertices* все такие вершины *vert*, что на *vert* определен атрибут *attr*, и для значения этого атрибута выполнено условие *cond*. Здесь знак «:=» означает присваивание, знак «-» - разность множеств; конструкция *{v | F(v)}* – выделение множества таких значений *v*, что предикат *F(v)* обращается в истину, функция *integerAttributeValue(a)(o)* возвращает значение целочисленного атрибута *a* объекта *o*.

Спецификация, выражающая семантику синтетической графовой модели, представляется в языке AMN конструкцией

```
REFINEMENT GraphDM
```

Переменные, составляющие пространство состояний объектной модели, объявлены в разделе ABSTRACT_VARIABLES машины *GraphDM*:

```
ABSTRACT_VARIABLES
  vertexTypeIDs, edgeTypeIDs, attributeIDs,
  typeName, attributes, attributeName, attributeTyping,
  directed, restricted, headType, tailType,
  vertices, vertixType, edges, edgeType,
  headVertix, tailVertix,
  g_integerAttributeValue
```

Идентификаторы типов вершин представлены переменной *vertexTypeIDs*; идентификаторы типов ребер - переменной *edgeTypeIDs*; идентификаторы атрибутов – переменной *attributeIDs*; имена типов - переменной *typeName*; принадлежность атрибутов типам – переменной *attributes*; имена атрибутов – переменной *attributeName*; типы значений атрибутов – переменной *attributeTyping*; направленность ребер - переменной *directed*; определенность типов исходящей и входящей вершин ребра - переменными *restricted*, *headType, tailType*; вершины и ребра, составляющие базу данных - переменными *vertices, edges*; типы конкретных вершин и ребер - переменными *vertixType, edgeType;*исходящая и входящая вершины конкретных ребер - переменными *headVertix, tailVertix;* значения целочисленных атрибутов - переменной *g_integerAttributeValue.* Функции, представляющие значения атрибутов других типов (например, *BOOL* или *STRING*), определяются аналогично.

Переменные типизируются в разделе INVARIANT при помощи частичных (знак «+->») и тотальных функций аналогично переменным, использующимся для придания семантики объектной модели:

```
INVARIANT
    vertexTypeIDs: POW(NAT) &
    edgeTypeIDs: POW(NAT) &
    attributeIDs: POW(NAT) &
    typeName: vertexTypeIDs \/ edgeTypeIDs -->
       STRING_Type &
    attributes: vertexTypeIDs \/ edgeTypeIDs -->
       POW(attributeIDs) &
    directed: edgeTypeIDs --> BOOL &
    restricted: edgeTypeIDs --> BOOL &
    headType: edgeTypeIDs +-> vertexTypeIDs &
    tailType: edgeTypeIDs +-> vertexTypeIDs &
    attributeName: attributeIDs --> STRING_Type &
    attributeTyping: attributeIDs --> BuiltInTypes &
    vertices: POW(NAT) &
    vertixType: vertices --> vertexTypeIDs &
    edges: POW(NAT) &
    edgeType: edges --> edgeTypeIDs &
    headVertix: edges --> vertices &
    tailVertix: edges --> vertices &
    g_integerAttributeValue:
       (vertices \/ edges)*attributeIDs +-> INT &
```

Здесь знак «\/» означает объединение множеств.

Дополнительные необходимые свойства переменных состояния представлены конъюнктивными компонентами инварианта. Так, для тех типов, метаатрибут *restricted* которых принимает значение *TRUE*, заданы типы исходящей и входящей вершин:

```
!(type).(type: edgeTypeIDs =>
      (restricted(type) = TRUE =>
         type: dom(headType) & type: dom(tailType)) &
      (restricted(type) = FALSE =>
         type /: dom(headType) & type /: dom(tailType)) )
```

Здесь функция *dom* возвращает область определения функции, знак «/:» означает непринадлежность элемента множеству.

Функция *g_integerAttributeValue* определена только для целочисленных атрибутов. Атрибут, для которого определена эта функция, принадлежит типу соответствующей вершины или ребра:

```
!(vert, attr).(vert: vertices & attr: attributeIDs =>
      ((vert |-> attr) : dom(g_integerAttributeValue) =>
         attributeTyping(attr) = Integer) &
         attr: attributes(vertixType(vert))  ) &
!(edg, attr).(edg: edges & attr: attributeIDs =>
      ((edg |-> attr) : dom(g_integerAttributeValue) =>
        attributeTyping(attr) = Integer) &
        attr: attributes(edgeType(edg))  )
```

Если типы вершин, соединяемых ребром, определены (значение метаатрибута *restricted* типа этого ребра принимает значение *TRUE*), то они должны принадлежать типам, задаваемым метаатрибутами *headType* и *tailType* типа ребра:

```
!edg.(edg: edges =>
      (restricted(edg) = TRUE =>
         vertixType(headVertix(edg)) =
            headType(edgeType(edg)) &
         vertixType(tailVertix(edg)) =
            tailType(edgeType(edg))  )  )
```

Аналогично объектной модели рассмотрена единственная операция ЯМД – операция удаления вершины *deleteVertex*:

```
OPERATIONS
deleteVertex(attr, cond) =
PRE attr: attributeIDs & cond: INT --> BOOL &
   attributeTyping(attr) = Integer
THEN
    vertices := vertices -
    {vert | vert: vertices &
       attr: attributes(vertixType(vert)) &
       cond(g_integerAttributeValue(vert, attr)) = TRUE }
END
```

Сигнатура операции совпадает с сигнатурой операции объектной модели. Семантика операции также аналогична: вершина *vert* удаляется из базы данных (множества *vertices*), если на *vert* определен атрибут *attr*, и для значения этого атрибута выполнено условие *cond.*

Для формального доказательства того, что машина *GraphDM* уточняет машину *ObjectDM* необходимо построить *инвариант уточнения*, связывающий переменные машин и добавить его к инварианту уточняющей машины.

Инвариант формализует принципы отображения ЯОД, изложенные в разделе 3.1 и объединяет их в одну конъюнкцию.

Множество имен типов графовой модели совпадает с множеством имен классов объектной модели (за исключением предопределенных классов *c_edges, c_vertices*):

```
ran(typeName) = classNames - {c_edges, c_vertices}
```

Множество атрибутов типов графовой модели соответствует множеству атрибутов объектной модели (за исключением предопределенных атрибутов *a_startVertex, a_endVertex*):

```
attributeIDs = dom(attributeNames) –
      {a_startVertex, a_endVertex}
```

Имена и типы атрибутов графовой и объектной модели совпадают:

```
!attr.(attr: attributeIDs =>
      attributeName(attr) = attributeNames(attr) &
      attributeTyping(attr) = attributeType(attr)  )
```

Вершины и ребра графовой базы данных соответствуют объектам классов *c_vertices* и c*_edges*:

```
vertices = objectsOfClass(c_vertices) &
edges = objectsOfClass(c_edges)
!vert.(vert: vertices =>
 ((vert: objectsOfClass(typeName(vertixType(vert)))) <=>
 (vert: vertices))  ) &
!edg.(edg: edges =>
  ((edg: objectsOfClass(typeName(edgeType(edg)))) <=>
  (edg: edges))  )
```

Значения атрибутов вершин и ребер графовой модели совпадают со значениями соответствующих атрибутов соответствующих объектов:

```
!(vert, attr).(vert: vertices & attr: attributeIDs =>
      ((vert |-> attr) : dom(g_integerAttributeValue) =>
         g_integerAttributeValue(vert, attr) =
            integerAttributeValue(attr)(vert))   )
!(edg, attr).(edg: edges & attr: attributeIDs =>
      ((edg |-> attr) : dom(g_integerAttributeValue) =>
         g_integerAttributeValue(edg, attr) =
            integerAttributeValue(attr)(edg))   )
```

Для указания того, что машина *GraphDM* уточняет машину *ObjectDM*, в машину *GraphDM* была добавлена директива

```
REFINES ObjectDM
```

Спецификации *ObjectDM* и *GraphDM* вместе с инвариантом уточнения были загружены в инструментальное средство Atelier B [4]. Автоматически были сгенерированы теоремы, выражающие уточнение спецификаций. В частности, для операции *deleteVertex* были сгенерированы 15 теорем, все они были доказаны также автоматически.

5 Родственные исследования и направления дальнейшей работы

Известно сравнительно небольшое количество работ, в которых исследуются вопросы интеграции или отображения графовых моделей данных. Например, в работе [18] язык запросов над гиперграфами используется для описания взглядов при интеграции графовых баз данных в посредниках. В работе [13] гиперграфовая модель также используется для интеграции графовых баз данных. Предлагается набор операций в рамках гиперграфовой модели для преобразования схемы ресурса в федеративную схему. В данных работах вопрос модельной неоднородности не встает, так как и в качестве канонической модели, и в качестве модели ресурсов выступает гиперграфовая модель. В работе [14] рассматривается отображение реляционной модели в гиперграфовую и императивная реализация операций реляционной алгебры в гиперграфовой модели. Таким образом, в качестве канонической модели также выступает гиперграфовая модель, а в качестве модели ресурса – реляционная.

В области интеграции графовых баз данных существует еще одна группа работ, в которых рассматриваются вопросы поглощения запросов, ответа на запросы и переписывания запросов с использованием взглядов (представлений). Текущие результаты в данной области изложены в работе [5]. Получены верхние границы сложности ответа на запросы, переписывания запросов с использованием GLAV-взглядов (Global and Local As View) в графовых моделях; доказана разрешимость поглощения запросов.

Основные особенности настоящей работы состоят в следующем. Целью работы является устранение модельной неоднородности современных графовых СУБД для дальнейшей их виртуальной или материализованной интеграции. В качестве исходной модели при отображении используется синтетическая модель, структуры данных которой покрывают возможности современных СУБД, основанных на простых и атрибутированных графах. В качестве целевой модели используется каноническая объектно-фреймовая модель - язык СИНТЕЗ. Для отображения обеспечивается формальное доказательство сохранения информации и семантики операций ЯМД.

Дальнейшая работа включает следующие этапы:

- выбор конкретных графовых моделей, основанных на простых и атрибутированных графах и построение трансформаций, реализующих изложенное отображение;
- расширение инструментальных средств поддержки предметных посредников для виртуальной интеграции графовых баз данных;
- применение технологии предметных посредников для решения научных задач в некоторой предметной области над множеством неоднородных ресурсов, включающим графовые базы данных.

Литература

[1] Abrial J.-R. The B-Book: Assigning Programs to Meanings. Cambridge: Cambridge University Press, 1996.

[2] R. Angles, C. Gutierrez. Survey of Graph Database Models. ACM Computing Surveys, Vol. 40, No. 1. Article No. 1, 2008.

[3] R. Angles. A Comparison of Current Graph Database Models. Proc. IEEE 28th International Conference on Data Engineering Workshops (ICDEW), 2012. – P. 171-177.

[4] Atelier B, the industrial tool to efficiently deploy the B Method. http://www.atelierb.eu/index-en.php

[5] Diego Calvanese, Giuseppe De Giacomo, Maurizio Lenzerini, Moshe Vardi. Query Processing under GLAV Mappings for Relational and Graph Databases. VLDB 2013: 61-72 (2013)

[6] Dex User Manual. 2013. http://www.sparsity-technologies.com/downloads/UserManual.pdf

[7] B. Iordanov. Hypergraphdb: a generalized graph database. Proc. 2010 International Conference on Web-age information management (WAIM). Springer-Verlag, 2010, pp. 25–36.

[8] Kalinichenko L.A., Briukhov D.O., Martynov D.O., Skvortsov N.A., Stupnikov S.A. Mediation Framework for Enterprise Information System Infrastructures. Proc. of the 9th International Conference on Enterprise Information Systems ICEIS 2007. - Funchal, 2007. - Volume Databases and Information Systems Integration. - P. 246-251.

[9] Kalinichenko L.A., Stupnikov S.A. Heterogeneous information model unification as a pre-requisite to resource schema mapping // A. D'Atri and D. Saccà (eds.), Information Systems: People, Organizations, Institutions, and Technologies (Proc. of the V Conference of the Italian Chapter of Association for Information Systems itAIS). – Berlin-Heidelberg: Springer Physica Verlag, 2010. – P. 373-380.

[10] Kalinichenko L.A., Stupnikov S.A., Martynov D.O. SYNTHESIS: a Language for Canonical Information Modeling and Mediator Definition for Problem Solving in Heterogeneous Information Resource Environments. Moscow: IPI RAN, 2007. - 171 p.

[11] Neo4j Graph Database. - http://www.neo4j.org/

[12] RDF Primer. W3C Recommendation 10 February 2004. Eds. F. Manola, E. Miller. 2004. - http://www.w3.org/TR/2004/REC-rdf-primer-20040210/

[13] Srikrishnan Sundaresan, Gongzhu Hu: Schema integration of distributed databases using hyper-graph data model. IRI 2005:548-553

[14] Amani Tahat, Maurice H. T. Ling: Mapping Relational Operations onto Hypergraph Model CoRR abs/1105.6118 (2011)

[15] The Dex Graph Database Management System. http://www.sparsity-technologies.com/dex.php

[16] The Fourth Paradigm: Data-Intensive Scientific Discovery. Eds. Tony Hey, Stewart Tansley, and Kristin Tolle. Redmond: Microsoft Research, 2009.

[17] The Neo4j Manual. 2013. - http://docs.neo4j.org/

[18] Dimitri Theodoratos: Semantic Integration and Querying of Heterogeneous Data Sources Using a Hypergraph Data Model. BNCOD 2002:166-182

[19] P. T. Wood. Query languages for graph databases. ACM SIGMOD Record. 2012. V. 41, I. 1. P. 50-60.

[20] Захаров В. Н., Калиниченко Л. А., Соколов И. А., Ступников С. А. Конструирование канонических информационных моделей для интегрированных информационных систем // Информатика и ее применения. – М., 2007. – Т. 1, Вып. 2. – С. 15-38.

[21] Скворцов Н. А. Отображение моделей данных NoSQL в объектные спецификации. Труды RCDL'2012. – Переславль-Залесский: Университет города Переславля, 2012. С. 78-87.

[22] Ступников С. А. Унификация модели данных, основанной на многомерных массивах, при интеграции неоднородных информационных ресурсов. Труды RCDL'2012. – Переславль-Залесский: Университет города Переславля, 2012. С. 67-77.

Mapping of a Graph Data Model into a Canonical Information Model for the Heterogeneous Information Resource Integration

Sergey Stupnikov

In the paper a mapping of an attributed graph data model into an object-frame canonical information model used for virtual or materialized database integration is presented.

An aim of the work is developing of a sound theoretical basis for the integration of graph-based resources. A verification of the mapping using a formal specification language and a specific theorem prover is provided.

Отображение модели данных RDF в каноническую модель предметных посредников

© Н. А. Скворцов
Институт проблем информатики РАН
Москва
nskv@ipi.ac.ru

Аннотация

Модель данных RDF предназначена для описания ресурсов произвольного вида в открытой информационной среде и их идентификации в ней. При решении задач над множественными неоднородными информационными ресурсами возникает необходимость использовать данные, представленные в данной модели. В статье рассмотрены подходы к отображению модели данных RDF и сопутствующих ей языков RDF-Schema и SPARQL в объектную модель языка СИНТЕЗ, используемого в качестве унифицирующей информационной модели при интеграции множественных неоднородных информационных ресурсов в информационные системы, создаваемые на основе технологии предметных посредников. Предложен подход, учитывающий выявление семантики данных RDF и схемы данных в случае её присутствия.

Работа выполнена при поддержке РФФИ (гранты 11-07-00402-а, 13-07-00579-а) и Президиума РАН (программа 16П, проект 4.2).

1 Введение

Исследования, посвящённые решению научных задач над множественными неоднородными информационными ресурсами, включают разработку предметных посредников [14]. Предметный посредник определяется: канонической унифицирующей моделью данных, необходимой для однородного представления информации и манипулирования ею; спецификациями онтологии, описывающей предметную область посредством понятий, и концептуальной схемы, определяющей структуру и поведение информационных объектов предметной области в терминах канонической модели; набором информационных ресурсов, интегрированных в предметную область.

Для интеграции информационных ресурсов в спецификацию предметной области посредника модели данных, в которых представлены интерфейсы информационных ресурсов, отображаются в унифицирующую каноническую модель посредника, а после отображения моделей данных схемы данных ресурсов в канонической модели отображаются в спецификации предметной области. Отображение схем представляется в виде взглядов над концептуальной схемой предметной области посредника.

Каноническая модель посредника представляет собой ядро, в качестве которого используется расширяемая модель данных, определяемая языком СИНТЕЗ [8], и набор расширений в терминах ядра, соответствующих моделям интегрируемых в посредник информационных ресурсов. Разработка канонической модели посредника состоит в унификации моделей, при которой модели информационных ресурсов отображаются в ядро канонической модели, и формируют новые расширения, либо отображаются в существующие [7].

Неоднородность информационных ресурсов, которые могут быть интегрированы в предметные посредники, понимается широко. Она включает как неоднородность на уровне схем, так и практически произвольные модели данных. Помимо методов отображения в каноническую модель традиционных моделей данных, таких как объектная и реляционная модели, в последнее время были рассмотрены принципы отображения в каноническую модель различных перспективных моделей данных, включая сервисы, онтологические модели [18], модели многомерных массивов данных [16], модели баз данных NoSQL [15]. Рассматриваются также графовые модели [17].

Ещё одной широко используемой моделью данных является модель RDF [rdf]. Она используется для описания ресурсов произвольного вида в открытой информационной среде, а также в качестве базовой модели в проектах

Труды 15-й Всероссийской научной конференции «Электронные библиотеки: перспективные методы и технологии, электронные коллекции» — RCDL-2013, Ярославль, Россия, 14-17 октября 2013 г.

Семантического Веба. В основе модели лежат триплеты «субъект-предикат-объект», посредством которых описываются ресурсы.

RDF является разновидностью графовой модели данных, в которой субъекты и объекты являются узлами, а предикаты – направленными именованными рёбрами. Однако относительно графов модель RDF привносит свою специфику. Во-первых, посредством использования URI она определяет глобальную идентификацию ресурсов в открытом информационном пространстве. Во-вторых, модель является расширяемой. За рёбрами RDF-графов может скрываться определённая семантика в зависимости от используемых словарей. В частности, RDF-модель используется в качестве базиса расширяющих её моделей, в которых предопределены предикаты, несущие конкретную семантику элементов моделей. В-третьих, в модели RDF предусмотрены правила вывода неявных связей, которые также определяют их семантику элементов модели. В зависимости от словарей, определяющих семантику описаний, набор правил вывода также может расширяться. Все эти особенности отражаются на подходах к отображению данной модели в каноническую модель предметных посредников. Таким образом, модель данных RDF заслуживает отдельного рассмотрения.

С моделью RDF неразрывно связаны модель RDF-Schema [2], предоставляющая средства определения схем данных для RDF-описаний, а также язык запросов SPARQL [3], основанный на использовании триплетов в запросах. Фактически они вместе определяют обогащённую модель данных, включающую языки определения данных и манипулирования данными, для описаний в модели RDF. Помимо этого, модель RDF является базисом для других моделей данных. В частности, язык OWL [1] со определёнными семантикой и набором правил вывода, в одном из своих синтаксисов основывается на RDF, и данные, представленные в модели OWL, обретают вид RDF-ресурсов.

Данные RDF могут быть распределены в Вебе и представляться в открытой среде в виде документов, сгенерированных в соответствии с одним из синтаксисов, представляемых для модели RDF. Для поиска данных служат специализированные поисковые механизмы, обращение к которым возможно с помощью языка SPARQL [3]. С другой стороны, для хранения и манипулирования RDF-описаниями разработан ряд специализированных хранилищ. Большинство из них использовали реляционные отношения для представления графов и триплетов в них. Сегодня RDF-хранилища используют колоночную организацию хранения триплетов и горизонтальное масштабирование [6]. Помимо этого, множество проектов в Вебе предоставляют API-интерфейс к своим данным в виде RDF в соответствии с определённой схемой и точек доступа SPARQL.

Таким образом, информационными ресурсами с данными RDF, интегрируемыми в предметные посредники, могут становиться поисковые механизмы с точками доступа SPARQL общего назначения, RDF-хранилища или Веб-проекты с API, представляющим данные в виде RDF.

Правила вывода [2] над RDF-описаниями могут прорабатывать в разное время в зависимости от системы и соображений эффективности. Обычно они применяются при обновлении данных для обеспечения эффективности запросов, однако они могут вызываться и при выполнении запросов, если обеспечение скорости обновления данных важнее скорости выполнения запросов. Некоторые RDF-хранилища могут не поддерживать вывод, отключать его, обеспечивать вывод только подмножества правил. Состав правил может быть и расширен, например, за счёт использования производных моделей, таких как OWL [1]. При интеграции информационных ресурсов, предоставляющих данные в модели RDF и при отображении модели эти особенности должны быть учтены.

Дальнейшее изложение посвящено описанию отображаемой и целевой моделей (раздел 2), опыту отображения RDF в других исследованиях (раздел 3), выбору подхода для отображения модели RDF в спецификации в унифицирующей модели, используемой в среде предметных посредников, и отображению языков RDF, RDF Schema и SPARQL в язык СИНТЕЗ (раздел 4).

2 Связанные работы

Отображение и трансляция данных между моделью RDF и другими моделями данных не редкость. Большинство данных хранится и обрабатывается в моделях, отличных от RDF, а трансляция их в RDF необходима для обмена информацией между проектами, между различными представлениями данных, для публикации данных в семантическом вебе в таких проектах как Linked Open Data [12] и так далее. В основном, разрабатываемые отображения имеют следующие направления.

Большинство инструментов для работы с RDF используют исключительно триплеты в интерфейсе для всех предикатов, включая определённые в самих моделях RDF и RDF Schema. При этом сами инструменты могут быть предназначены для использования в других моделях данных, например, с объектно-ориентированными языками программирования. Отображение RDF в другие модели в них производится без ухода от триплетной формы [4], а основные преобразования из моделей, используемых в инструменте, в RDF приходится совершать на уровне прикладных программ.

В реляционных хранилищах триплетов интересно рассмотреть подходы к их хранению. Первый из них заключается в хранении триплетов в единственном отношении с тремя основными

атрибутами, соответствующими субъекту, предикату и объекту. Помимо этого в нём же могут храниться идентификаторы графов и другие необходимые элементы. Чтобы не хранить все данные в одном большом отношении, его разбивают по некоторому принципу, например, по принадлежности субъектов определённым классам [11]. Но для этого над RDF-данным должна быть определена некоторая схема. Другой способ разбиения отношения – по предикату [virt], при котором в отношении хранятся субъекты и объекты, связанные посредством одного и того же предиката. Предлагаемые представления продолжают поступать [13], мотивируемые поиском эффективного хранения и доступа к RDF-графам.

Отображения, учитывающие семантику данных, обычно разрабатывается для обратной задачи: трансляции данных из реляционной или объектной модели в RDF [10]. В этом случае обратная трансляция должна восстановить исходные спецификации.

Работы, связанные с отображением языка RDF с учётом семантики данных в высокоуровневые модели данных, разрабатываются для работы с RDF-данными из языков программирования [9] или языков декларативных запросов [5].

3 Модели данных RDF и СИНТЕЗ

Модель данных RDF включает в себя спецификации триплетов «субъект-предикат-объект», из которых строится направленный помеченный граф с субъектами и объектами в узлах и предикатами в дугах. Семантика предикатов и ресурсов определяется пространствами имён, соответствующими словарям. С ресурсами, фигурирующими в субъектах и объектах, связаны глобальные идентификаторы URI [uri]. Допускаются безымянные узлы, не имеющие идентификации. Также предусмотрено использование в качестве объектов в триплетах примитивных типов данных, определённых в XML-Schema [xmls], а также контейнеры (bag, seq, alt) и коллекции (list). Модель RDF определяет несколько синтаксисов сериализации описаний. Определяется графический синтаксис, два подхода к представлению триплетов, XML-синтаксис.

RDF Schema [2] привносит спецификации множеств ресурсов для определения схем данных RDF. Определены классы ресурсов, принадлежность к которым определяется предикатом type, и свойства с указанием классов области определения и области значений. Введены отношения обобщения/специализации для классов и для свойств. Определён список простых правил вывода [2], таких как транзитивность и рефлексивность отношений обобщения/специализации, отнесение ресурсов к классу в соответствии с областью определения или значения свойства, правила овеществления и другие. Они должны выполняться над спецификациями для выявления новых связей.

Язык запросов SPARQL [3] включает в себя средства задания запросов в виде конъюнкции триплетов, в которых некоторые элементы могут быть переменными. Помимо триплетной записи предоставляются возможности задания объединений, фильтров с помощью выражений, отметки необязательных элементов, сортировки, исключения дубликатов, предикативных запросов и другие. В заголовке запроса указываются используемые пространства имён. Результаты возвращаются в виде списка кортежей в соответствии с указанной структурой в виде XML, JSON или в виде триплетов.

Целевой моделью для отображения этих языков, связанных с моделью RDF, является язык СИНТЕЗ [8], выполняющий роль ядра для формирования канонической модели посредников. Это язык спецификации информационных ресурсов, который включает в качестве синтаксической основы язык фреймов, построенную над ним объектную модель и средства выражения логических формул.

Фреймы имеют идентификатор и набор слотов, каждый из которых может иметь набор значений слота. Язык фреймов позволяет специфицировать метафреймы, метаслоты и метазначения, которые сами определяются как фреймы. Средствами языка фреймов выражаются произвольные виды информации, в том числе, слабоструктурированные.

Объектная модель рассматривает фреймы как типизированные значения. Фрейм может отражать состояние объекта определённого типа, в этом случае, его структура должна соответствовать спецификации некоторого абстрактного типа данных. Помимо абстрактных типов данных, определяющих структуру и поведение объектов, вводятся также классы как множества однотипных объектов. Фрейм может становиться экземпляром класса, если соответствует типу экземпляров этого класса. Особым видом класса является метакласс ассоциаций, представляющий собой множества ассоциаций, между объектами в соответствии со спецификациями атрибутов типов.

Язык формул в язык СИНТЕЗ используется для предикативных спецификаций, задающих ограничения целостности в типах, для определения правил в логических программах и для задания запросов над спецификациями задач или информационных ресурсов.

Приведёнными средствами языка СИНТЕЗ мы выражаем спецификации, отображённые из модели RDF.

4 Подход к отображению модели RDF с сохранением семантики данных

Тривиальным отображением модели данных RDF в каноническую модель могло бы стать отображение отношений, хранящих триплеты, в абстрактный тип с атрибутами, соответствующими субъектам, предикатам и объектам. Однако такой подход не приемлем в среде предметных

посредников. Для них важна семантическая интеграция информационных ресурсов, при которой все обнаруживаемые связи данных с семантикой предметной области должны отображаться в понятия предметной области посредника и в соответствующие структуры концептуальной схемы посредника. Поэтому выявленную семантику предметной области следует применить для создания соответствующих типов в спецификациях информационных ресурсов, интегрируемых в посредник.

4.1 Отображение модели RDF во фреймовую модель

Отображать данные в базовой модели RDF в каноническую модель необходимо на уровне фреймов языка СИНТЕЗ. Фреймы, как и графы RDF, призваны описывать ресурсы произвольной природы. И при отображении RDF триплеты с общим субъектом как описывающие один и тот же ресурс должны ассоциироваться с определённым фреймом. Далее в примерах RDF-спецификаций используем синтаксис Turtle:

```
prefix vCard:
      <http://www.w3.org/2001/vcard-rdf/3.0#> .
prefix rdf:
      <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
prefix : <#> .
<http://somewhere/MattJones/>
          vCard:FN "Matt Jones" ;
          vCard:N [ vCard:Family "Jones";
                    vCard:Given "Matthew" ] .
```

Пространства имён, определяемые в RDF как словари, отображаются в миры фреймов. Мир фреймов в языке СИНТЕЗ представляется с помощью фрейма с указанием принадлежности его метаклассу world, определяющему семантику этого фрейма как мира. Внутри мира располагаются спецификации входящих в него фреймов.

```
{ vCard;
  in: world;
  …
}
```

Значение слота может включать вложенный фрейм в соответствии с возможностью задавать в RDF пустые узлы.

С определённым субъектом триплетов связывается фрейм с соответствующим именем. Набор триплетов с общим субъектом отображается в набор слотов одного фрейма. Предикатам, связанным с этим субъектом, соответствуют слоты фрейма. Объектам – значения слотов. Если тип значения не определён, он назначается как тип Frame, так как фреймы формируют произвольные значения. Набор триплетов, различающихся объектами, отображается в слот фрейма с несколькими значениями.

```
{ MattJones;
  in: frame;
  FN: Matt Jones;
  N: { in: frame; Family: Jones;},
     { in: frame; Given: Matthew;};
}
```

В графе имеются неименованные узлы, являющиеся объектами предиката N, которые выражаются встроенными фреймами без идентификатора.

Таким образом, данные в базовой модели RDF могут представляться как база фреймов. В RDF словари расширяют модель и могут привносить определённую семантику элементам RDF-модели. Поэтому определение схем над RDF-графами, исполняемое в модели RDF Schema, несёт в себе семантику определения структуры и влияет на отображение RDF Schema в язык СИНТЕЗ.

4.2 Отображение модели RDF-Schema в объектную модель

Описания в модели RDF Schema отображаются в объектную модель языка СИНТЕЗ.

Сложности отображения RDF в объектную модель (как и реляционную) связаны, в основном, с конфликтом парадигм открытого мира в случае RDF и закрытого мира в случае объектной модели [9]. Во-первых, описываемые ресурсы могут принадлежать нескольким RDF-классам одновременно. Большинство объектных моделей исключает такую возможность, определяя принадлежность объекта строго одному типу. Во-вторых, свойства в RDF являются самостоятельными сущностями, которые могут использоваться в разных классах и ресурсах. Их связь с классами определяется только указанием классов домена и области значений свойств. Тем временем, атрибуты типов в объектной модели обычно жёстко связаны с типами, а также наследуются от супертипов. В-третьих, структура экземпляров в RDF не ограничивается спецификациями классов. В большинстве объектных моделей объект жёстко соответствует структуре, описанной в типе, которому он принадлежит. Средства языка СИНТЕЗ позволяют избежать указанных проблем, благодаря разделению в языке на классы как множества и абстрактные типы данных как интенсиональные описания, а также возможности использовать типовые выражения для их комбинации.

Определения RDF Schema определяют схему RDF-данных посредством задания ресурсов, классов, свойств и их ограничений.

```
prefix rdf:
      <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
prefix rdfs:
      <http://www.w3.org/2000/01/rdf-schema#>
prefix vcard:
      <http://www.w3.org/2001/vcard-rdf/3.0#>
vCard:Individual rdf:type rdf:Class .
vCard:FN rdf:type rdf:Property;
          rdfs:domain vCard:Individual .
vCard:NPROPERTIES rdf:type rdfs:Class .
vCard:N rdf:type rdf:Property;
          rdfs:domain vCard:Individual;
          rdfs:range vCard:NPROPERTIES .
vCard:Family rdf:type rdf:Class;
          rdfs:subClassOf vCard: NPROPERTIES .
vCard:Given rdf:type rdf:Class;
           rdfs:subClassOf vCard: NPROPERTIES .
```

Ресурс RDF Schema отображается во фрейм языка СИНТЕЗ. Класс RDF Schema отображается в класс языка СИНТЕЗ с типом экземпляров.

```
{ individual;
  in: class;
  instance_section:
  {
    FN: {set_of: Frame};
    N: {set_of: nproperties};
  };
},
{ nproperties;
  in: class;
  instance_section: Frame
},
{ family;
  in: class;
  superclass: nproperties
  instance_section: Frame
},
{ given;
  in: class;
  superclass: nproperties
  instance_section: Frame
};
```

Свойства RDF Schema отображаются в общем случае в метакласс ассоциаций, тем самым создавая соответствующую самостоятельную сущность.

```
 { n;
   in: association, metaclass;
   instance_section:
   { association_type: {0, inf},{0, inf};
    domain: individual;
    range: nproperties;
   }
 }
```

В этом случае атрибут типа определяется с указанием метакласса ассоциаций:

```
N: {set_of: NPROPERTIES};
   metaslot
     in: n;
   end
```

Свойства domain и range соответствуют определению домена и диапазона значений в метаклассе ассоциаций. Если домен определяется как единственный класс, вместо метакласса ассоциаций возможно определять обычный атрибут в типе.

Свойства subclass и subproperty соответствуют отношению подкласса между классами канонической модели и метаклассами ассоциаций соответственно.

Конструкции Bag, Seq, List отображаются в коллекции канонической модели.

В случае присутствия в ресурсе свойства type по отношению к классу RDF Schema ресурс отображается в экземпляр класса, и фрейм, представляющий его, рассматривается как объект типа экземпляров данного класса.

```
{ MattJones;
  in: individual;
  FN: Matt Jones;
  N: { Jones; in: family;}, { Matthew; in: given;};
}
```

В языке RDF Schema определяется набор правил вывода, который должен использоваться над спецификациями. Некоторые из этих правил представимы в виде правил и инвариантов на языке формул канонической модели, а некоторые – предполагаются языком СИНТЕЗ и не требуют дополнительной спецификации. Однако разные реализации RDF ограничиваются использованием подмножества правил, указанных в стандарте RDF. В зависимости от набора реализованных правил вывода в информационном ресурсе, они должны

отображаться или не отображаться в спецификацию на языке СИНТЕЗ. Также необходимость их спецификации зависит от уровня их применения: они могут применяться при обновлении данных в информационном ресурсе, чего посредник СИНТЕЗ не предполагает делать; либо их применение зашито в обработчике запросов ресурса, тогда в спецификации достаточно инвариантов, проверяющих целостность данных.

4.3 Отображение запросов в язык SPARQL

В модели RDF нет собственного языка манипулирования данными. Для запросов над базами RDF-документов обычно используется язык SPARQL. Запросы на языке SPARQL представляют собой конъюнкции триплетов, на месте субъектов, предикатов и объектов в которых могут быть определённые значения или переменные. Пример запроса:

```
prefix vcard: <http://www.w3.org/2001/vcard-rdf/3.0#>
SELECT ?x ?fname ?gname
{
  ?x  vcard:N  ?vc .
  ?vc vcard:Family ?fname .
  ?vc vcard:Given ?gname .
}
```

Для языка манипулирования данными строится обратное отображение: из языка СИНТЕЗ в запрос на языке SPARQL – для возможности переписывания запросов к информационным ресурсам. В соответствии с возможностями языка запросов SPARQL, в них возможно отображение формул языка СИНТЕЗ в конъюнктивной форме над предикатами классов и условиями. Для формирования приведённого выше запроса на языке SPARQL в посреднике должен быть задан запрос на языке формул:

```
q([x, fname, gname]) :-
        individual(x),
        ex y (family(y) & is_in(y, x.N) & fname=y),
        ex z (given(z) & is_in(z, x.N) & gname=z)
```

В случае отображения RDF во фреймы канонической модели в качестве запросов применяются формулы канонической модели над базой фреймов [8]. В них фигурируют операции над фреймами и не используется типизация.

Таким образом, возможно отображение данных в модели RDF в базу фреймов языка СИНТЕЗ. При наличии спецификации схемы данных RDF она отображается в объектную спецификацию. Формулы ограниченного вида в запросах и программах посредника преобразуются в запросы на SPARQL.

4.4 Учёт производных моделей данных и логического вывода

Востребованность модели данных RDF возрастала последовательно в течение ряда лет. Устойчивым интересом к себе она обязана тем, что стала базовой моделью для семантического веба. Тем временем, хранилища RDF долгое время были слабо масштабируемы до развития принципов вертикального хранения, что на интенсивности использования модели сказывалось негативно. Однако производная модель данных OWL, благодаря своей выразительности и возможности автоматического вывода в задаче включения классов, стала использоваться для описания схем баз данных и знаний.

Экземпляры классов в модели OWL представимы в модели RDF. Поэтому в случае, когда RDF-данные ограничены схемой, описанной на языка OWL, это существенно влияет на семантическое отображение таких RDF-данных в другие модели, в частности, в язык СИНТЕЗ.

Отображение моделей данных OWL и OWL 2 в модель, определяемую языком СИНТЕЗ, было подробно рассмотрено ранее [18].

Логический вывод, предполагаемый в модели OWL, учитывается при обновлении данных. В этом случае, правила вывода далее не должны учитываться в отображении моделей.

5 Заключение

Исследованы модели данных RDF и RDF Schema, определёны принципы отображения RDF-спецификаций во фреймовую модель унифицирующей модели предметных посредников, определены принципы отображения спецификаций RDF Schema в объектную каноническую модель, описаны возможности запросов в канонической модели, отображаемых в язык запросов SPARQL, используемый для RDF-спецификаций и производных от них моделей. Результаты работы могут быть использованы для семантического отображения RDF-описаний в предметную область посредников и доступа к RDF-данным в них.

Литература

[1] OWL Web Ontology Language Reference. M. Dean, G. Schreiber (Eds.), W3C Recommendation. – W3C, 2004. – URL: http://www.w3.org/TR/2004/REC-owl-ref-20040210/.

[2] RDF vocabulary description language 1.0: RDF schema. D. Brickley, R.V. Guha (Eds.), W3C Recommendation. – W3C, 2004. – URL: http://www.w3.org/TR/2004/REC-rdf-schema-20040210.

[3] SPARQL Query Language for RDF. E. Prud'hommeaux, A. Seaborne (eds.), W3C Recommendation. – W3C, 2008. – URL: http://www.w3.org/TR/rdf-sparql-query/

[4] D. Beckett, J. Grant. SWAD-Europe Deliverable 10.2: Mapping Semantic Web data with RDBMSes. – 2003. – URL: http://www.w3.org/2001/sw/Europe/reports/scalable_rdbms_mapping_report.

[5] Chebotko A., Lu S., Fotouhi F. Semantics preserving SPARQL-to-SQL translation //Data & Knowledge Engineering. – 2009. – Т. 68. – №. 10. – С. 973-1000.

[6] Erling O. Virtuoso, a Hybrid RDBMS/Graph Column Store //Data Engineering. – 2012. – С. 3.

[7] Kalinichenko L.A., Stupnikov S.A. Heterogeneous information model unification as a pre-requisite to resource schema mapping. // A. D'Atri and D. Sacca (eds.), Information Systems: People, Organizations, Institutions, and Technologies (Proc. of the V Conference of the Italian Chapter of Association for Information Systems itAIS). – Berlin-Heidelberg: Springer Physica Verlag, 2009. – P. 373-380.

[8] Kalinichenko L.A., Stupnikov S.A., Martynov D.O. SYNTHESIS: a Language for Canonical Information Modeling and Mediator Definition for Problem Solving in Heterogeneous Information Resource Environments. – Moscow: IPI RAN, 2007. – 171 p.

[9] Oren E. et al. ActiveRDF: Object-oriented semantic web programming. // Proceedings of the 16th international conference on World Wide Web. – ACM, 2007. – С. 817-824.

[10] Sahoo S. S. et al. A survey of current approaches for mapping of relational databases to rdf // W3C RDB2RDF Incubator Group Report. – 2009.

[11] K. Wilkinson, C. Sayers, H. A. Kuno, D. Reynolds. Efficient RDF Storage and Retrieval in Jena2.// In Semantic Web and Databases Workshop. P. 131–150, 2003.

[12] Yu L. Linked open data //A Developer's Guide to the Semantic Web. – Springer Berlin Heidelberg, 2011. – С. 409-466.

[13] M. A. Bornea, et al. Building an Efficient RDF Store Over a Relational Database. // Proc. of the 2013 ACM SIGMOD International Conference on Management of Data. – New York, 2013. – P. 121-132.

[14] Брюхов Д.О., Вовченко А. Е., Захаров В.Н., Желенкова О.П., Калиниченко Л.А., Мартынов Д.О., Скворцов Н.А., Ступников С.А. Архитектура промежуточного слоя предметных посредников для решения задач над множеством интегрируемых неоднородных распределенных информационных ресурсов в гибридной грид-инфраструктуре виртуальных обсерваторий // Информатика и ее применения. – М., 2008. – Т. 2, Вып. 1. – С. 2-34.

[15] Н. А. Скворцов. Отображение моделей данных NoSQL в объектные спецификации // RCDL'2012. – CEUR Workshop Proceedings, 2012. – Т. 934. – С. 53-62.

[16] С. А. Ступников. Унификация модели данных, основанной на многомерных массивах, при интеграции неоднородных информационных ресурсов // RCDL'2012. – CEUR Workshop Proceedings, 2012. – Т. 934. – С. 42-52.

[17] С. А. Ступников. Отображение графовых моделей данных в каноническую информационную модель при интеграции неоднородных информационных ресурсов // RCDL'2013. – Готовится к печати.

[18] С. А. Ступников, Н. А. Скворцов. Взаимное отображение канонической информационной модели и языка OWL 2 // RCDL'2010. – Казань: КФУ, 2010. – Р. 392-398.

Mapping of RDF Data Model into the Canonical Model of Subject Mediators

Nikolay A. Skvortsov

The RDF data model becomes more and more widely used to identify, describe and interrelate resources of any kind in the open information environment. During the problem solving, there exists a need to involve data represented in this model in the process of multiple heterogeneous information resource integration. The paper analyzes approaches for mapping of the RDF data model expressed by its accompanying languages (RDF Schema and SPARQL) into the SYNTHESIS language. The SYNTHESIS language is used as a unifying canonical model for integration of multiple heterogeneous information resources in information systems created using subject mediation technique. An approach for RDF data model unification is proposed that takes into account discovery of RDF data semantics and schema.

СВОБОДНО РАСПРОСТРАНЯЕМЫЕ СИСТЕМЫ УПРАВЛЕНИЯ ЭЛЕКТРОННЫМИ НАУЧНЫМИ ЖУРНАЛАМИ И ТЕХНОЛОГИИ ЭЛЕКТРОННЫХ БИБЛИОТЕК♣

© А.М. Елизаров, Д.С. Зуев, Е.К. Липачёв
Институт математики и механики им. Н.И. Лобачевского
Казанского (Приволжского) федерального университета
amelizarov@gmail.com, dzuev11@gmail.com, elipachev@gmail.com

Аннотация

Представлены современные информационные системы, предназначенные для автоматизации полного цикла подготовки и издания электронных научных журналов. Показаны преимущества использования журнальных систем открытого доступа. Обоснован выбор системы OJS в качестве платформы построения электронного хранилища научных журналов Казанского федерального университета (КФУ). Представлен опыт реализации пилотных проектов КФУ, выполненных на базе OJS.

1 Введение

В настоящее время информационно-коммуникационные технологии (ИКТ) применяются практически на каждом этапе проведения научно-образовательной деятельности, а электронная форма представления научных и образовательных материалов неуклонно вытесняет бумажные издания. Более того, знакомство с новыми научными результатами и взаимодействие ученых происходят с помощью компьютерных сетей. Современные формы хранения, методы обработки и передачи информации основаны на цифровых технологиях, что в конечном итоге делает электронные ресурсы более привлекательными по сравнению с печатными изданиями.

С развитием глобальной телекоммуникационной инфраструктуры и появлением нового поколения мобильных устройств привычные книги и журналы менее востребованы в процессах научно-образовательной деятельности. Это подтверждают, в частности, снижающиеся (и так сравнительно небольшие) тиражи новых печатных научных изданий и увеличивающееся количество электронных научно-образовательных ресурсов. Вместе с тем, научно-образовательные электронные издания и ресурсы растворены в потоке электронной информации, объем которого лавинообразно растет; постоянно увеличивается и объем научных публикаций. Рост количества электронных документов требует их оптимальной организации, а также создания условий для обеспечения успешного поиска релевантной информации и удобства ее использования как локальному, так и удаленному пользователю.

Традиционный подход к организации хранения электронных публикаций и доступа к ним через интерфейс полнотекстовых поисковых систем является в наши дни наиболее распространенным, однако в силу растущих объемов электронной информации, а также особенностей жизненного цикла электронных научных публикаций использование стандартных сервисов и поисковых средств интернета применительно к научной электронной информации становится все менее эффективным. Актуальной является задача интеграции электронных документов, в том числе, научного и образовательного содержания, в едином информационном пространстве. В определенной степени она может быть решена путем создания специализированных информационных систем.

Интеграция информационных ресурсов традиционно является одной из базовых функций научных библиотек, ещё недавно игравших роль единственного хранилища научной информации. Сегодня они активно осваивают новые функции, связанные с оцифровкой бумажного фонда и хранением электронной информации, интеграцией электронных ресурсов и обеспечением эффективной навигации в них. Ведущие мировые научные библиотеки участвуют в формировании системы научной коммуникации и, используя сетевую инфраструктуру, налаживают новую систему сервисов интеграции научной литературы, тем самым выполняя функцию, ранее доступную только издательствам (см., например, [1, 2]).

Одним из ярких проявлений современных мировых тенденций формирования информационного общества и, в частности, информатизации библиотечно-информационной сферы стали появление и развитие информационных систем нового типа, электронных библиотек (ЭБ) – распределенных информационных систем, позволяющих надежно сохранять и эффективно использовать разнообразные коллекции электронных документов (текст, графика,

Труды 15-й Всероссийской научной конференции «Электронные библиотеки: перспективные методы и технологии, электронные коллекции» — RCDL-2013, Ярославль**, Россия, 14 – 17 октября 2013 г.**

аудио, видео и т. п.), доступные в удобном для конечного пользователя виде через глобальные сети передачи данных [3]. Составляющими ЭБ служат специализированные электронные коллекции информационных ресурсов. Сегодня общепризнано, что использование электронных библиотек, которые позволяют с учетом требований копирайта обеспечить пользователей информации удобным и представительным сервисом, является одним из наиболее перспективных способов информационного обеспечения науки, образования и культуры. ЭБ создаются как в университетах и исследовательских организациях, так и являются междисциплинарными проектами. Появление новых электронных библиотек, увеличение числа хранимых в них документов, расширение набора и повышение качества предоставляемых ими сервисов способствуют развитию науки, облегчая (иногда просто открывая) ученым единственно возможный доступ к источникам информации, а также предоставляя им эффективное средство распространения научных результатов и взаимодействия на основе сетевых коммуникаций.

В области хранения информации широко применяются различные технологии электронных библиотек, созданы соответствующие информационные системы, успешно реализован ряд проектов, проблематике ЭБ посвящено большое количество исследований (см., например, [4 – 6]). Информационные системы, поддерживающие современные электронные научные журналы или электронные версии печатных изданий, также нацелены на формирование электронных коллекций, входящих в соответствующие научные ЭБ. Однако вопросы создания самого контента, размещаемого впоследствии в электронных коллекциях ЭБ, исследованы значительно меньше. Вместе с тем, современные информационные системы управления научными журналами и публикациями являются специальным подклассом систем управления ЭБ – СУЭБ (по терминологии [7], определение приведено ниже). Следовательно, при их создании могут быть использованы развитые и широко применяемые технологии ЭБ с учетом специфики бизнес-процессов, характерных для научного издания.

Целями настоящей работы являются обзор существующих открытых проектов в области управления электронными публикациями, анализ созданных систем с позиций методологии оценки СУЭБ, разработанной в рамках европейского проекта DELOS (http://www.delos.info), а также описание опыта применения технологий ЭБ для автоматизации функционирования ряда журналов, издаваемых сегодня Казанским федеральным университетом (КФУ).

2 ИКТ в информационно-издательской деятельности

Использование ИКТ в информационно-издательской деятельности позволило не только наладить опережающий выпуск электронных версий научных изданий (книг, журналов, трудов конференций, справочников и т. д.), но и предоставить авторам, читателям, редакционным коллегиям и редакциям множество новых сервисов для работы с информацией. Так, например, составной частью практически всех современных информационных систем, используемых производителями и распространителями научной и образовательной информации, являются сервисы получения наукометрических данных, а учет этих данных при анализе публикационной активности сотрудников университетов и НИИ и выявлении наиболее перспективных направлений развития научных исследований в этих организациях становится повсеместной практикой.

Крупнейшие мировые издательства научной литературы одними из первых стали использовать ИКТ в своей работе, внедрили и постоянно развивают собственные системы электронного книгоиздания. Примерами служат информационная система издательства Springer (www.springer. com), платформа Science Direct (http://www.sciencedirect.com) издательства Elsevier (www.elsevier.com), а также система электронных публикаций научного архива arXiv.org (http://arxiv.org/). Два российских проекта – eLIBRARY.ru (http://elibrary.ru) и математический портал Math-Net.Ru (www.mathnet.ru) – по ряду используемых решений являются инновационными (см. [8, 9]).

Отметим также проект автоматизации электронного журнала Lobachevskii Journal of Mathematics (www.ljm.ru), в рамках которого был полностью автоматизирован процесс рассмотрения научной работы редколлегией журнала (которая фактически стала сетевой), включая автоматическое назначение рецензентов из базы экспертов, поддержку системы уведомлений и контроль сроков [10, 11]. Впервые в электронном математическом журнале были организованы конвертация поступающих статей и их хранение в формате MathML, что позволило реализовать систему поиска по формулам (см. [12, 13]).

Издание научных журналов, сборников статей и трудов конференций, а также формирование электронных образовательных и научных коллекций являются сегодня неотъемлемой частью научно-исследовательской и образовательной деятельности любого ведущего университета и НИИ. Для осуществления этой деятельности в 2004 – 2008 гг. в мире был создан целый ряд информационных систем управления научными журналами и публикациями. С практической точки зрения наибольший интерес вызывают те из них, которые являются свободно распространяемыми (open source), – благодаря открытому коду появляется возможность доработки таких систем и придания им требуемой функциональности. Важным обстоятельством является также наличие у многих таких систем групп разработчиков, выкладывающих на соответствующие сайты новые модули, часто выполненные инновационными методами с применением передовых информационных технологий.

В функционале современных информационных систем управления научными журналами обязательно должны присутствовать сервисы, регулирующие процесс рецензирования и обеспечивающие коллективное редактирование электронных документов. Кроме того, системы такого типа должны предоставлять такие редакционные сервисы, как классификация, аннотирование, выделение метаданных, публикация, долгосрочное хранение, конвертирование, распространение, синдикация, статистика использования, харвестинг, объединение в коллекцию, взаимодействие с институциональными репозиториями, контроль доступа, подписка, рассылка уведомлений, новые поступления. Вместе с тем, современные информационные системы управления электронными научными публикациями не ограничиваются сервисами удаленного представления статей в научный журнал и их дальнейшей обработки для окончательной публикации, а обеспечивают доступ к сформированному контенту и расширенный поиск (по автору, названию статьи, ключевым словам и др.) в соответствующих электронных коллекциях, т. е. в полном объеме реализуют функциональные возможности, присущие электронным библиотекам. С этой точки зрения электронный научный журнал можно рассматривать как научную ЭБ, оперирующую статьями журнала как информационными объектами. Следовательно, при создании информационных систем управления электронными научными публикациями могут быть использованы хорошо развитые технологии ЭБ, а при анализе существующих систем такого типа – подходы, разработанные при формировании концептуальных моделей, обобщающих накопленный опыт в сфере создания и использования ЭБ, в частности, эталонной модели ЭБ (Digital Library Reference Model, DLRM) [7], построенной в 2005 – 2007 гг. в рамках проекта DELOS. Эталонная модель была предназначена для разработки более узких моделей ЭБ с конкретной архитектурой и последующей их реализации в рамках создаваемых информационных систем.

3 Системы управления электронными журналами – специальный класс СУЭБ

Как известно, в модели DELOS DLRM выделено три основных понятия для разграничения того, что называется электронной библиотекой:

- *ЭБ* – конкретная электронная библиотека с ее контентом, пользователями, правилами работы и пр.;
- *система ЭБ* – программное обеспечение (ПО), на основе которого создаются ЭБ, т. е. СУЭБ, адаптированная для управления конкретной ЭБ, вместе со специальными приложениями;
- *система управления ЭБ (СУЭБ)* – ПО для создания и управления системами ЭБ, реализующее функциональные возможности ЭБ.

В ролевом аспекте (с точки зрения разных категорий пользователей) в модели DELOS DLRM рассматриваются: конечный пользователь ЭБ; разработчик ЭБ; системный администратор ЭБ и разработчик приложений для ЭБ. Соответственно должны быть сформированы четыре уровня пользовательских представлений. Наконец, в рассматриваемой модели выделены шесть ключевых областей, в каждой из которых вводятся и определяются свои сущности и их свойства: архитектура, информационное пространство, функциональные возможности, пользователи, политики и качество предоставляемых услуг. Указанные области (критерии) оценки универсальны и могут быть применены без потери общности для анализа практически любой информационной системы в смежных предметных областях.

Таким образом, в соответствии с моделью DELOS DLRM, электронная библиотека – это система для сбора, сохранения в течение длительного времени информационных объектов и управления ими в соответствии с принятыми политиками и измеряемым качеством, которая предоставляет сообществам пользователей специализированные функциональные возможности, связанные с содержанием информационных объектов.

Любой электронный научный журнал – это хранилище статей, к которому предоставляется доступ конечным пользователям, а система управления электронным журналом – это набор программного обеспечения, реализующий функции хранения, сбора и предоставления доступа к информационным объектам. Таким образом, система управления электронными журналами является разновидностью СУЭБ, также имеет ролевую модель пользователей и использует метаданные при формировании выпусков журнала и описаний статей. Особенность состоит лишь в том, что в системе управления электронным журналом должен быть предусмотрен более сложный процесс публикации информационных объектов, отражающий фактическую работу любого издательства, публикующего научные журналы.

Приведем результаты анализа существующих информационных систем управления электронными журналами.

4 Информационные системы управления электронными журналами

Большинство современных научных изданий представлено в интернете, периодические издания имеют сайты с электронными версиями опубликованных материалов или аннотациями статей. Эти сайты поддерживаются либо автономной системой управления, обеспечивающей навигацию по контенту, либо являются частью какой-либо объемлющей информационной системы (например, университета в целом). Как правило, эти разработки ограничены функционально, не учитывают специфики научных журналов и, как следствие, не обеспечивают автоматизации всех бизнес-процессов, связанных с управлением электронными научными журналами. По этой причине подобные системы не анализировались. Кроме того, нами рассматривались только

некоммерческие, свободно распространяемые платформы, причем предпочтение отдавалось развивающимся проектам с реализованной или планируемой к реализации русской локализацией.

При анализе названных систем использовались результаты работы [14], в которой сравнение проведено по набору ключевых параметров; среди них – базовое программное обеспечение (как следствие, различная степень сложности установки), количество успешных инсталляций, наличие и полнота сопровождающей технической документации.

В итоге проведенного анализа были отобраны следующие системы управления электронным журналами.

Open Journal System (OJS) (http://pkp.sfu.ca/?q =ojs) – программная система с открытым исходным кодом для управления электронными научными журналами; разрабатывается в рамках проекта Public Knowledge Project (http://pkp.sfu.ca/about) в Канаде университетами Саймона Фрейзера (Simon Fraser University), Британской Колумбии (University of British Columbia), Советом университетских библиотек Онтарио (Ontario Council of University Libraries) и в США Школой образования в Стэнфордском университете (School of Education at Stanford University), университетом Питтсбурга (University of Pittsburgh) и Калифорнийской электронной библиотекой (California Digital Library).

Система OJS распространяется по лицензии GNU/GPL. Проект постоянно развивается, выходят новые версии системы, доступна стабильная полная версия для самостоятельной установки. По состоянию на декабрь 2011 года система OJS используется более чем 11500 журналами по всему миру, часть которых зарегистрирована на сайте проекта (http://pkp.sfu.ca/ ojs-journals). В этом списке есть и несколько российских журналов, в частности, Russian Journal of Herpetology (http://www.folium.ru/ rjh/index.php/rjh), а из электронных журналов Санкт-Петербургского государственного университета (http://ojs.spbu.ru/) там представлен «Петербургский психологический журнал». Платформа OJS внедряется в научно-издательскую инфраструктуру Украины как общегосударственная платформа научной периодики (см., например, [15]). Отметим также проект перевода на платформу OJS ряда научных периодических изданий Казанского федерального университета (см. [16]).

Система OJS представляет собой единую платформу для управления электронными журналами, поддерживающую широкий спектр бизнес-моделей для периодики и настроек предоставления доступа от полностью открытого доступа к ресурсам до предоставления кратких аннотаций и коммерческой подписки. Четкое разделение позволяет использовать систему как единую общую платформу для управления всеми периодическими ресурсами отдельной научно-исследовательской или образовательной организации, поскольку размещаемые журналы управляются абсолютно независимо, и при этом настройки одного из них никак не влияют на работу другого.

Система OJS настраивается как облачный программный комплекс, может развертываться и управляться локально, все бизнес-процессы настраиваются непосредственно редакторами каждого конкретного издания. OJS предоставляет специальный инструментарий для чтения и просмотра публикаций как в pdf-, так и в html-формате, доступен ряд функций для работы с библиографией, метаданными и др.

Система OJS имеет модульную архитектуру, хорошо документирована, что позволяет при необходимости не только освоить имеющиеся функции, но и разработать собственные классы и модули. Система имеет MVC-структуру (Model-View-Controller), соответственно хранилище данных, пользовательские интерфейсы и управляющие функции разделены на разные уровни взаимодействия. Несмотря на кажущуюся сложность, такая архитектура обеспечивает отказоустойчивость, производительность, гибкость и масштабирование всей системы.

Система OJS платформонезависима и может быть установлена как под ОС Windows, так и на Unix-подобных операционных системах, используются свободно распространяемые PHP и Apache, а также СУБД (MySQL, PostgreSQL); процесс установки является стандартным для систем управления сайтом. Важно также отметить, что для OJS имеется многоуровневая документация.

В дистрибутиве системы OJS заложена поддержка русского языка. В стандартную поставку входит ряд библиотек и расширений, предоставляющих различные функции: обработку цитат и отображения статей в pdf- или html-формате, анализатор трафика phpMyVisites, шлюз METS для обмена данными, добавление OpenURL-дескриптора к статье, WYSIWYG-редактор страниц и другие.

Система OJS корректно работает не только на персональных компьютерах, но и на смартфонах и других мобильных устройствах, что актуально в связи наметившейся ориентаций информационных технологий на BYOD (Bring Your Own Device). Возможно также подключение модуля приема оплаты, отвечающего за предоставление платного доступа к ресурсам.

Система OJS имеет ролевую модель пользователей с разными правами доступа и многоступенчатый процесс публикации ресурсов, который поддерживает все стадии жизненного цикла статьи от первоначальной загрузки ее авторской редакции до размещения в интернете окончательного варианта и формирования соответствующих индексов и ссылок. Функционал системы позволяет реализовать взаимодействие участников редакционного процесса в режиме онлайн. Интерфейсные модули OJS реализованы в виде наборов шаблонов Smarty (http:// smarty.php.net), что позволяет гибко изменять пользовательские интерфейсы системы. Интерфейс и функциональные возможности системы OJS могут

быть настроены и адаптированы под бизнес-процесс конкретного научного издания.

Еще раз подчеркнем, что возможности системы OJS и приемы работы в ней представлены в большом количестве руководств и публикаций (например, [17]). Преимуществом OJS как базовой платформы является отлаженная методика использования (см. http://pkp.sfu.ca/ojs-journals). Наличие постоянно пополняемой галереи модулей (http://pkp.sfu.ca/support/forum/viewforum.php?f=28) также служит важным обстоятельством и позволяет учесть особенности научных изданий и не пытаться унифицировать издательскую деятельность в полном объеме (в настоящее время это вряд ли возможно и, по нашему мнению, не нужно). Технология создания программных модулей основана на открытом коде, что позволяет включать в систему сервисы, учитывающие специфику отдельных научных изданий.

ePublishing Toolkit (ePubTK, https://dev.livingreviews.org/projects/epubtk#) – издательский набор инструментов, разрабатываемый обществом Max Planck Society (https://dev.livingreviews.org/ projects/epubtk/) для управления семейством электронных научных журналов научного онлайн-издательства Living Reviews (www.livingreviews.org). Отдельного законченного дистрибутива для установки не существует, однако все исходные коды системы доступны в онлайн-репозитории разработчиков. Отсутствие версионности не позволяет сделать вывод о периодичности обновлений и реальных планах развития системы.

Информационное пространство системы ePubTK состоит из семейства журналов, которое в свою очередь делится на отдельные журналы. Каждый журнал является контейнером для публикаций, практически все функциональные возможности системы ePubTK связаны именно с журналами. При создании каждому журналу в рамках одной инсталляции системы присваивается уникальный идентификатор, который в дальнейшем используется в различных сценариях работы системы.

Архитектурно система ePubTK состоит из компонент, которые могут работать независимо. Каждый компонент содержит набор функций для работы с отдельным классом объектов системы. Базовые функции, требуемые во многих компонентах, выполнены в виде общих библиотек. Отдельный компонент отвечает за создание публикаций из исходного материала (pubBuilder) и представления в Вебе; для управления ссылками используется компонент refdb; бэк-офис управления жизненным циклом и бизнес-процессами издательства обеспечивается специальной подсистемой управления EIMS (Editorial Information Management System), которая также является отдельным компонентом ePubTK (http://www.carpet-project.net/en/catalogue/detail/ eims-editorial-information-management-system-workflowsupport-living-reviews/).

Гибкость конфигурирования системы ePubTK для разных журналов достигается за счет использования шаблонов XSLT, на основе которых генерируются веб-страницы, шаблоны писем и т. п.

Декларируется максимальное соответствие открытым стандартам OpenSeacrh, OAI-PMH, unAPI, авторизация возможна с помощью OpenID.

Система ePubTK также имеет ролевую модель пользователей с разными правами доступа и многоступенчатый процесс публикации ресурсов, который поддерживает все стадии жизненного цикла статьи от первоначальной загрузки черновика до размещения итогового варианта в интернете, адаптированные под процессы Living Reviews.

Систему ePubTK можно установить в ОС MS Windows (win32) и операционных Linux-системах, для работы требуются установка Python (версии не ниже 2.3), а также ряд пакетов Python (см. https://dev.livingreviews.org/projects/epubtk/wiki/Requirements), что делает процесс установки достаточно трудоемким. Настройка системы требует достаточно высокой квалификации персонала.

Digital Publishing System (DPubS, http://dpubs.org/) – свободно распространяемая информационная система для онлайн-публикации академических научных и образовательных журналов, трудов конференций и монографий. Она разрабатывалась в 2004 – 2008 гг. в США Корнэльским университетом (Cornell University) и университетом Пенсильвании (Pennsylvania State University). На базе этой системы Библиотекой Корнэльского университета реализован проект Project Euclid (www.projecteuclid.org). С 2008 года дальнейших обновлений системы не было. На данный момент времени на базе DPubS реализовано порядка 10 проектов, так или иначе связанных с организациями, разработавшими эту систему.

Основной особенностью системы DPubS можно считать то, что инициатором ее разработки выступила Библиотека Корнэльского университета (с целью создания системы электронного издательства), а не различные научные и образовательные сообщества. Это отразилось в особенностях функциональных возможностей системы. В частности, система DPubS спроектирована с учетом проблем по обеспечению сохранности информационных ресурсов и отказоустойчивости, которые остро стоят перед всеми ЭБ; кроме того, имеется поддержка работы с издательским ПО и такими хранилищами информационных объектов (институциональными репозиториями), как DSpace или FEDORA (Flexible Extensible Digital Object Repository Architecture).

Система DPubS представляет собой набор взаимосвязанных сервисов и имеет модульную архитектуру. Функционально DPubS состоит из модуля объединения в коллекции, редакционного сервиса, сервиса индексирования, поискового медиатора, модуля обратной связи, репозитория, сервисов подписки и модулей пользовательского интерфейса и администрирования.

Редакционный сервис обеспечивает первоначальную загрузку статей и передачу их рецензентам, дальнейшую подготовку и публикацию выпусков журналов и финальную их загрузку в хранилище

DPubS. Также реализована ролевая модель пользователей с разными правами доступа. Имеется возможность предоставления как платного, так и бесплатного доступа к ресурсам.

Документация к системе не соответствует реально выпущенной версии системы, функциональные возможности ряда модулей описаны недостаточно полно, отсутствует какое-либо руководство пользователя.

Установка DPubS требует учета особенностей архитектуры и внутренних взаимосвязей элементов системы. Отсутствие обновлений с 2008 года и соответствующей документации делают установку и внедрение этой системы весьма нетривиальной задачей.

GAPWorks (http://gapworks.berlios.de/) – электронная издательская системы, которая разрабатывалась в рамках проекта немецких академических издательств (German Academic Publishers, GAP), финансируемого Немецким научно-исследовательским фондом (DFG). GARWorks предоставляет компоненты для обеспечения работы электронного издательства (с поддержкой процесса рецензирования), управления пользователями, ролями и т. п.

Система GAPWorks реализована с использованием PHP и СУБД PostGreSQL. Она обеспечивает процесс рецензирования, функции управления пользователями, поддержку OAI-PMH, имеет настраиваемый набор шаблонов. Несмотря на то, что дистрибутив GAPWorks доступен для скачивания, сведений о развитии системы с 2006 года нет, данные о реализованных проектах также отсутствуют.

Ambra Publishing System (Ambra, http://www.topazproject.org/trac/wiki/Ambra) – система для электронного издательства, разработанная некоммерческой организацией Topaz (www.topazproject.org) на базе одноименной платформы и связанная с Публичной научной библиотекой (Public Library of Science, PLOS, www.plos.org). Ambra – это веб-приложение, имеющее сервис-ориентированную архитектуру, для публикации материалов исследований во всех областях науки и призванное помочь «оживить» опубликованные научные статьи – система позволяет пользователям оценивать, аннотировать и комментировать публикации, что дает возможность сообществу авторов и читателей оперативно обмениваться новыми научными идеями. Система Ambra также используется в качестве платформы для размещения ряда журналов PLOS.

Информационная модель системы Ambra основана на платформе Topaz, в качестве хранилища данных используются специально настроенные репозиторий FEDORA (www.fedora-commons.org) и СУБД Mulgara (RDF база данных с открытым исходным кодом, www.mulgara.org). Для характеристики системы Ambra целесообразно описать платформу Topaz, на которой она построена.

Topaz – это библиотека программ управления объектами, использующая технологию объектно-реляционного отображения и позволяющая разрабатывать собственные хранимые классы и объекты в соответствии с парадигмой объектно-ориентированного программирования. Все данные приложений хранятся с использованием RDF, для описания отображения объектов в RDF используются классы Java. Также в библиотеку встроена поддержка специального blob-хранилища для хранения данных типа blob. В качестве объектного хранилища метаданных используется СУБД Mulgara, для blob-данных (статьи, тексты, фото, видео и др.) – репозиторий FEDORA.

Основной особенностью системы Ambra можно считать использование технологии объектно-реляционного отображения – при разработке системы, а также нереляционной СУБД – в качестве хранилища части информационных объектов. Поскольку взаимодействие между отдельными модулями системы Ambra осуществляется по протоколу TCP, структура система может быть распределенной. Процесс загрузки публикаций упрощен и состоит всего из двух ступеней (загрузка пользователем и подтверждение администратором), отсутствуют специальные роли для редакторов и рецензентов. Поскольку все статьи хранятся в репозитории FEDORA, а сами статьи в системе Ambra связываются с информационными объектами этого репозитория, то фактически для материалов системы Ambra становятся доступны все функции FEDORA API, например, обеспечивается поддержка протокола OAI-PMH.

Веб-приложение Ambra можно установить как для ОС Windows, так и операционных UNIX-систем, однако дистрибутив не содержит мастера-установщика, в связи с чем установка комплекса становится весьма непростой. Последний релиз системы датирован 2009 годом, поэтому сделать выводы о дальнейшем развитии проекта затруднительно.

Drupal E-Journal (http://drupal.org/project/ejournal) – специально разработанный модуль управления электронным журналом, созданный для известной системы управления контентом Drupal. Изначально этот модуль разрабатывался как аналог системы OJS для Drupal и предоставляет функции управления журналами, их выпусками и статьями, также имеется поддержка ролей пользователей и прав доступа. Поскольку система Drupal E-Journal архитектурно является отдельным модулем Drupal, то возможно совместное использование с ней других надстроек и модулей Drupal, что представляется весьма полезным.

На данный момент времени модуль не закончен, поэтому говорить о полнофункциональной системе управления электронным журналом нельзя. Последняя версия выпущена в 2011 году, также доступна стабильная сборка модуля для Drupal версий 5.x и 6.x.

Сравнительная таблица систем управления электронными научными журналами

Система / Критерий	OJS	ePubTK	DPubS	GAPWorks	Ambra	e-Journal
Пользователь: ролевая модель, политики, группы	Имеются ролевая модель и регистрация пользователей. Права доступа и доступные функции зависят от роли пользователя	Имеется ролевая модель пользователей, авторизация возможна с помощью OpenID	Имеется ролевая модель пользователей	Декларируется возможность управления пользователями и ролями	Имеется идентификация пользователей, ролевая модель упрощена	Имеется поддержка ролей пользователей и прав доступа
Информационное пространство: информационный объект, контент, метаданные, коллекции	Имеется иерархия объектов – журнал, выпуски, статьи. Декларируется соответствие метаданных OAI-PMH, есть возможность создавать метаданные статей. Метаданные хранятся в БД, используется единая схема для всех журналов	Иерархия объектов: семейство журналов делится на отдельные журналы; каждый из них является контейнером для публикаций; метаданные соответствуют OAI-PMH	Поддерживаются метаданные, но есть существенные ограничения; метаданные генерируются для всех журналов, возможна небольшая настройка администратором системы		Используются информационные объекты и особенности FEDORA, доступны все функции FEDORA API, обеспечивается поддержка протокола OAI-PMH	
Функциональные возможности: процессы публикации и рецензирования, контент, управление системой, персонализация	Имеются настраиваемые процессы рецензирования и публикации. Отслеживается весь жизненный цикл от черновика до законченной публикации. Есть возможность видоизменять жизненный цикл статьи в рамках одного журнала. Управление системой простое, часть операций может быть выполнена без предварительного изучения документации. Персонализация достигается за счет использования шаблонов Smarty	Гибкость конфигурирования для разных журналов достигается за счет использования шаблонов XSLT. Для данной системы журнал – это минимальный объект, с которым связаны все функции. Процесс настройки системы требует высокой квалификации	Имеются многоступенчатые процессы публикации и рецензирования. Есть поддержка метаданных, однако не ясно, как они хранятся и к какому стандарту относятся. Отсутствует описание ряда модулей. Нет новых версий с 2008 года	Декларируется возможность обеспечения работы электронного издательства с поддержкой процесса рецензирования	Процесс загрузки публикаций упрощен и состоит из двух ступеней (загрузка пользователем и подтверждение администратором), отсутствуют специальные роли для редакторов и рецензентов	Предоставляет функции управления журналами, их выпусками и статьями

Качество предоставления услуги: мультиязычность; безопасность; отказоустойчивость; расширяемость; модульность; кросс-платформенность и т. д.	Система платформонезависима. Для обеспечения безопасной работы используются HTTP-сессии; действия логгируются. Имеется встроенная поддержка мультиязычности, в т. ч. русского языка. Использование MVC-парадигмы обеспечивает отказоустойчивость и масштабируемость. Установка системы производится с помощью специального мастера-установщика и весьма проста	Система может быть установлена как под ОС Windows, так и для Linux, однако процесс установки достаточно сложен	Контроль доступа основан на скрытии/показе прямых ссылок на документы, таким образом, документ всегда можно найти, зная прямую ссылку на него. Документация отсутствует, версия системы выпущена давно. Для системы требуется отдельный сервер, на котором не должно быть никаких других веб-приложений	Последняя версия системы датирована 2006 годом, дальнейших обновлений ПО либо документации нет	Последний релиз датирован 2009 годом, систему можно установить на разные ОС, однако дистрибутив не содержит мастера-установщика	

HyperJournal (http://www.hjournal.org) – проект, инициированный в 2004 году Groupement de Recherche Europeen (GDREplus) и поддержанный Centre National de la Recherche Scientifique (CNRS); в настоящее время развивается также с помощью волонтеров при поддержке Dipartimento di Scienze della Politica, University of Pisa.

Система HyperJournal устанавливается под ОС Linux, для работы требуется дополнительная установка PHP и СУБД MySQL. Дистрибутив системы доступен по адресу http://sourceforge.net/projects/hyperjournal/.

Выше представлена сравнительная таблица, отражающая проведенный анализ рассмотренных систем. При этом использованы те же критерии оценки, которые применялись при исследовании СУЭБ в рамках проекта DELOS. Заметим лишь, что нами специально делался акцент на функциональных возможностях систем, связанных именно с редакционно-издательской деятельностью. В случае, когда однозначный вывод о соответствии тому или иному критерию сделать было невозможно, поле оставлено не заполненным. В частности, по этой причине не удалось включить в таблицу информацию о системе HyperJournal.

В заключение настоящего раздела отметим, что описание информационных систем управления научными журналами и публикациями, имеющихся в свободном доступе, содержится также в работах [18, 19].

Заключение

Проведенный анализ проектов создания системы управления электронными научными журналами позволил сформулировать следующие выводы:

- практически все информационные системы, связанные с электронными журналами и электронными издательствами (OJS, ePubTK, DPubS, Ambra), были созданы в период 2004 – 2008 гг. и разрабатывались для обеспечения функционирования конкретных электронных изданий; это привело к существенным различиям как в архитектуре систем, так и функциональных возможностях;
- не существует универсальной модели системы управления электронным журналом с описанием конкретных требований и сервисов; разработчики таких систем часто брали за основу опыт создания конкретных систем управления электронными библиотеками и не использовали в полном объеме результаты, достигнутые в области ЭБ;
- практически все проекты создания систем управления электронными научными журналами, рассмотренные выше, поддерживают общепринятые стандарты в области интеграции и обмена данными;
- на текущий момент времени большинство проектов, представленных выше, не получило дальнейшего существенного развития; исключением является всего лишь один активно развивающийся проект – система Open Journal Systems.

Таким образом, учитывая вышесказанное, целесообразно в качестве основы системы управления электронными научными журналами использовать именно Open Journal Systems как наиболее динамично развивающуюся, хорошо документированную информационную систему. Именно такое решение было принято в 2012 году в Казанском федеральном университете. В качестве пилотного проекта произведена установка системы OJS на серверных мощностях университета, осуществлен перевод ряда журналов под управление этой системы, идет подготовка к всестороннему тестированию системы для ее дальнейшей интеграции в единую научно-образовательную среду университета.

При доработке платформы единого электронного хранилища научных журналов КФУ признано, что требования к информационной журнальной системе, названные выше в п. 2, должны быть дополнены возможностью локализации на русский и татарский языки, способностью системы управлять междисциплинарным контентом, наличием или возможностью подключения семантических инструментов обработки информации (см., например, [10, 11]). В частности, для математических журналов исследовались способы подключения скриптов поиска по фрагментам формул (см. [20]), а также методы формирования математических электронных коллекций (см. [21]). Стратегическими являются вопросы внедрения технологий Cloud Computing (например, [22]).

Литература

[1] Хокинс К. Научная библиотека как издательство: опыт Мичиганского университета (США) // Вестник Пермского университета. Серия История. – 2009. – Вып. 3 (10). – С. 119-122.

[2] Хокинс К. Библиотеки как издатели: перемены в жизненном цикле информации. – http://www.ultraslavonic.info/talks/20050304.ru.pdf; http://www.umich.edu/~kshawkin/ talks/20050304.pdf.

[3] Ершова Т.В., Хохлов Ю.Е. Межведомственная программа «Российские электронные библиотеки» // Электронные библиотеки: рос. науч. электронный журн. – 1999. – Т. 2, Вып. 2. – http://www.elbib.ru/index.phtml?page=elbib/rus/journal/1999/part2/ershova.

[4] A Guide to Institutional Repository Software. 3rd Edition. Open Society Institute. 2004. – http://www.soros.org/openaccess/pdf/OSI_Guide_to_IR_Software_v3.pdf.

[5] Candela L., Castelli D., Fuhr N., Ioannidis Y., Klas C.-P., Pagano P., Ross S., Saidis C., Schek H.-J., Schuldt H., Springmann M. DELOS Workpackage 1. D1.4.1 – Current digital library systems: user requirements vs provided functionality. – 2005.

[6] Candela L., Castelli D., Fuhr N., Ioannidis Y., Klas C.-P., Pagano P., Ross S., Saidis C., Schek H.-J., Schuldt H., Springmann M. Current digital library systems: user requirements vs provided functionality. IST-2002-2.3.1.12. Technology-enhanced Learning and Access to Cultural Heritage. March 2006.

[7] Candela L., Castelli D., Dobreva M., Ferro N., Ioanni-dis Y., Katifori H., Koutrika G., Meghini C., Pagano P., Ross S., Agosti M., Schuldt H., Soergel D. The DELOS Digital Library Reference Model Foundations for Digital Libraries. IST-2002-2.3.1.12. Technology-enhanced Learning and Access to Cultural Heritage. Version 0.98, December 2007. – http://www.delos.info/files/pdf/ ReferenceModel/DELOS_DLReferenceModel_0.98.pdf.

[8] Жижченко А.Б., Изаак А.Д. Информационная система Math-Net.Ru. Применение современных технологий в научной работе математика //Успехи матем. наук. – 2007. – Т. 62, Вып. 5 (377). – С. 107-132.

[9] Жижченко А.Б., Изаак А.Д. Информационная система Math-Net.Ru. Современное состояние и перспективы развития. Импакт-факторы российских математических журналов // Успехи матем. наук. – 2009. – Т. 64, Вып. 4 (388). – С. 195-204.

[10] Глухов В.А., Елизаров А.М., Липачев Е.К., Малахальцев М.А. Электронные научные издания: переход на технологии Семантического веба // Электронные библиотеки. – 2007. – Т. 10, Вып. 1. – http://www.elbib.ru/index.phtml?page=elbib/rus/journal/2007/part1/GELM.

[11] Веселаго В.Г., Елизаров А.М., Липачёв Е.К., Малахальцев М.А. Формирование и поддержка физико-математических электронных научных

изданий: переход на технологии семантического веба// В кн. «Научно-исследовательский институт математики и механики им. Н.Г. Чеботарева Казанского государственного университета. 2003 – 2007 гг.». Кол. монография под ред. А.М. Елизарова. – Казань: Изд-во Казан. ун-та, 2008. – С. 456-476.

[12] Елизаров А.М., Липачев Е.К., Малахальцев М.А. Технологии Semantic Web в практике работы электронного журнала по математике //Тр. 8-й Всерос. науч. конф. «Электронные библиотеки: перспективные методы и технологии, электронные коллекции» – RCDL'2006, Суздаль, Россия, 2006. – С. 215-218.

[13] Елизаров А.М., Липачев Е.К., Малахальцев М.А. Веб-технологии для математика: Основы MathML. Практическое руководство. – М.: Физматлит, 2010. – 216 с.

[14] Chýla Ch. What open source webpublishing software has the scientific community for e-journals?// In CASLIN 2007, Stupava (Slovak Republic). – http://eprints.rclis.org/10055/.

[15] Состояние и перспективы развития научной периодики Украины. – http://nv.nmu.org.ua/index.php/ru/glavnaya/38-ruscat/novosti/86-sostoyanie-i-perspektivy-razvitiya-nauchnoj-periodiki-ukrainy.

[16] Бабин Е.Н., Елизаров А.М., Липачёв Е.К. Открытые информационные системы управления научными публикациями как основа построения научных электронных библиотек Казанского университета // Учёные записки института социальных и гуманитарных знаний. – 2013. – №1 (11). – С. 55-59.

[17] Willinsky J., Stranack K., Smecher A., MacGregor J. Open Journal Systems: a complete guide to online publishing. Simon Fraser University Library, 2010. – 273 p. – http://pkp.sfu.ca/ojs/docs/userguide/2.3.1/index.html.

[18] Cyzyk M., Choudhury S. A survey and evaluation of open-source electronic publishing systems. 2008. – http://jhir.library.jhu.edu/handle/1774.2/ 32737.

[19] Tools and Platforms. – http://www.openoasis.org/index.php?option=com_content&view=article&id=353&Itemid=379/

[20] Елизаров А.М., Липачёв Е.К., Хохлов Ю.Е. Технологии облачных вычислений для поддержки функционирования электронного научного журнала // Материалы Межд. науч.-практ. конф. «Информационные технологии в образовании и науке – ИТОН-2012» (8 – 12 октября, Казань). Казань: Казанский ун-т, 2012. – С. 82-85. – http://vuz.exponenta.ru/PDF/NAUKA/Sbornik12ito.pdf.

[21] Елизаров А.М., Липачёв Е.К. Технологии формирования и поддержки электронных научных математических коллекций: опыт Казанского университета // Система обеспечения российских организаций научно-технической информацией в электронном виде. Отчетная конференция по проекту МОН. – http://conf.neicon.ru/index.php/science/mon2012/paper/ view/31/28.

[22] Елизаров А.М., Липачёв Е.К., Хохлов Ю.Е. Технологии облачных вычислений для поддержки функционирования электронного научного журнала // Материалы Межд. науч.-практ. конф. «Информационные технологии в образовании и науке – ИТОН-2012» (8 – 12 октября, Казань). Казань: Казанский ун-т, 2012. С. 82-85. – http://vuz.exponenta.ru/PDF/NAUKA/Sbornik12ito.pdf.

Open scientific e-journals management systems and digital libraries technology

Alexander Elizarov, Denis Zuev, Eugene Lipachev

Modern information systems designed to automate full cycle of preparation and publishing of electronic scientific journal are presented. Benefits of using open-access journal systems are showed.

In our work we also discuss the choice of OJS as a digital publishing system for scientific journals of Kazan Federal University.

♣Работа поддержана РФФИ (проекты №№ 12-07-00667 и 12-07-97018-р_поволжье)

A Model for Integrating the Publication and Preservation of Journal Articles

© Kevin S. Hawkins
University of Michigan,
Ann Arbor
kshawkin@umich.edu

Abstract

There are policy, technical, and workflow gaps in library efforts to preserve online journal literature. Since libraries are increasingly involved in journal publishing, HathiTrust, a shared preservation-quality digital repository, is a natural place to archive and provide access to journal literature to ensure its long-term preservation and discoverability. The U-M Library is funding the creation of mPach, an open-source, end-to-end publishing system in which archiving in HathiTrust happens as a byproduct of publication rather than being carried out after the fact. The architecture of mPach, its envisioned workflow, and plans for creating a shared infrastructure for publishing open-access journals are all summarized.

1 The deficit in journal preservation

Until quite recently, publishers produced documents on physical media, and libraries acquired and preserved copies of these documents. But in the era of the Internet, when publishers host content online, the library's role in acquiring and preserving the content is in jeopardy: without special licensing arrangements such as those often provided by open-access journals, a library has no legal right to make a copy of the content for preservation.

Various business models have evolved to address this situation, especially for journals, which are increasingly available only online. For non-open-access journals, research libraries often negotiate the right to create a digital copy of any content acquired during the period of subscription [1] and make this content available only to their patrons [2], though few are equipped to provide this kind of restricted access and archiving with integrated browse and search functions. To address the more pressing concern of publishers going out of business without *any* libraries holding a copy of the content, libraries and publishers have collaborated in initiatives like LOCKSS [3], CLOCKSS [4], and Portico [5] in order to guarantee that one or more copy of the content will become available if it is no longer available from the publisher. Similarly, the Koninklijke Bibliotheek and Elsevier reached an agreement in 2002 whereby the KB will preserve Elsevier journals under terms similar to those governing journals that use LOCKSS, CLOCKSS, and Portico [6]. Still, there are problems with these models. LOCKSS and CLOCKSS use web crawling, which captures only the appearance of webpages but not their underlying structure or search functionality. Portico and the KB, on the other hand, rely on publishers to deliver journal articles in valid file formats, and not just the version first published but also any corrected versions of these articles.

One way to ensure that a library always has access to the latest content is for the library to operate the very system used to publish the journal. A survey in 2010 of a cross-section of North American academic libraries found that, of 144 responding institutions, 43 offered "operational publishing services" to their scholars at the institution [7]. Of these 43 institutions, most host publications using open-source software such as Open Journal Systems (OJS) [8] or DSpace [9], while about a quarter use Digital Commons [10], a hosted platform provided by bepress. Unfortunately, all of these platforms deliver to users only those files (primarily PDF files) created and uploaded by a journal editor. Since the library is not in a position to control the software and workflows used to create these files, the library can only provide bitwise preservation of the files, severely hampering future migration of the content.

2 A higher standard for preservation

Since libraries are increasingly involved in journal publishing, HathiTrust [11], a shared preservation-quality digital repository, is a natural place to archive and provide access to journal literature to ensure its long-term preservation and discoverability. HathiTrust already archives and provides access to reformatted library holdings, but the University of Michigan Library, a founding member of HathiTrust, sees an opportunity to use HathiTrust for publishing born-digital journals as well. To develop an infrastructure in support of low-cost university-based publishing that addresses the needs and values of both content creators and librarians, the U-M Library is funding the creation

Proceedings of the 15th All-Russian Conference "Digital Libraries: Advanced Methods and Technologies, Digital Collections" — RCDL-2013, Yaroslavl, Russia, October 14-18 2013.

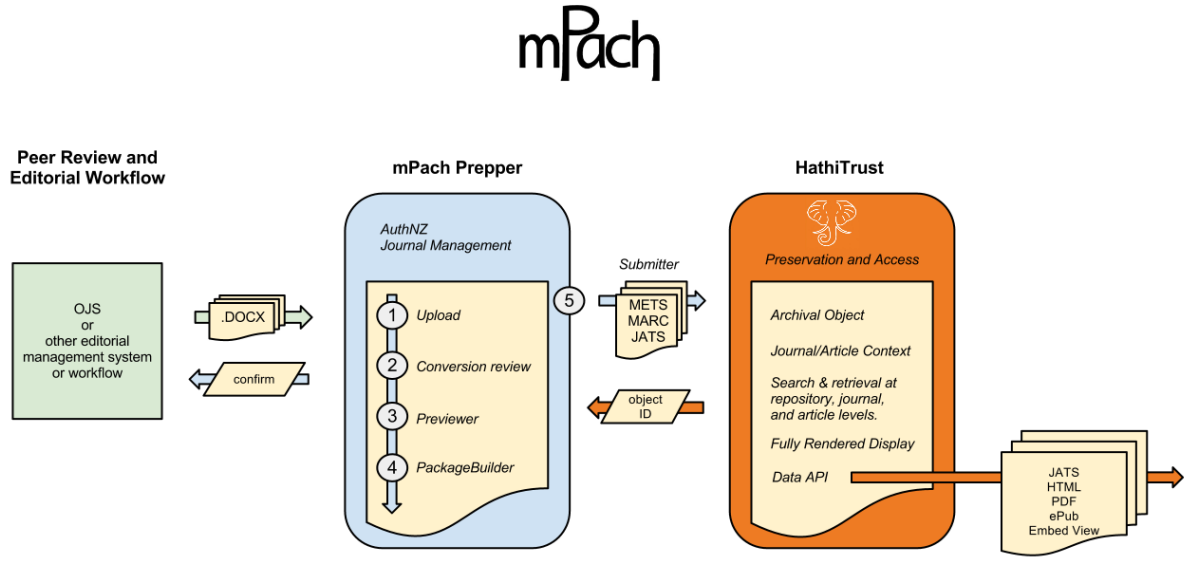


Figure 1: Major parts of mPach

of mPach [12], an open-source, end-to-end publishing system in which the act of publishing and the act of archiving are unified. In other words, archiving in HathiTrust happens as a byproduct of publication rather than being carried out after the fact. mPach leverages existing components of HathiTrust and available open-source software where appropriate.

Archiving is not as simple as saving a copy of a file produced by a journal editor, as OJS and institutional repositories generally do. Instead, the content needs to be stored in a format that allows digital preservation. PDF/A, a non-proprietary variant of the PDF family standardized as ISO 19005, is often suggested for such needs, but even a PDF/A file is poorly suited for use with screen readers for the visually impaired and for any non-paginated display, and is suboptimal even for searching and data mining.

Rather than preserving the paginated appearance of a document, the text of the article needs to be stored in a format that reflects its structure and semantics, with associated media in formats that can be preserved and rendered. mPach has developed a specification for journal articles that uses the Journal Article Tag Suite (JATS), an application of NISO Z39.96-2012 [13], for the text and stores this with high-quality versions of media objects and with a METS record containing structural and preservation metadata.

3 An overview of mPach

There are three major parts of mPach (see also figure 1), each of which includes components in various stages of development at the time of writing:

- **the peer review and editorial system**: what authors and reviewers interact with
- **Prepper**: what prepares the article for ingest into HathiTrust for archiving and publication
- **modified HathiTrust components**: various modifications to existing components of the HathiTrust environment to support born-digital journal articles

As a modular system, mPach could be used with any peer review and editorial system that is capable of interacting with Prepper; however, the developers have chosen to provide OJS as the default option. Despite having no support for digital preservation, OJS is already widely used for library-based journal publishing, and mPach's integration with this software will allow for a smooth transition of journals already published using OJS into the HathiTrust repository. Integration with mPach requires that manuscripts that reach the "layout" stage in OJS be sent to Prepper, which prepares the HathiTrust Submission Information Package (SIP).

Prepper provides a user interface for the editor of a journal: a dashboard for administering the journal and putting manuscripts through a production process—akin to composition and typesetting—that prepares all content according to the preservation standard developed for mPach content in HathiTrust. Prepper invokes Norm, a Python application developed to convert manuscripts from Office Open XML ("DOCX") format [14] into XML that conforms to JATS. DOCX is the default option because, like OJS, it is widely used in the editorial process of journals published by libraries. The Prepper interface also guides the staff member through a review of validation errors detected by Norm's conversion, uploading high-resolution figures, supplying "alt text" for figures, previewing the article as rendered using the default stylesheet (based on the Preview XSLT stylesheets [15]), uploading supplementary material [16], and submitting for ingest into HathiTrust.

mPach requires a number of significant modifications to HathiTrust components and workflows

Figure 2: Mockup of an article viewed in HathiTrust's user interface

originally designed to support reformatted print materials. The reading interface in HathiTrust, which previously supported only rendering of digitized page images, renders JATS XML in HTML and allows a user to download a dynamically generated PDF and EPUB, display metadata specific to articles (figure 2), and link to a special "collection" for the journal in HathiTrust's Collections application [17] that allows for browsing volumes and issues of the journal (figure 3).

Discovery of known items in HathiTrust using metadata like title and author is currently provided for by a catalog of MARC records, with one per item in the repository. For mPach, each article has its own analytic catalog record, tied to a monographic record for the journal as a whole. Finally, the HathiTrust Data API [18] allows for the content of each article to be retrieved for use outside of the native HathiTrust interface.

Note that by policy HathiTrust only closes access to content for legal reasons, not because a rightsholder wants to restrict access. Therefore, mPach only supports

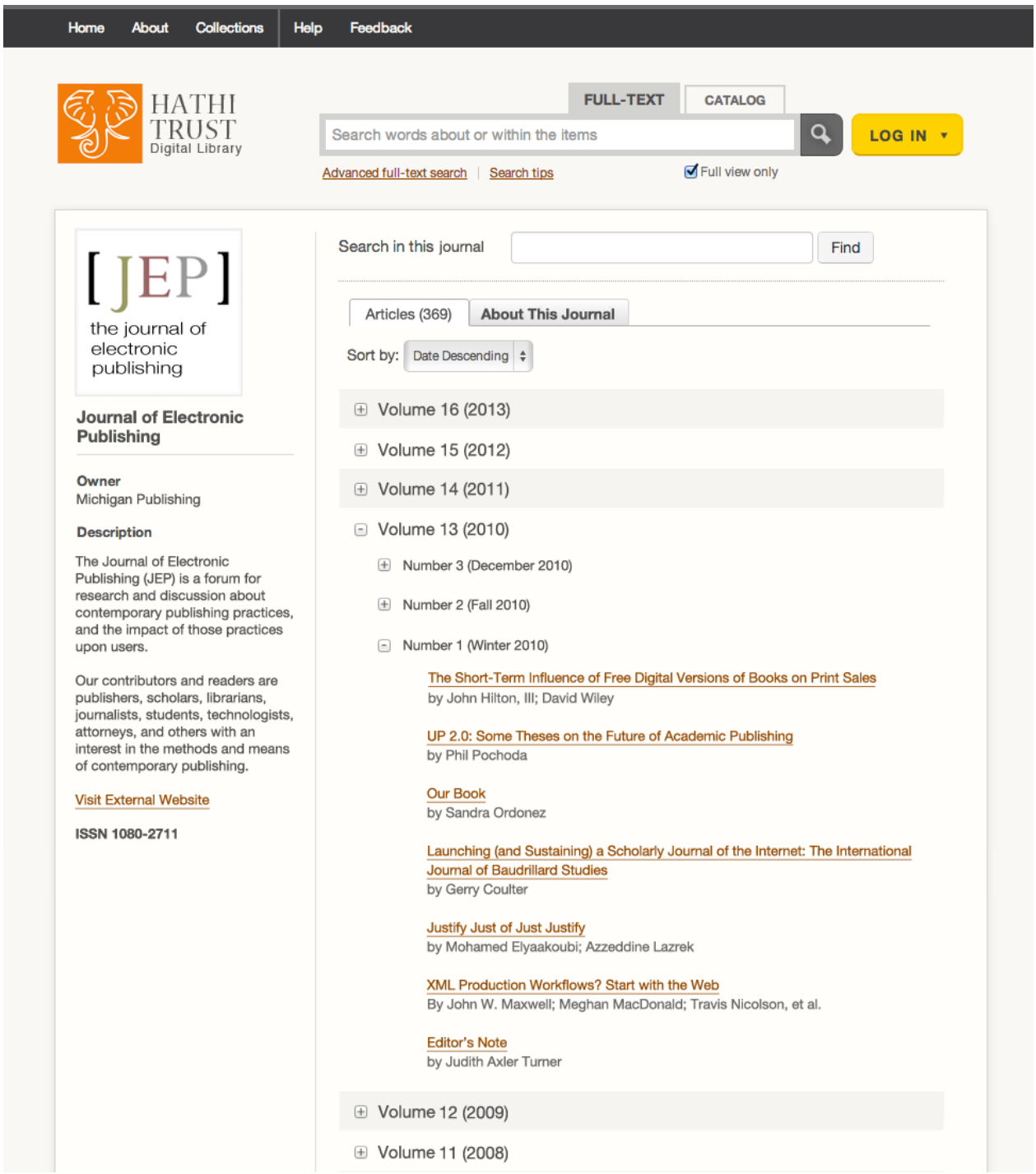


Figure 3: Mockup of a journal viewed in HathiTrust's user interface

the publishing of open-access journals.

4 Workflow

In the typical workflow for publishing a journal using mPach, a journal editor uses OJS to manage submissions, peer review, and the editing process. Once an article reaches the "layout" stage (where a combination of composition and typesetting allows the article to be formatted in a consistent way), the journal editor formats it according to a predefined list of styles in Microsoft Word and submits the article in DOCX to mPach's Prepper, which guides the editor through conversion to JATS XML, preparation of the SIP, and submission for ingest. Prepper keeps track of articles so that a revised version can be submitted for ingest. Currently the ingest process overwrites any previous version of an item with the same identifier, but eventually HathiTrust will archive past versions and allow users to navigate among them.

5 mPach as a shared infrastructure

The U-M Library plans to host the Prepper system,

including the submission module, to facilitate authorized deposit of content, and will make this system available for use by organizations wishing to publish journal literature in HathiTrust. The developers envision extending the Norm component to handle OpenDocument ("ODT") [19] and LaTeX as input formats, each of which is more commonly used in certain communities. Furthermore, if the Book Interchange Tag Suite [20] is adopted as a standard, the mPach architecture might be extended to support monograph publishing. While mPach is currently being developed to meet the needs of the U-M Library, the contribution of the sourcecode to the planned HathiTrust Development Environment should foster contributions from developers not at U-M and therefore lead to the creation of a truly shared infrastructure for publishing open-access scholarly journals.

References

[1] Sadie L. Honey. Preservation of electronic scholarly publishing: an analysis of three approaches. Portal: Libraries and the Academy, 5(1):59-75, Jan. 2005.

[2] NISO SERU Standing Committee, SERU: A Shared Electronic Resource Understanding: A Recommended Practice of the National Information Standards Organization. National Information Standards Organization (NISO), May 2012. http://www.niso.org/publications/rp/RP-7-2012_SERU.pdf.

[3] Lots of Copies Keeps Stuff Safe. http://www.lockss.org/.

[4] CLOCKSS. http://www.clockss.org/.

[5] Portico. http://www.portico.org/.

[6] National Library of the Netherlands and Elsevier Science make digital preservation history: permanent digital archive assures perpetual accessibility of scientific heritage. August 20, 2002. http://www.kb.nl/en/news/news-archive-2002/national-library-of-the-netherlands-and-elsevier-science-make-digital-preservation-history.

[7] James L. Mullins, Catherine Murray-Rust, Joyce L. Ogburn, Raym Crow, October Ivens, Allyson Mower, Daureen Nesdill, Mark Newton, Julie Speer, and Charles Watkinson. Library Publishing Services: Strategies for Success: Final Research Report. March 2012. http://wp.sparc.arl.org/lps/.

[8] Open Journal Systems. http://pkp.sfu.ca/ojs/.

[9] DSpace. http://www.dspace.org/.

[10] Digital Commons. http://digitalcommons.bepress.com/.

[11] HathiTrust Digital Library. http://www.hathitrust.org/.

[12] mPach. http://www.lib.umich.edu/mpach.

[13] Journal Article Tag Suite. http://jats.nlm.nih.gov/.

[14] Office Open XML. Wikipedia. http://en.wikipedia.org/wiki/Office_Open_XML.

[15] NISO Journal Article Tag Set (JATS) version 1.0: Preview XSLT stylesheets. https://github.com/NCBITools/JATSPreviewStylesheets.

[16] Recommended Practices for Online Supplemental Journal Article Materials: a recommended practice of the National Information Standards Organization and the National Federation of Advanced Information Services. January 2013. http://www.niso.org/publications/rp/rp-15-2013.

[17] Collections. HathiTrust Digital Library. http://babel.hathitrust.org/cgi/mb.

[18] HathiTrust Data API. http://www.hathitrust.org/data_api.

[19] OpenDocument. Wikipedia. http://en.wikipedia.org/wiki/OpenDocument.

[20] Book Interchange Tag Suite (BITS) 0.2 DRAFT. http://jats.nlm.nih.gov/extensions/bits/.

Метки времени для глобальной идентификации версий

© С. В. Знаменский
Институт программных систем имени А.К. Айламазяна РАН
Переславль-Залесский
svz@latex.pereslavl.ru

Аннотация

Децентрализованное управление резервированием, локальными перезапусками и обновлением исполняемого в распределённой компьютерной системе кода придаёт системе такие качества интернета как высокая доступность и катастрофоустойчивость. Система становится эклектичной, подверженной потере централизованного управления и непротиворечивости.

Однако только эклектичные системы дают возможность опереться на децентрализацию разработок для решения сложных междисциплинарных задач.

Описана проблема обеспечения доступности и согласованности данных в эклектичных компьютерных системах, решаемая идентификацией версий данных временем изменения.

Предлагаемая полиритмичная система автоматического версионирования объектов разделяемой памяти включает в себя способ компактного единообразного представления меток времени с разным разрешением и способ обеспечения согласованности информации на основе таких меток.

Введение

Для повышения надёжности и качества обслуживания мультипроцессорных компьютерных систем широко используется децентрализованное администрирование.

Без объявления технологических перерывов в обслуживании во многих системах проводятся локальные перезапуски подсистем, переключения на резервные сервисы, автоматические обновления исполняемого кода. Нередко используются резервные копии данных и откатываются автоматические обновления.

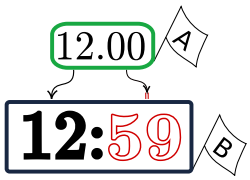


Рис. 1: Пример асинхронного сбоя (*устаревшие данные помечены красным контуром*)

Децентрализация управления потоками данных создаёт проблему системных нарушений порядка событий. Рисунок 1 поясняет её на примере простейшей сети из двух узлов, расположенных в удалении друг от друга, связанных двумя каналами.

Узел A выдаёт по одному каналу текущий час 0,1, ..., 23, 0, ..., а по другому - текущие минуты: 0... 59, 0,...

Узел B это часы, запрашивающие данные у A и выдающие время.

Мы будем считать, что все составляющие системы и каналы связи работают безупречно. Это не отменяет возможных задержек изменчивой продолжительности, неизбежных при передаче данных между узлами. Поэтому около полудня возможны показания 12.59 или 11.00 с ошибкой на час.

Пример иллюстрирует особый класс сбоев в компьютерных системах, связанный с нарушением синхронности процессов. Сбой такого рода способен разрушать связи родственных данных в сложных распределённых системах. В роли часов и минут могут оказаться любые синхронно изменяющиеся данные при возможности распространения по разным путям. Под угрозой разрушения оказываются не только связи данных наблюдений или их обработки, но и связи фрагментов исполняемого кода, метаданных и(или) исходного кода.

Возможность конфликтного сочетания независимых изменений данных может ускользать от внимания разработчиков. Неразумно усложнять систему ради пренебрежимо маловероятной ситуации.

Труды 15-й Всероссийской научной конференции «Электронные библиотеки: перспективные методы и технологии, электронные коллекции» — RCDL-2013, Ярославль, Россия, 14-17 октября 2013 г.

1 Угроза асинхронного сбоя

Определение. *Будем называть* асинхронным сбоем *(asynchronic failure) негативные последствия непредусмотренных задержек или схожих дефектов синхронизации данных.*

Асинхронный сбой выражается в потерях:

согласованности — когда информация различных версий объекта смешивается;

порядка — когда задержки передачи или локального восстановления непроизвольно сменяют новое значение старым;

данных по тайм-ауту при передачах или сложных расчётах.

В результате система может потерять адекватность, продуктивность или отзывчивость вплоть до непригодности.

1.1 Факторы риска

Пример вскрывает факторы асинхронного сбоя:

1. *Неполнота онтологии* системы, допускающей неучтённую связь между данными.
2. *Получение по разным каналам* неявно связанных данных извне системы.
3. *Разброс продолжительности* прохождения информации обозначим δ. Тогда вероятность асинхронного сбоя равна $\frac{\delta}{\tau}$, где τ это средняя продолжительность времени между связанными изменениями данных.

1.2 Характер угрозы

Особое коварство асинхронных сбоев состоит в их способности ускользать от тестирования и в невозможности воспроизведения.

В отличие от случайных сбоев их вероятность пренебрежимо мала. В штатном режиме задержки малы и стабильны, поэтому статистические оценки результатов тестирования асинхронных сбоев не замечают. В нештатных ситуациях временные локальные перезагрузки, потери, очереди, блокировки местами удлиняют задержки обработки или передачи данных в многие тысячи раз, создавая практически неповторимые сочетания локальных задержек, приводящие к асинхронным сбоям.

Диагностика и воспроизведение затруднены отсутствием достаточно полных сведений о временных локальных перегрузках.

В результате, спроектированная и испытанная по всем правилам система сбоит в нештатной ситуации так, что воспроизвести дефект в лабораторных условиях оказывается практически невозможным.

Такая скрытность асинхронных сбоев выводит их за рамки современной теории надёжности, разделяющей дефекты технической системы на случайные, вероятность которых можно оценить с помощью теории и предварительных испытаний, и систематические, выявляемые тщательным тестированием компонент и системы в целом. Получается, что достаточно сложная распределённая система, даже построенная по всем канонам теории надёжности на основе действующих стандартов, требует особых архитектурных решений чтобы не оказаться фатально подверженной асинхронным сбоям [1].

2 Подходы

Теория и практика разработки информационных систем настоятельно рекомендуют базироваться на стандартах [2], основанных на математической теории предикатов и надёжно проверенных широкой практикой создания систем вчерашнего и сегодняшнего дня. Рациональный подход к обеспечению качества информации имеет центральным стержнем

1. Тщательную централизованную проработку логической модели системы.
2. Чёткую организация разработки, точно следующей выверенной логической модели.

В каждый момент времени система имеет формально непротиворечивое в этой модели состояние, которое моментально изменяется на новое столь же логически целостное состояние.

Это означает, что любое событие непосредственно порождает транзакцию, как бы мгновенно вносящую все сопутствующие изменения во все части системы. Параллельно могут выполняться только такие транзакции, которые никак не взаимодействуют по данным. Результат должен быть таким, как если бы все транзакции выполнялись последовательно, что проблематично [3, 4]. Такая изоляция строгих транзакций имеет целью создание для программиста иллюзии (view serializability), что он работает с системой один, чтобы избавить его от сложностей конкурентного взаимодействия. Во многих практических задачах такая иллюзия допустима и рациональный подход оказывается вне конкуренции.

2.1 Ограниченность рационального подхода

Ряд востребованных в ближайшем будущем приложений страдает от принципиальных ограничений рационального подхода:

- Единый лог транзакций [5] **ограничивает масштабируемость**.
- Для системы, которая по сути должна базироваться на неустойчивой сети (например, межпланетные или армейские

системы) **централизованная обработка данных** неприемлема: временно теряющие связь части очевидно должны продолжать функционировать корректно и по возможности эффективно. Распространённое мнение о том, что известная теорема CAP Брювера [6] противоречит возможности создания таких систем, ошибочно [7, 8].

- Гибкость, адаптивность и другие полезные свойства в полной мере реализуемы лишь многими сильными независимыми командами разработчиков при демократическом взаимодействии. **Жёсткий упор на централизацию разработки и сопровождения** резко ограничивает допустимую сложность системы одной головой генерального конструктора [9].
- **Неизменность модели данных** не свойственна долгоживущим системам [10].
- **Монополия централизованной разработки** ограничивает пригодность систем к независимой локальной модификации (информационная система крупной корпорации, межгосударственная система взаимной безопасности, интеллектуальная самовосстанавливающаяся [11] или эволюционирующая [12] системы).

Любое отклонение от рационального подхода обостряет проблему изучения и предупреждения асинхронных сбоев.

2.2 Борьба с задержками

Последний фактор риска — неравномерность задержек — уменьшается такими мерами как

- Использование новейших технологий ускорения обмена и обработки данных.
- Упреждающая балансировка ресурсов, снижающая риски перегрузок [13].
- Поддержание запаса системных ресурсов, обеспечивающего устойчивость к перегрузкам.

Эти меры снижают риск возможно в сотни раз кроме систем, которые (само)восстанавливаются или реконструируются в ходе эксплуатации: восстановление всегда означает существенную неискоренимую задержку.

Если возможно гарантировано ограничить время задержки малой величиной, то риск асинхронных сбоев можно полностью исключён гистерезисной фильтрацией неустойчивых изменений. Для этого результаты каждого запроса должны сверяться с предыдущим показанием. Если оно изменилось, то надо скажем каждую секунду или с другим периодом Δ запрашивать время снова и пока полученное значение не повторится, считать его неустойчивым и не показывать. Если разница в задержках δ не превысит Δ, то такой фильтр исключит ошибку.

Для полного подавления ошибок требуется $\Delta \gg \delta$. При этом часы будут всегда отставать не менее чем на Δ. Уменьшение Δ повысит риск ошибки часов, увеличение усилит их отставание. Если же ускорение достигнуто оптимизацией в штатных ситуациях, то оно вправе не сработать в особом случае, задержка превысит гистерезисную и сбой произойдёт.

Таким образом, гистерезисная фильтрация полезна лишь на входе системы.

Остаётся путь версионирования данных эклектичной системы.

3 Эклектичность

Эклектичность модельного примера лишает эффективности верифицированные протоколы обмена и обработки информации и строгих транзакций.

3.1 Апология эклектичности

Популярная идея единой универсальной онтологии, на которой базируются стандарты и теория разработки информационных систем, проекты Semantic Web, Web 2.0 и Web 3.0, к сожалению принципиально не реализуема. Теорема Гёделя о неполноте подсказывает, что полная непротиворечивая онтология возможна лишь для примитивных систем.

Чтобы показать, что онтологии изменчивы в любой практически значимой области, рассмотрим символ астрономической точности — измерение времени.

В сутках 24 часа. Однако в системе, учитывающей переход на летнее поясное время, всё гораздо сложнее и количество часов в сутках иногда оказываться 23 или 25, что зависит от административного подчинения. С секундами в минуте сложнее. В последнюю минуту полугодия иногда (но почти каждый год) по итогам астрономических наблюдений вводится дополнительная високосная 61-я секунда. Никто не знает, будет ли последняя секунда следующего года високосной. Более того, Международный астрономический союз всерьёз обсуждает возможности отмены високосных секунд в частности с заменой их високосными часами.

Даже в математике общеизвестные факты уточняются с новых позиций. Пифагорейцы хранили в тайне иррациональность корня из двух поскольку это противоречило понятию числа как отношения. Ещё до рождения Христова было известно, что сумма углов в треугольнике равна развёрнутому, а квадрат любого числа положителен. Всё это неверно в геометриях

Лобачевского и Пуанкаре и алгебре комплексных чисел. В учебниках по математическому анализу написано, что разрывные функции производных не имеют. Но теория обобщённых функций изучает их производные.

В предметных областях, отличных от математики и астрономии, уточнение онтологии ещё быстрее выводит на передний край науки и научно-технической политики и становится зыбким и неустойчивым. В частности, это верно и для сетевых технологий [14].

Безупречность онтологии оказывается жёстко ограниченной во времени и привязанной к конкретной сфере приложений.

Правильное понимание недолговечно

Но именно правильное понимание является основой стандартов разработки компьютерных систем. Вывод парадоксален и неутешителен: *императивные компьютерные системы в обозримой перспективе обречены продолжать разочаровывать пользователей* [15].

Выход подсказывает общепризнанные принципы разделения труда и разделения зон внимания (separation of concerns). Будущее за надёжными согласованно развивающимися взаимопроникающими системами, улучшаемыми независимыми группами, компетентными в своих областях [16–18]. Такие системы правильно называть эклектичными. Интернет в целом — это пример эклектичной системы.

Негативное отношение, закрепившееся за термином и системами, исходит от наивной мечты об абсолютном знании и отсутствия теории, технологий и примеров согласованных эклектичных систем.

Не исключено, что технологии эклектичных систем откроют путь к эффективной децентрализованной разработке на фоне постоянно доступного качественного сервиса.

Подобно тому, как объединив творческие силы мира Википедия превзошла по широте, полноте, доступности, актуальности и популярности все лучшие энциклопедии мира, совместное творчество независимых групп разработчиков [21] способно привести к результату, превосходящему лучшие ожидания [22]. Речь идёт о качественно превосходной точности, надёжности и дружелюбности взаимодействующих компьютерных систем, о самовосстанавливающихся, эволюционирующих и мультиагентных системах, о качественном скачке функциональности, надёжности и удобства пользовательских интерфейсов.

3.2 Модель эклектичной системы

Эклектичные системы не похожи на конечные автоматы: их состояние меняется не мгновенно. Это ближе к реальности распределённых систем, в которых допускаются переключение на резервный сервис, перезапуск локального сервиса, восстановление из резервной копии и иные внезапные приостановки активности. Для эклектичных систем (ЭС) более адекватной представляется модель, основанная на идеях контекстной автономности из [19] и [20]:

- ЭС может состоять из подсистем, являющихся ЭС, и входить в другие ЭС.
- ЭС может считывать информацию датчиков, пользовательских интерфейсов и иных источников.
- ЭС предоставляет доступ
  - (1) к актуальной на указанный момент версии каждого выработанного ею информационного объекта,
  - (2) к описаниям белых пятен истории, т.е. промежутков времени переходов к согласованным состояниям в указанном контексте данных.

Проблема распределения ресурсов между направлениями и темпами обновления и поддержкой функционирующих сервисов сложна и многопланова, но делится на очевидные части:

1. Уточнение стратегии развития.
2. Определение приоритетов внутренних процессов.
3. Организация обработки информации в соответствии с установленными приоритетами.

Рассмотрение первых двух частей выходит за рамки настоящей статьи. Её задача описать систему приоретизированной обработки информации, полностью исключающую асинхронные сбои.

3.3 Ретроспективность и обновления

Любые данные поступают в эклектичную систему с указанием времени. Полезны протоколы своевременного получения информации об изменениях (comet).

Если в источнике есть информация о времени актуальности данных, то она должна быть корректно использована. Иначе идентификаторы версий могут быть сгенерированы на основе текущего времени.

Вывод данных из эклектической системы по запросу может в основном осуществляться двумя способами:

Проспективный означает ожидание когда появится версия, актуальная на момент запроса или более поздний.

Ретроспективный [19] означает немедленную выдачу последней на момент запроса версии данных.

Проспективный вариант привычно включает ожидание завершения обработки всех внесённых на момент запроса данных. Ретроспективный непривычен тем, что исключает ожидания обработки. Это напоминает чтение данных их кэша или реакцию поисковых серверов интернет.

Механизм автоматического версионирования заменяет иллюзию изоляции необходимостью корректной идентификации версий объектов. Происходившее в системе становится доступным не через лог, а через машину времени, показывающую входные и выходные данные в динамике обработки. Обработка может программироваться на любых языках программирования, а её корректность диагностироваться и отлаживаться средствами этих языков. Требуется лишь чтобы выходные данные правильно соответствовали входным.

Гладкое обновление естественно будет происходить поэтапно:

- подготовка и отладка новой версии исполняемого кода системы,
- запуск его в параллель с работающим с автоматическим извещениями о расхождениях в выходных данных,
- включение в качестве бета-версии с возможностью мгновенного переключения между версиями,
- бета-тестирование потребителями,
- получение статуса базовой версии,
- получение статуса резервной версии,
- отключение по причине длительной невостребованности.

От программистов потребуется не только тестирование своей системы, но и сравнительное тестирование версий систем-поставщиков входной информации.

Организация исполнения в эклектичных системах может основываться на следующей схеме:

- Автоматическое версионирование закладывается на низшем уровне, а к нему адаптируются структуры данных и интерфейсы.
- Использование штампа времени в качестве метки версии обеспечивает синхронность данных разной природы и происхождения.
- При обмене информацией между узлами передаются фрагменты истории изменений, а при обработке совместно обрабатываются или показываются данные, относящиеся только к одной версии.

На пути реализации этой идеи лежат многочисленные подводные камни.

4 Трудности автоматического версионирования

4.1 Одновременность

Первая проблема состоит в определении того, какие данные считать одновременно актуальными, то есть относящимися к одному состоянию системы.

Классические исследования показали запредельную техническую сложность синхронизации всех изменений в распределённой компьютерной системе. Система рассматривалась при этом как набор конечных автоматов, обменивающихся сообщениями о событиях. Время в ней трактовалось как вспомогательное средство для выстраивания всех событий в единый линейный порядок.

В интересующей нас эклектичной модели время это показания локальных часов, то есть физическая величина, измеренная с некоторой точностью.

Выделить актуальные в общий момент времени данные далеко не просто по многим причинам.

Во-первых, окончание актуальности данного X может быть настолько близко к началу актуальности данного Y, что момент их одновременной актуальности существует лишь с некоторой вероятностью.

Во-вторых, погрешность указания времени Δ может варьироваться от наносекунд для данных быстро протекающих процессов до тысячелетий для археологических данных.

В-третьих, конкретная разница во времени событий может свидетельствовать об одновременности или о неодновременности событий в зависимости от постановки задачи.

В-четвёртых, различаются актуальность в некоторый момент промежутка и актуальность в течение всего промежутка.

В-пятых, проблематично корректно совместить данные, относящиеся к приближенно известной общей границе промежутков шкалы времени с одним из этих промежутков.

В-шестых, случается, что актуальная достоверная информация недоступна и пользователям порой нужен доступ к новейшим неполным или непроверенным данным. В таких ситуациях

разумно предоставлять особый интерфейс к неполной сводной информации.

Нужен прозрачный способ безупречно автоматически выделять одновременно актуальные версии данных. Идентификация моментами времени должна

- быть достаточно универсальной,
- отражать значение и погрешность промежутка актуальности в широких диапазонах,
- позволять быстро и просто собирать последние одновременные данные.

Организация исполнения должна до предела снижать вероятность ситуации, в которой обработка задерживается из-за неспособности системы выделить одновременно актуальные данные.

4.2 Эффективность и алгоритмы

Работа эклектичной системы должна опираться на эффективные алгоритмы решения принципиально новых, ранее не рассматривавшихся задач, таких как

- поддержание и сохранение истории датированных изменений,
- обмен фрагментами истории изменений в семантике глобальной разделяемой памяти,
- отбор согласованных версий входных данных для совместной выдачи или обработки,
- эволюция структур данных во времени.

В последнем пункте соединяются два направления поиска. Во-первых, поддержка истории изменений структур традиционно ведётся без привязки ко времени. Обычная структура в памяти либо не хранит историю изменений объекта, либо хранит фиксированное количество последних изменений, либо (persistent data) хранит полностью, но только порядок событий без привязки ко времени. Последнее практически означает экспоненциальный рост требуемых ресурсов памяти.

Привязка ко времени позволила бы без дублирования информации постоянно поддерживать достаточное число снимков данных (snapshots), синхронно забывая данные промежуточных изменений [23].

Например, можно сохранить все ежечасные снимки прошлого месяца, все ежедневные прошлого года, чтобы существенно сэкономить ресурсы памяти, оставляя доступной для разнообразной обработки (такой как машина времени [24, 26–28]) значимую часть истории.

Во-вторых, это поддержка эволюции структур данных. Для иерархической структуры (скажем, дерево XML-документа) это прежде всего появление промежуточного уровня иерархии. Скажем, была классификация данных по городам России, а понадобилось ввести административные округа и распределить данные городов по ним.

Сейчас такая структурная перестройка требует удаления всех веток изменяющегося уровня и добавления их к добавленным узлам промежуточного уровня. Эта операция нарушает связи между элементами структуры, портит предысторию веток и тем самым делает невозможным гладкое (без перерывов в обслуживании) обновление. Для представления данных, которые должны быть непосредственно доступны с длительной историей структурных изменений, необходима дальнейшая проработка универсальных и эффективных гибких структур данных, поддерживающих эволюцию, см., например, [25].

4.3 Продолжительные изменения и согласованность

Пока обработка использует версии данных, актуальные на общий момент времени, результат однозначно определяется этими данными и идентифицируется тем же временем, угрозы рассогласования данных не возникнет. Поэтому сколько бы ни длилась такая обработка, её результат относится к той же версии данных и должен быть помечен тем же временем, чтобы не нарушить логический порядок.

Угроза рассогласование станет явной только если обработка соединит данные, не актуальные на общий момент времени. Это может случиться и для изменений, помеченных общим временем если одно из данных снова изменилось, а допустимое отставание локальных часов отнесло это событие к следующему моменту.

Если при алгоритмической обработке есть надежда избежать всего этого, то в пользовательских интерфейсах проблема остаётся. Как бы тщательно ни были отфильтрованы для пользователя согласованные данные, привычное использование дополнительных источников актуальной информации может сделать результат неадекватным.

Это придётся учитывать при создании пользовательских интерфейсов. Недостаточно просто принимать все меры к тому, чтобы информация в интерфейсе для пользователя была предельно актуальной и полной. Важно правильно идентифицировать результат. Идентификатор должен остаться прежним лишь в случае, когда пользователь действовал по ранее установленному алгоритму и не использовал сторонней более поздней информации. Иначе при сохранении идентификатор должен корректно отразить возникший разброс времени и вероятен конфликт для разрешения которого потребуются адекватные инструменты.

5 Система и перспективы

5.1 Способ кодирования времени

Шкалой S назовём разбиение числовой прямой на полусегменты равной длины, называемой шагом дискретизации. Шаг дискретизации определяется прикладными задачами и должен быть практически несущественным. Например, для полученных из веб-формы данных не имеет практического значения, нажал ли пользователь кнопку отправки десятой долей секунды раньше или позже.

Правый конец в полусегмент не входит, этим из предыстории исключаются события, на обработку которых не было времени. Базовый пример даёт разбиение на целочисленные промежутки $[n, n+1), n \in \mathbb{Z}$.

Мы будем использовать глобальные отметки времени (Новый год, полночь и т.д.) как разделители. Поскольку потребуется отчёт за год, то каждое событие должно определённо относится либо к прошлому году, либо к будущему. Сложность с событием, произошедшим в момент такого разделителя: допустимая погрешность часов позволяет отнести его в одних подсистемах к предшествующему разделителю полусегменту, а в других к последующему за разделителем.

В таких случаях логически неправомерно производить обработку данных на момент конца года и нужно выбирать момент, на который череда изменений приостановилась. Пользователя однако может интересовать состояние на конец года. По-видимому, лучшее, что может сделать система в этой ситуации, это выдать корректный ответ на близкий момент времени. В ситуации продолжительной высокой интенсивности конфликтующих изменений входных данных, в которой пользователя интересует не только (вероятно отдалённый) момент истины, но заведомо логически небезупречная оперативная оценка текущего состояния дел.

Для получения такой оценки можно игнорировать несущественные расхождения во времени, но при записи результата указать оценку дефекта (времени рассогласованности) τ. Эта оценка должна быть использована при выдаче пользователю предупреждения о размере возможной некорректности данных.

Одновременное использование многих шкал нуждается в их согласованности, означающей что из двух пересекающихся полусегментов один обязательно содержит другой. Обозначим $\mathbb{B} = \{k \cdot 2^l | k, l \in \mathbb{Z}\}$ множество всех двоичных дробей и назовём *бинарным семейством* такое семейство шкал, все сегменты которого имеют вид $[t-\delta, t) = [k \cdot 2^l, (k+1) \cdot 2^l)$, где $k, l \in \mathbb{Z}$.

Предложение 1. *Полусегмент бинарного семейства однозначно определяется своей серединой. Множество серединок таких полусегментов совпадает с* $\mathbb{B}$

Простота идентификации полусегментов бинарного семейства делает их привлекательными для использования в качестве идентификаторов моментов событий с учётом погрешности определения времени. Например, момент события, предшествовавшего моменту 2 с точностью примерно 0.1%, идентифицируется как $1.11111111111_{(2)}$.

Для ускорения поиска нужной версии в бинарном дереве важно чтобы идентификаторы были лексикографически упорядочены по актуальности. Несмотря на простоту и естественность, порядок при таком представлении не согласуется с желаемым лексикографическим: правый конец полусегмента t существенно более значим, чем левый $t - \delta$.

К сожалению, количество дней в году и другие соотношения мер времени не являются степенями двойки (и, к тому же, не все являются заранее фиксированными числами). Моменты времени практически задаются конечными наборами $t = (Y, M, D, h, m, s, \mu_0, \mu_1, ...)$ из указаний года Y, месяца M, дня D, часа h, минут m, секунд s, миллисекунд μ_0, микросекунд μ_1 и т.д.

Предложение 2. *Функция*

$$\begin{aligned} x(t) = {} & 2^{-3}Y + 2^{-7}(M-1) + 2^{-12}(D-1) \\ & + 2^{-18}h + 2^{-24}m + 2^{-30}s \\ & + \sum_{k \geqslant 0} 2^{-30-10k}\mu_k \end{aligned}$$

монотонно и однозначно представляет моменты времени двоичными дробями.

Замечание 1. *Функция теоремы 2 переводит привычные шкалы времени (годы, месяцы, дни, недели, часы и т. д) в шкалы бинарного семейства.*

Например, 13:35 30 июня 2014 года с секундной точностью идентифицируется двоичной дробью

$$\begin{aligned} x = {} & 2014 \cdot 2^{-3} + 29 \cdot 2^{-7} + 13 \cdot 2^{-12} \\ & + 35 \cdot 2^{-18} + 0.5 \cdot 2^{-24} + 2^{-31} \\ = {} & 11111011.110\,0101\,11101\,01101\,100010\,111111\,1_{(2)} \\ = {} & 2^{11} - 100.001\,1010\,00010\,10010\,011101\,000000\,1_{(2)}. \end{aligned}$$

Следующее утверждение описывает ускоряющий сравнение переход от чисел произвольной длины со знаком к битовым массивам (строкам произвольной длины).

Предложение 3. *Кусочно-линейная функция*

$$B(x) = \frac{1}{2} + \frac{\operatorname{sgn}(x)}{2}\left(1 - \frac{3}{2^{k(x)+1}} - \frac{1+|x|}{2^{2k(x)}}\right), \text{ где}$$
$$k(x) = \operatorname{ceil}\left(\log_2(1+|x|)\right),$$

непрерывна, монотонна и взаимно-однозначно отображает множество $\mathbb{B}$ *всех двоичных дробей на* $(0,1)\cap\mathbb{B}$.

Для промежутка с 1984 по 2112 годы это выражение удобно масштабируется $B(x(t)-2^{11})$ и упрощается до $B(x(t)-2^{11}=\frac{1}{2}+\frac{x}{(}t)2$ потому что $|x-1|<1$. Для длительности δ удобнее суточный масштаб времени $2B(-2^{1}2x(\delta))-1$, а для дефекта τ минутный $2B(-2^{2}4x(\delta))-1$.

Замечание 2. *Обратная к* $y=B(x)$ *функция вычисляется по формуле*

$$x = 1+\operatorname{sgn}(2y-1)\left(3\alpha-2\alpha^2|2y-1|-1\right),\ \textit{где}$$
$$\alpha(y)=2^{k(y)-1},$$
$$\tilde{k}(y)=-\operatorname{floor}\left(\log_2\left(1-|2y-1|\right)\right).$$

Замечание 3. *Количество значащих цифр после запятой в двоичной записи* $B(x)$ *равно сумме* $|k(x)|$ *и количества цифр в записи двоичной дроби* $|x|+1$, *взятого без заключительных нулей.*

В частности, образ $B(x(t)-2^{11})$ 34-значной двоичной дроби из последнего примера длиннее записи самой дроби на два знака поскольку для неё $k(x)=2$.

Для полусегментов $[t-\delta,t)$ и дефектов τ требуется монотонно закодировать лексикографически упорядоченные вектора из записи двоичных дробей $B(x(t)),B(2^{12}x(-\delta))B(2^{24}x(-\tau)),)$.

Замечание 4. *Можно монотонно закодировать лексикографически упорядоченные вектора из записи двоичных дробей* $(x_1,x_2,x_3)\in[0,1]$ *строками из* $\sum_{i=1}^{3}\operatorname{ceil}\left(\frac{\operatorname{length}(x_i)}{6}\right)+2$ *байт, принимающих 65 различных значений.*

Для этого битовый массив разбивается на группы по 6 бит, последняя дополняется нулями, каждая группа бит кодируется байтом (Base64) и x_i представляется строкой (в нашем примере длины 6 байт). Пусть «!» - байт с меньшим кодом, чем использованные в Base64. Тогда конкатенация $x_1.\text{«!»}.x_2.\text{«!»}.x_3$ даёт эффективную кодировку для замечания 4.

История изменений информационного объекта предстаёт упорядоченным по актуальности списком версий. Представление этого списка в виде структуры $B^{+}Tree$ [31, 32] обеспечивает быстрый доступ к актуальной на любой заданный момент версии. Версии с недостаточно высокими значениями при этом быстро пропускаются.

5.2 Способ организации исполнения

Функционирование эклектичной системы распадается на процессы обработки данных. Это как прикладные сервисные процессы, так и

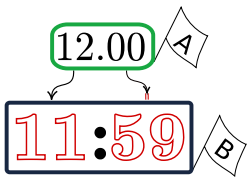


Рис. 2: Устойчивость системы к асинхронным сбоям (*устаревшие данные помечены контуром*)

системные, обеспечивающие сопровождение согласования стратегии развития, планирования и реализации обновлений, мониторинга состояния системы и уточнения приоритетов исполнения. Системная архитектура должна незаметно для прикладных программистов обеспечивать, что старый согласованный результат останется в действии и будет обновлён лишь когда полностью соберётся требуемая для согласованного обновления информация, см. рисунок 2.

Процессу доступны упорядоченные списки версий каждого объекта входных данных для обработки. Системный планировщик определяет идентификатор версии объектов результата обработки и обеспечивает обработчику быстрый доступ к версиям объектов, которые должны быть включены в обработку. Прикладная программа-обработчик (бинарный код или скрипт на любом языке программирования) обрабатывает пакет изменившихся входных данных. Пакетная обработка известна как средство резкого ускорения обработки данных [30].

Запуск очередной обработки данных осуществляется системным планировщиком в установленном для процесса ритме (аналогично [29]). Если предыдущая обработка этого процесса не завершилась, то запуск отменяется.

Результат может быть сохранён с версией (t,δ,τ) если версия любого использованного при обработке объекта o удовлетворяет условию, зависящему от положительности τ.

$\tau>0,\ t-\delta\le t_o-\delta<t_o+\tau_o\le t+\tau$;

$\tau=0,\ t-\delta\le t_o-\delta<t_o=t_o+\tau_o\le t$ и некоторое время после t_o; объект не менялся

При запуске обработки правый конец отрезка-идентификатора результата выбирается на шкале с ритмом запуска обработки так, чтобы оказаться на шкале с наибольшим шагом. Идентификатор всегда должен быть больше предыдущего сохранения, но меньше текущего момента.

Длина полусегмента δ изначально берётся вдвое меньше, чем было в предыдущей сохранённой версии результата этого процесса и $\tau=0$. Если для такого результата не хватает актуальных входных данных, то δ увеличивается вдвое, если не помогает, то ещё вдвое, а если и это не помогает и есть другие варианты

выбора для t, то проверяются они и если версия не выбирается и есть давно необработанные входные данные то появляется $\tau > 0$,

Последнее обеспечивает мягкую деградацию качества сервисов при перегрузках и аномальных задержках (передачи и исполнения) и незамедлительное полное восстановление качества сервисов при исчезновении перегрузок и задержек.

6 Выводы

1. Эклектичные компьютерные системы обещают качественное превосходство по важнейшим показателям.
2. Конфликты версий неизбежны внутри сложных эклектичных систем.
3. Автоматическое версионирование на основе меток времени открывает путь к исключению таких конфликтов.
4. В начале этого пути теоретическая проработка и создание новых алгоритмов и структур данных.
5. Описаны способ маркировки моментов событий и способ корректной организации исполнения в эклектичных системах.

Список литературы

[1] M. Ghafari, P. Jamshidi, S. Shahbazi, H. Haghighi. An architectural approach to ensure globally consistent dynamic reconfiguration of component-based systems // Proceedings of the 15th International ACM SIGSOFT Symposium on Component-based Software Engineering (CBSE'2012). – Bertinoro, Italy, June 2012.

[2] J. Kasser, D.K. Hitchins. Unifying systems engineering: Seven principles for systems engineered solution systems // The 20th International Symposium of the INCOSE. – Denver, 2011. – P. 1–11.

[3] H. Berenson, P. Bernstein, J. Gray, J. Melton, E. O'Neil, P. O'Neil. A Critique of ANSI SQL Isolation Levels // Proc. of the ACM SIGMOD International Conference on Management of Data. – 1995. – P. 1–10.

[4] R. Normann, L.T. Østby. A theoretical study of 'snapshot isolation'// ICDT. – 2010.– P. 44–49.

[5] С.Д. Кузнецов. Транзакционные параллельные СУБД: новая волна // Труды Института системного программирования РАН. – 2011. — Т. 20 – http://cyberleninka.ru/article/n/tranzaktsionnye-parallelnye-subd-novaya-volna.

[6] S. Gilbert, N.A. Lynch. Perspectives on the CAP Theorem // IEEE Computer Society. – 2012.– P. 30–36.

[7] K.P. Birman et al. Overcoming CAP with consistent soft-state replication // Computer. — 2012. – Т. 45. – № 2. — С. 50–58.

[8] С.В. Знаменский. Ретроспективная основа распределённой памяти для изменчивой вычислительной среды // Материалы VI Международной конференции «Параллельные вычисления и задачи управления» (PACO'2012). – М.: ИПУ РАН, 2012. – Т. 2. – С. 259–272.

[9] I. Sommerville, D. Cliff, R. Calinescu, J. Keen, T. Kelly, M. Kwiatkowska, J. McDermid, R. Paige. Large-scale Complex IT Systems // Communications of the ACM. – 2012. – V. 55, Issue 7. – P. 71–77.

[10] V. Andrikopoulos, S. Benbernou, M.P. Papazoglou. On the evolution of services // IEEE Transactions on Software Engineering. – 2012. – V. 38, Issue 3. – P. 609–628.

[11] E. Hollnagel. From protection to resilience: Changing views on how to achieve safety // 8th International Symposium of the Australian Aviation Psychology Association. – Sydney, Australia, 2008.

[12] I. Fehérvári, W. Elmenreich. Evolutionary methods in self-organizing system design // Proceedings of the 2009 International Conference on Genetic and Evolutionary Methods. — 2009. – P. 10–15.

[13] T. Chalermarrewong, S. See, T. Achalakul. Parameter Prediction in Fault Management Framework // Proceedings of The International Symposium on Grids and Clouds (ICGC 2012). – 26 February–2 March 2012, Taipei, Taiwan. – http://pos.sissa.it/archive/conferences/153/005/ISGC%202012_005.pdf – 2012. — Т. 1. – P. 5.

[14] C.M. Kelty. Conceiving Open Systems // Wash. UJL & Pol'y. – 2009. – Т. 30. – P. 139.

[15] Y. Merali, T. Papadopoulos, T. Nadkarni. Information systems strategy: Past, present, future? // J. Strateg. Inform. Syst. – 2012. – http://dx.doi.org/10.1016/j.jsis.2012.04.002

[16] P. Feiler, R.P. Gabriel, J. Goodenough, R. Linger, T. Longstaff, R. Kazman, M. Klein, L. Northrop, D. Schmidt, K. Sullivan, K. Wallnau. Ultra-Large-Scale Systems: The Software Challenge of the Future // Technical Report. – Carnegie Mellon University Software Engineering Institute. – 2006.

[17] L. M. Northrop. Ultra-Large-Scale Systems: Scale Changes Everything // SMART Ultra-Large-Scale Systems Forum. – March 6, 2008.

[18] C. Ncube. On the Engineering of Systems of Systems: key challenges for the requirements engineering community // Requirements Engineering for Systems, Services and Systems-of-Systems (RESS). – Aug. 2011. – P. 70–73.

[19] С. В. Знаменский. Ретроспективная основа совместной реорганизации сложных информационных ресурсов // Электронные библиотеки: перспективные методы и технологии, электронные коллекции». – RCDL-2011. – Воронеж, Воронежский госуниверситет, 2011. – С. 93–101. – http://ceur-ws.org/Vol-803/paper10.pdf

[20] P. Helland, D. Haderle. Engagements: Building Eventually ACiD Business Transactions // 6th Biennial Conference on Innovative Data Systems Research (CIDR '13). – January 6–9, 2013. – 12 pp.

[21] S. R. Jeffery, L. Sun, M. DeLand, N. Pendar, R. Barber, A. Galdi. Arnold: Declarative Crowd-Machine Data Integration // 6th Biennial Conference on Innovative Data Systems Research (CIDR '13). – January 6-9, 2013. – 8 pp.

[22] W. Van Osch, M. Avital. Collective Generativity: The Emergence of IT-Induced Mass Innovation // Proceedings of JAIS Theory Development Workshop. – http://sprouts.aisnet.org/9-54.

[23] J. Stender, M. Hogqvist, B. Kolbeck. Loosely time-synchronized snapshots in object-based file systems // Performance Computing and Communications Conference (IPCCC). – IEEE 29th International. – IEEE, 2010. — P. 188–197.

[24] L. Shrira, C. van Ingen, R. Shaull. Time travel in the virtualized past: Cheap fares and first class seats // Haifa Systems and Storage Conference. – SYSTOR 2007.

[25] Yu. Rogozov, A. Sviridov, S. Kucherov. Meta-Database for the Information Systems Development Platform // 6th Spring/Summer Young Researchers' Colloquium on Software Engineering (SYRCoSE 2012), – P. 164–171.

[26] P. Ta-Shma, G. Laden, M. Ben-Yehuda. Factor: Virtual machine time travel using continuous data protection and checkpointing // Operating Systems Review. – 2008. – V. 42, Issue 1. – P. 127–134.

[27] Ragib Hasan. Trustworthy History and Provenance for Files and Databases // PhD thesis, University of Illinois at Urbana-Champaign. – Urbana, Illinois, October 2009.

[28] G. Fourny, D. Florescu, D. Kossmann. A time machine for XML // Technical report № 734. – ETH Zurich, Switzerland, 2011.

[29] J. Wang, K. Y. Lam, S. Han, S. H. Son, A. K. Mok. On Co-Scheduling of Periodic Update and Application Transactions with Fixed Priority Assignment for Real-Time Monitoring // Advanced Information Networking and Applications (AINA). – 2012 IEEE 26th International Conference. – 2012, March. – P. 253–260.

[30] A. Thomson, T. Diamond, S. C. Weng, K. Ren, P. Shao, D. J. Abadi. Calvin: fast distributed transactions for partitioned database systems // Proceedings of the 2012 international conference on Management of Data. – 2012, May. – P. 1–12.

[31] Г. М. Адельсон-Вельский, Е. М. Ландис. Один алгоритм организации информации // Доклады АН СССР. — 1962. — Т. 146, № 2. -- С. 263–266.

[32] L. Jiang, B. Salzberg, D. Lomet, M. Barrena. The BT-Tree: A Branched and Temporal Access Method // VLDB'00 Proceedings. – 2000. – P. 451–460.

Timestamps for a global version identification

Sergej Znamenskii

Decentralized management of data restore, local restarts and code updates for a distributed computer system gives the system some of such the Internet's qualities as high availability and disaster recovery. The system becomes eclectic, prone to loss of centralized control and consistency.

However, only the eclectic systems make it possible to rely on the decentralization of development to solve complex interdisciplinary problems

The article describes the problem of ensuring the data consistency and availability in eclectic computer systems to be solved with timestamp-based versioning.

Proposed polyrythmic system of automatic versioning of shared memory objects includes short uniform presentation of timestamps with different resolution and a timestamps-based way to ensure consistency.

Вероятностные модели и методы оценки качества эталонных массивов текстов при классификации

© В. Г. Васильев
ООО «ЛАН-ПРОЕКТ»,
г. Москва
vvg_2000@mail.ru

Аннотация

В работе рассматриваются вероятностные модели ошибок экспертов при формировании эталонных массивов текстов, а также методы их вычисления. В рамках данных моделей находятся взаимосвязи между истинными и наблюдаемыми показателями качества, определяются размеры тестовых выборок и максимальные значения показателей качества. Приводятся примеры вычисления ошибок на материалах дорожек РОМИП.

1 Введение

При практическом построении средств автоматической классификации возникает большое количество различных проблем, связанных сложностью и недостатками исходных данных, ограниченностью существующих методов классификации и др. [7],[10],[12],[15].

При оценке качества классификации обычно производится сравнение результатов автоматической классификации с результатами ручной классификации, выполненной экспертами. При этом предполагается, что в эталонной ручной классификации ошибки отсутствуют. Однако на практике эксперты при оценивании документов также совершают ошибки, которые могут быть вызваны различными причинами: невнимательностью, случайными опечатками, неоднозначностью наименования рубрик, низкой квалификацией экспертов в рассматриваемой предметной области, большим количеством рубрик и др. В результате получаемые оценки качества являются искаженными и даже для полностью правильной классификации показатели качества могут отличаться от своих максимальных значений.

Развитие специальных сервисов в сети Интернет, которые обеспечивают привлечение к работе по формированию эталонных массивов текстов большого количества анонимных пользователей, дополнительно повышают актуальность исследований в области оценки качества получаемых таким образом массивов.

Оценку качества эталонных массивов проводить в следующих двух ситуациях.

1. Эталонный массив подготовлен ранее неизвестными экспертами – в данном случае у документа имеется только одна оценка, полученная неизвестным экспертом, и нет возможности управления работой экспертов.

2. Эталонный массив формируется известными экспертами – в данном случае для каждого документа имеется фиксированное число оценок, выполненных известными экспертами, и можно управлять процессом оценки документов.

В современной литературе основное внимание уделяется второй ситуации и оценка качества эталонного массива часто сводится к простой оценке степени согласованности мнений экспертов.

Для оценки согласованности мнений экспертов разработано много различных коэффициентов и показателей. При этом наибольшее распространение получили методы [1],[3],[5],[14], основанные на использовании различных вариантов статистики κ, которая имеет следующий вид:

$$\kappa = \frac{A_0 - A_e}{A_{max} - A_e},$$

где A_0 - базовая статистика, оценивающая согласованность мнений экспертов, A_e - оценка значения A_0 в случае выполнения случайной классификации объектов, A_{max} - оценка максимально возможного значения A_0.

Для проверки гипотезы о наличии статистически значимого отклонения меры согласованности от нулевого значения требуется знание распределения κ. На практике такое распределение удается аналитически описать только для простейших случаев. По этой причине для проверки соответствующих гипотез обычно используют бутстреп метод [9]. В некоторых работах [2] предлагают использовать следующие неформальные оценки значений κ. Если $\kappa < 0.4$, то считается, что мнения не согласованы, если $0.4 \le \kappa < 0.75$, то считается, что мнения слабо согласованы, и, если $\kappa > 0.8$, то мнения сильно согласованы. Однако такой подход является не совсем корректным, так как с ростом числа объектов статистически значимыми являются и меньшие отклонения κ от 0.

Основным же недостатком данных методов является то, что значения статистики κ напрямую не связаны со значениями показателей качества классификации.

Отдельные вопросы прямого влияния ошибок экспертов на качество классификации и информационного поиска также рассматривались в ряде работ. В частности, в [6] приводятся теоретические оценки влияния ошибок экспертов на величину ошибки классификации, ее дисперсию, размер тестовых выборок, в [12] проводится анализ вероятностей ошибок, допускаемых оценщиками в различных ситуациях, в [4] информация о вероятностях ошибок экспертов использовалась для улучшения функций ранжирования документов, а в [8] для оценки верхних границ для показателей качества информационного поиска. Основной проблемой, которая пока не получила эффективного решения, при этом является нахождение соответствующих оценок вероятностей ошибок экспертов.

Рассмотрим теперь формальное описание моделей ошибок экспертов, в рамках которых можно явным образом оценивать их вероятности и влияние на показатели качества классификации.

2 Вероятностные модели ошибок экспертов

2.1 Общее описание вероятностных моделей

Для анализа влияния ошибок в эталонном массиве на показатели качества классификации будем считать, что имеется объект (текст) x, который может быть одновременно отнесен к нескольким классам из множества $\Omega = \{\omega_1, \dots, \omega_k\}$.

Результаты классификации отдельного объекта x можно представить с помощью следующих векторов:

$c^0 = (c_1^0, \dots, c_k^0)$, $c_j^0 \in \{0,1\}$ – ненаблюдаемый истинный вектор эталонной классификации объекта x;

$\hat{c}^0 = (\hat{c}_1^0, \dots, \hat{c}_k^0)$, $\hat{c}_j^0 \in \{0,1\}$ – наблюдаемый экспертный вектор эталонной классификации объекта x (данный вектор может отличаться от истинного вектора из-за наличия ошибок);

$c^1 = (c_1^1, \dots, c_k^1)$, $c_j^1 \in \{0,1\}$ – наблюдаемый вектор оцениваемой (автоматической) классификации объекта x (данный вектор может отличаться от истинного вектора),

где $c_j^0, \hat{c}_j^0, c_j^1 = 1$, если объект x относится к классу ω_j, и $c_j^0, \hat{c}_j^0, c_j^1 = 0$, в противном случае, $j = 1, \dots, k$.

Соответственно результаты классификации n объектов $x_1, \dots, x_n$, которые распределены по k классам $\omega_1, \dots, \omega_k$, могут быть представлены с помощью следующих матриц размера $k \times n$:

$C^0 = \left(c_{ji}^0\right)_{k \times n}$ – ненаблюдаемая истинная матрица эталонной классификации, в которой нет ошибок;

$\hat{C}^0 = \left(\hat{c}_{ji}^0\right)_{k \times n}$ – наблюдаемая экспертная матрица эталонной классификации, в которой есть ошибки;

$C^1 = \left(c_{ji}^1\right)_{k \times n}$ – наблюдаемая матрица автоматической классификации, качество которой оценивается,

где $c_{ji}^0, \hat{c}_{ji}^0, c_{ji}^1 = 1$, если объект x_i относится к классу ω_j, и $c_{ji}^0, \hat{c}_{ji}^0, c_{ji}^1 = 0$, в противном случае, $i = 1, \dots, n$, $j = 1, \dots, k$.

Основные показатели качества классификации могут быть представлены в виде следующих вероятностей:

$P_j^0 = P(\, c_j^0 = 1 |\, c_j^1 = 1)$, $P_j^1 = P(\, \hat{c}_j^0 = 1 |\, c_j^1 = 1)$ – истинное и наблюдаемое значение точности;

$R_j^0 = P(\, c_j^1 = 1 |\, c_j^0 = 1)$, $R_j^1 = P(\, c_j^1 = 1 |\, \hat{c}_j^0 = 1)$ – истинное и наблюдаемое значение полноты;

$E_j^0 = P(\, c_j^0 \neq c_j^1)$, $E_j^1 = P(\, \hat{c}_j^0 \neq c_j^1)$ – истинное и наблюдаемое значение ошибки классификации;

$F_j^0 = \frac{2P\left(c_j^0 = 1, c_j^1 = 1\right)}{P\left(c_j^0 = 1\right) + P\left(c_j^1 = 1\right)}$, $F_j^1 = \frac{2P\left(\hat{c}_j^0 = 1, c_j^1 = 1\right)}{P\left(\hat{c}_j^0 = 1\right) + P\left(c_j^1 = 1\right)}$ – истинное и наблюдаемое значение F-меры.

При этом для обозначения вероятностей классов будем использовать следующие обозначения: $\pi_j^0 = P\left(c_j^0 = 1\right), \pi_j^1 = P\left(c_j^1 = 1\right), \hat{\pi}_j = P\left(\hat{c}_j^0 = 1\right)$.

Далее будем считать, что зафиксирован класс $\omega_j, j = 1, \dots, k$, и все показатели качества, а также элементы векторов и матриц классификации, для сокращения записи будем записывать без индекса j. Например, элементы $c_j^0, \hat{c}_j^0, c_j^1$ будем записывать $c^0, \hat{c}^0, c^1$.

Рассмотрим следующие модели ошибок экспертов:

- модель независимых ошибок – предполагается, что ошибки носят случайный характер и не зависят от значений истинного вектора эталонной классификации;

- модель условных ошибок – предполагается, что ошибки, совершаемые экспертом, зависят от значений истинного вектора эталонной классификации.

2.2 Модель независимых ошибок экспертов

В рамках данной модели взаимосвязь истинной и экспертной классификации можно представить в виде следующего соотношения:

$$\hat{c}^0 = c^0(1 - z) + (1 - c^0)z = c^0 + z - 2c^0 z,$$

где $z \sim Ber(\epsilon)$ – независимая случайная величина, $\epsilon \in [0,1]$ – вероятность успеха, т.е. $P(z = 1) = \epsilon$ и $P(z = 0) = 1 - \epsilon$. Заметим, что при $\mathrm{z} = 1$ справедливо $\hat{\mathrm{c}}^0 \neq \mathrm{c}^0$, а при $\mathrm{z} = 1$ справедливо $\hat{\mathrm{c}}^0 = \mathrm{c}^0$.

Можно показать, что вероятности ошибок первого и второго рода, а также ошибки классификации совпадают и равны ϵ. Также справедливо свойство о независимости ошибок экспертной и автоматической классификации.

Утверждение 1. Пусть $\epsilon \neq \frac{1}{2}$, тогда в рамках модели независимых ошибок справедливы следующие соотношения между истинными и наблюдаемыми значениями показателей качества классификации:

$E^0 = \frac{E^1-\epsilon}{1-2\epsilon}$, $E^1 = E^0(1-2\epsilon)+\epsilon$, где $E^1 \geq \epsilon$,

$P^0 = \frac{P^1-\epsilon}{1-2\epsilon}$, $P^1 = P^0(1-2\epsilon)+\epsilon$, где $P^0 \geq \epsilon$,

$R^0 = \frac{R^1\hat{\pi}-\epsilon\pi^1}{\hat{\pi}-\epsilon}$, $R^1 = \frac{\left((1-2\epsilon)\pi^0R^0+\epsilon\pi^1\right)}{(1-2\epsilon)\pi^0+\epsilon}$, где $R^1\hat{\pi} \geq \epsilon\pi^1$

$F^0 = \frac{\frac{1}{2}F^1(\pi^1+\hat{\pi})-\epsilon\pi^1}{(1-2\epsilon)\pi^1+(\hat{\pi}-\epsilon)}$, $F^1 = 2\frac{\epsilon\pi^1+(1-2\epsilon)(\pi^1+\pi^0)F^0}{\pi^1+\pi^0(1-2\epsilon)+\epsilon}$. ■

Таким образом, с использованием выражений, приведенных в утверждении 1, можно зная уровень ошибок экспертов восстанавливать истинные значения показателей качества классификации по наблюдаемым экспертным показателям.

Заметим, что при $\epsilon = \frac{1}{2}$ экспертные оценки показателей качества становятся не связанными с истинными значениями показателей качества, так как в данном случае $E^1 = 1/2$, $P^1 = 1/2$, $R^1 = \pi^1$, что не позволяет восстанавливать значения истинных показателей качества.

С использованием приведенных соотношений можно оценить диапазон изменения показателей качества при изменении уровня ошибок экспертов.

Следствие 1. При фиксированном значении $\epsilon \in (0,1)$ получаем, что $E^1 \in (\epsilon, 1-\epsilon)$, $P^1 \in (\epsilon, 1-\epsilon)$, $R^1 \in \left(\frac{\epsilon\pi^1}{(1-2\epsilon)\pi^0+\epsilon}, 1-\frac{\epsilon(1-\pi^1)}{(1-2\epsilon)\pi^0+\epsilon}\right)$.

На следующих рисунках приведены значения наблюдаемых показателей качества при фиксированных значениях истинных показателях качества и различных значениях ошибки.

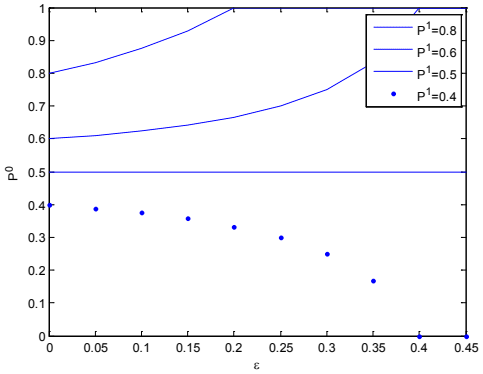


Рис. 1. График зависимости оценки истинной точности P^0 от ошибки эксперта при различных наблюдаемых значениях точности P^1

Из приведенного рисунка видно, что при наблюдаемых значениях точности меньше 0.5 при наличии ошибок истинные значения могут быть еще меньше. При наблюдаемой точности выше 0.5, напротив, истинные значения оказываются выше наблюдаемых значений.

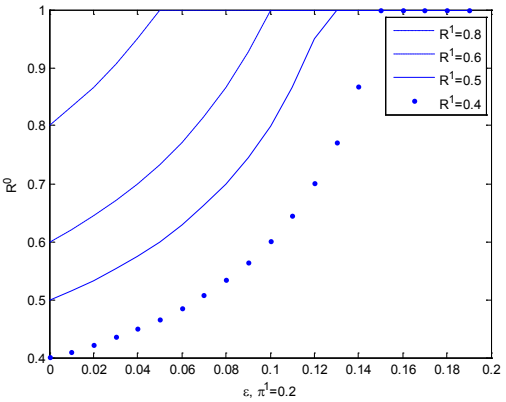


Рис. 2. График зависимости оценки истинной полноты R^0 от вероятности ошибки эксперта при различных фиксированных значениях наблюдаемой полноты R^1 и фиксированных значениях $\pi^1 = \hat{\pi} = 0.2$

Как можно заметить из приведенного рисунка даже при небольших значениях вероятности ошибки эксперта ϵ истинные и наблюдаемые значения полноты могут существенно отличаться.

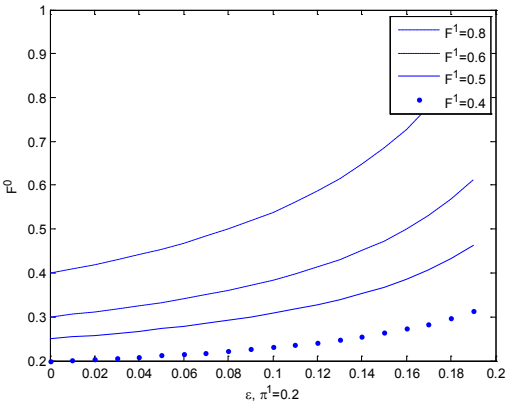


Рис. 3. График зависимости оценки истинной F-меры F^0 от вероятности ошибки эксперта ϵ при различных фиксированных значениях наблюдаемой F-меры F^1и фиксированных значениях $\pi^1 = \hat{\pi} = 0.2$

Как можно заметить из приведенного рисунка ошибка эксперта оказывает относительно меньшее влияние на значения F-меры, чем на значения полноты, но наблюдаемые значения все равно могут заметно отличаться.

2.3 Модель условных ошибок экспертов

В рамках данной модели предполагается, что вероятность ошибки эксперта зависит от того относится документ к рубрике или нет. Взаимосвязь истинной и экспертной классификации можно представить в виде следующего соотношения:

$$\hat{c}^0 = c^0(1-z^1) + (1-c^0)z^2,$$

где $z^1 \sim Ber(\alpha)$ – независимая случайная величина, которая определяет ошибки первого рода, $z_2 \sim Ber(\beta)$ – независимая случайная величина, которая определяет ошибки второго рода.

Действительно, если $c^0 = 1$ и $z_1 = 1$, то $\hat{c}^0 = 0$, что соответствует ошибке первого рода. Если же $c^0 = 0$ и $z_2 = 1$, то $\hat{c}^0 = 1$, что соответствует ошибке второго рода.

Утверждение 2. В рамках модели независимых ошибок справедливы следующие соотношения между истинными и наблюдаемыми значениями показателей качества классификации:

$$P^1 = (1-\alpha-\beta)P^0 + \beta,\ P^0 = \frac{P^1-\beta}{1-\alpha-\beta},$$

$$R^1 = \frac{\pi(1-\alpha-\beta)R^0+\pi^1\beta}{\pi(1-\alpha-\beta)+\beta},\ R^0 = \frac{R^1\hat{\pi}-\beta\pi^1}{\hat{\pi}-\beta}. \blacksquare$$

Таким образом, если известны оценки вероятностей ошибок первого и второго рода для экспертов и наблюдаемые экспертные оценки точности и полноты, то можно вычислить истинные значения показателей точности и полноты.

Полученные соотношения между истинными и наблюдаемыми показателями качества позволяют оценить максимально возможные значения показателей качества, достижимые при определенном уровне ошибок.

Следствие 1. При фиксированных значениях $\alpha, \beta \in (0,1)$ получаем, что $P^1 \in (\beta, 1-\alpha)$, $R^1 \in \left(\frac{\pi^1\beta}{\pi(1-\alpha-\beta)+\beta}, 1-\frac{(1-\pi^1)\beta}{\pi(1-\alpha-\beta)+\beta}\right)$.

3 Оценка размеров эталонных массивов текстов

3.1 Оценка размеров эталонных массивов в рамках модели независимых ошибок

Для оценки размеров эталонных массивов текстов рассмотрим влияние, оказываемое ошибками экспертов на дисперсию выборочных оценок ошибки $\tilde{E}^1$, точности $\tilde{P}^1$ и полноты $\tilde{R}^1$, которые вычисляются следующим образом.

$$\tilde{E}^1 = \frac{1}{n}\sum_{i=1}^{n} I(\tilde{c}_i^0 \neq c_i^1),$$

$$\tilde{P}^1 = \frac{\sum_{i=1,c_i^1=1}^{n} I\,(\hat{c}_i^0 = 1)}{\sum_{i=1}^{n} I\,(c_i^1 = 1)} = \frac{\sum_{i=1}^{n} \hat{c}_i^0 c_i^1}{\sum_{i=1}^{n} c_i^1},$$

$$\tilde{R}^1 = \frac{\sum_{i=1,\hat{c}_i^0=1}^{n} I\,(\hat{c}_i^0 = 1)}{\sum_{i=1}^{n} I\,(\hat{c}_i^0 = 1)} = \frac{\sum_{i=1}^{n} \hat{c}_i^0 c_i^1}{\sum_{i=1}^{n} \hat{c}_i^0},$$

где $I(x) \in \{0,1\}$ – индикаторная функция, $\tilde{n}_0 = \sum_{i=1}^{n} \hat{c}_i^0$, $\tilde{n}_1 = \sum_{i=1}^{n} c_i^1$. Несложно показать, что данные статистики имеют следующие математические ожидания и дисперсии:

$$E\left(\tilde{E}^1\right) = E^1,\ D\left(\tilde{E}^1\right) = \frac{1}{n}E^1(1-E^1),$$

$$E\left(\tilde{P}^1\right) = P^1,\ D\left(\tilde{P}^1\right) = P^1(1-P^1)N^1,$$

$$E\left(\tilde{R}^1\right) = R^1,\ D\left(\tilde{R}^1\right) = R^1(1-R^1)\hat{N}^0,$$

где $N^1 = \sum_{s=1}^{n} \frac{1}{s}\binom{n}{s}(\pi^1)^s(1-\pi^1)^{n-s}$, $\hat{N}^0 = \sum_{s=1}^{n} \frac{1}{s}\binom{n}{s}(\hat{\pi})^s(1-\hat{\pi})^{n-s}. \blacksquare$

Из приведенного утверждения следует, что оценки показателей качества являются несмещенными, но при этом дисперсия является сложной функцией от размера выборки и вероятности успеха.

Для оценки показателей и зависимостей между истинными и наблюдаемыми показателями можно найти оценки истинных показателей с использованием следующих статистик.

$$\tilde{E}^0 = \tilde{P}(\mathrm{c}^0 \neq \mathrm{c}^1) = \frac{\tilde{E}^1 - \epsilon}{1-2\epsilon},$$

$$\tilde{P}^0 = \tilde{P}(\mathrm{c}^0 = 1 | c^1 = 1) = \frac{\tilde{P}^1 - \epsilon}{1-2\epsilon},$$

$$\tilde{R}^0 = \tilde{P}(c^1 = 1 | c^0 = 1) = \frac{\tilde{R}^1\tilde{\pi}^0 - \epsilon\tilde{\pi}^1}{\tilde{\pi}^0 - \epsilon},$$

Утверждение 3. Для статистик $\tilde{E}^0$, $\tilde{P}^0$ и $\tilde{R}^0$ справедливы следующие свойства для математических ожиданий и дисперсий:

$$E\left(\tilde{E}^0\right) = \frac{E^1-\epsilon}{1-2\epsilon},\ D\left(\tilde{E}^0\right) = \frac{1}{n}\left(\frac{\epsilon(1-\epsilon)}{(1-2\epsilon)^2} + E^0(1-E^0)\right),$$

$$E\left(\tilde{P}^0\right) = \frac{P^1-\epsilon}{1-2\epsilon},\quad D\left(\tilde{P}^0\right) = \left(\frac{\epsilon(1-\epsilon)}{(1-2\epsilon)^2} + P^0(1-P^0)\right)N^1,$$

$$E\left(\tilde{R}^0\right) = \frac{R^1\hat{\pi}-\epsilon\pi^1}{\hat{\pi}-\epsilon},\quad D\left(\tilde{R}^0\right) = \left(R^0(1-R^0) + \frac{\epsilon\left(R^0(\hat{\pi}-\epsilon)(1-2\pi^1)+\pi^1(\hat{\pi}-\epsilon\pi^1)\right)}{(\hat{\pi}-\epsilon)^2}\right)N^0,$$

где $N^1 = \sum_{s=1}^{n} \frac{1}{s}\binom{n}{s}(\pi^1)^s(1-\pi^1)^{n-s}$, $\hat{N}^0 = \sum_{s=1}^{n} \frac{1}{s}\binom{n}{s}(\hat{\pi})^s(1-\hat{\pi})^{n-s}$ ■

Заметим, что если бы имелась возможность напрямую подсчитать статистики для истинных значений показателей качества, то их дисперсии были бы равны следующим величинам:

$$D\left(\tilde{\tilde{E}}^0\right) = \frac{1}{n}E^0(1-E^0),$$

$$D\left(\tilde{\tilde{P}}^0\right) = P^0(1-P^0)N^1,$$

$$D\left(\tilde{\tilde{R}}^0\right) = R^0(1-R^0)N^0,$$

где $\tilde{\tilde{E}}^0 = \frac{1}{n}\sum_{i=1}^{n} I(c_i^0 \neq c_i^1)$, $\tilde{\tilde{P}}^0 = \frac{\sum_{i=1}^{n} I(c_i^0=1,c_i^1=1)}{\sum_{i=1}^{n} I(c_i^1=1)}$, $\tilde{\tilde{R}}^0 = \frac{\sum_{i=1}^{n} I(c_i^0=1,c_i^1=1)}{\sum_{i=1}^{n} I(c_i^0=1)}$.

Отсюда получаем, что справедливо следующее следствие из приведенного утверждения.

Следствие 1. Для обеспечения сохранения дисперсии оценок показателей на исходном уровне (соответствует ситуации, когда ошибки отсутствуют) требуется увеличение размера выборки в следующее число раз:

$l_E = \frac{\epsilon(1-\epsilon)}{(1-2\epsilon)^2} + 1$ – увеличение размера выборки для сохранения точности оценивания E^0,

$l_P = \frac{\epsilon(1-\epsilon)}{(1-2\epsilon)^2} + 1$ – увеличение размера выборки для сохранения точности оценивания показателя P^0,

$l_R = \frac{\epsilon\left(R^0(\hat{\pi}-\epsilon)(1-2\pi^1)+\pi^1(\hat{\pi}-\epsilon\pi^1)\right)}{(\hat{\pi}-\epsilon)^2} + 1$ – увеличение размера выборки для сохранения точности оценивания R^0.

На следующем рисунке показаны соответствующие зависимости для различных показателей качества.

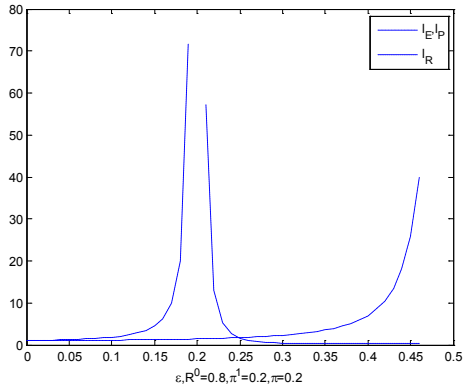


Рис. 4. Графики зависимости увеличения размера выборки для различных показателей качества от величины вероятности ошибки экспертной классификации

Из приведенного рисунка можно сделать вывод, что даже при относительно небольших значениях ошибки классификации может потребоваться существенное увеличение количества данных для обеспечения заданного уровня качества оценивания показателей. Более того чем меньше относительный размер класса, тем большее влияние оказывают случайные ошибки на показатели качества классификации. В частности, если размер класса составляет 20% от размера массива данных, то уже при уровне ошибки в 15% может потребоваться увеличение объема выборки в 10 раз.

3.2 Оценка размеров эталонных массивов в рамках модели условных ошибок экспертов

Для оценивания значений показателей качества воспользуемся статистиками $\tilde{P}^1$, $\tilde{R}^1$, которые были рассмотрены ранее. При этом можно использовать следующие статистики для оценки истинных значений показателей качества классификации:

$\tilde{P}^0 = \frac{\tilde{P}^1-\beta}{1-\alpha-\beta}$, $\tilde{R}^0 = \frac{\tilde{R}^1\hat{\pi}^0-\beta\tilde{\pi}^1}{\tilde{\pi}^0-\beta}$,

Утверждение 4. Для статистик $\tilde{P}^0$ и $\tilde{R}^0$ справедливы следующие свойства для математических ожиданий и дисперсий:

$E\left(\tilde{P}^0\right) = \frac{P^1-\beta}{1-\alpha-\beta}$, $D\left(\tilde{P}^0\right) = \left(P^0(1-P^0) + \frac{\epsilon(1-\epsilon)}{(1-\alpha-\beta)^2} + P^0\frac{\alpha-\beta}{(1-\alpha-\beta)}\right)N^1$,

$E\left(\tilde{R}^0\right) = \frac{R^1\hat{\pi}-\beta\pi^1}{\hat{\pi}-\beta}$,

$D\left(\tilde{R}^0\right) = \left(R^0(1-R^0) + \frac{\beta\left(R^0(\hat{\pi}-\beta)(1-2\pi^1)+\pi^1(\hat{\pi}-\beta\pi^1)\right)}{(\hat{\pi}-\beta)^2}\right)N^0$,

где $N^1 = \sum_{s=1}^{n}\frac{1}{s}\binom{n}{s}(\pi^1)^s(1-\pi^1)^{n-s}$, $\hat{N}^0 = \sum_{s=1}^{n}\frac{1}{s}\binom{n}{s}(\hat{\pi})^s(1-\hat{\pi})^{n-s}$ ■

Отсюда получаем, что справедливо следующее следствие из приведенного утверждения.

Следствие 1. Для обеспечения сохранения дисперсии оценок показателей на исходном уровне (соответствует ситуации, когда ошибки отсутствуют) требуется увеличение размера выборки в следующее число раз:

$l_P = \frac{\epsilon(1-\epsilon)}{(1-\alpha-\beta)^2} + P^0\frac{\alpha-\beta}{(1-\alpha-\beta)} + 1$ – увеличение размера выборки для сохранения точности оценивания показателя P^0,

$l_R = \frac{\beta\left(R^0(\hat{\pi}-\beta)(1-2\pi^1)+\pi^1(\hat{\pi}-\beta\pi^1)\right)}{(\hat{\pi}-\beta)^2} + 1$ – увеличение размера выборки для сохранения точности оценивания R^0.

4 Оценка вероятностей ошибок экспертов

4.1 Общее описание подхода

Для возможности практического использования выявленных зависимостей между истинными и наблюдаемыми значениями показателей качества классификации необходимо знать значения вероятностей ошибок экспертов. Однако их оценка является достаточно сложной задачей по следующим причинам:

1. Истинные матрицы эталонных классификаций являются неизвестными, что не позволяет вычислить ошибки экспертов напрямую;

2. В большинстве случаев доступной является только одна матрица экспертной классификации, что не позволяет оценивать качество работы одних экспертов по отношению к другим экспертам.

В ситуации, когда доступна только одна эталонная экспертная классификация массива документов, можно воспользоваться методами кластерного анализа для выявления «почти дубликатов» документов. Такой подход, в частности, подробно рассматривается и применяется в работах [10] и [4]. При отсутствии ошибок у документов, которые являются «почти дубликатами», должны быть одинаковые векторы классификации. При наличии же ошибок данные вектора будут отличаться.

Результаты выявления «почти дубликатов» или повторного оценивания объектов (документов) экспертами из множества $x_1, \dots, x_n$ можно представить в виде набора кластеров $\Psi = (\psi_1, \dots, \psi_s)$, где $\psi_l = \{x_{l1}, \dots, x_{lm_l}\}$, m_l – число элементов в кластере, s – число кластеров (число документов с повторной оценкой экспертами).

Пусть, как и ранее, зафиксирован некоторый класс ω_j, $j = 1, \dots, k$. Тогда каждому кластеру ψ_l, $l = 1, \dots, s$, можно поставить в соответствие $c_l^0 \in \{0,1\}$ – истинный признак относимости к классу ω_j и вектор наблюдаемых экспертных оценок $c_l = (c_{l1}, \dots, c_{lm_l})$, где $c_{lt} \in \{0,1\}$.

Рассмотрим теперь более подробно оценивание вероятностей ошибок экспертов в рамках модели независимых ошибок и в рамках модели условных ошибок.

4.2 Оценка вероятностей ошибок в рамках модели независимых ошибок

В рамках модели независимых ошибок справедливы следующие равенства:

$$c_{lt} = c_l^0 + z_{lt} - 2c_l^0 z_{lt},$$

где $z_{lt} \in \{0,1\}$ – независимая случайная величина, $P(z_{lt} = 1) = \epsilon, t = 1, .., m_l$.

Для нахождения оценки вероятности ошибки рассмотрим два подхода:

- прямая максимизация функции правдоподобия,

- использование ЕМ-алгоритма.

Нахождение оценки вероятности ошибки путем максимизации специальной функции правдоподобия. В данном случае для решения поставленной задачи рассмотрим для каждого кластера величину $u_l \in \{0,1\}, l = 1, \dots, s$, которая принимает значение равное 1, если $c_{l1} = c_{l2} = \dots = c_{lm_l}$, и 0, в противном случае. Тогда для $l = 1, \dots, s$ справедливо следующее равенство

$$P(u_l = 1) = P(c_{l1} = \dots = c_{lm_l}) = (1-\epsilon)^{m_l} + \epsilon^{m_l}.$$

Из приведенного утверждения следует, что $P(u_l = 1)$ является функцией от вероятности ошибки ϵ, но при этом для вычисления значений u_l не требуется знание истинных значений c_l^0, $l = 1, \dots, s$. Это свойство позволяет для нахождения ϵ воспользоваться методом максимального правдоподобия. В данном случае оценка ϵ является решением следующей оптимизационной задачи:

$$\epsilon^* = \arg\max_{\epsilon} L(u_1, \dots u_s|\epsilon),$$

где $L(u_1, \dots u_s|\epsilon) = \sum_{l=1}^{s} \log(P(u_l = 1)^{u_l}(1 - P(u_l = 1))^{1-u_l})$ - логарифм функции правдоподобия.

Можно показать, что максимум $L(u_1, \dots u_s|\epsilon)$ находится как решение следующего уравнения:

$$\sum_{l=1}^{s} m_l(\epsilon^{m_l-1} - (1-\epsilon)^{m_l-1})\left(\frac{u_l}{(1-\epsilon)^{m_l} + \epsilon^{m_l}} - \frac{(1-u_l)}{1-(1-\epsilon)^{m_l} - \epsilon^{m_l}}\right) = 0.$$

Прямое решение данного уравнения является достаточно сложной задачей. По этой причине для его решения можно воспользоваться численными методами. В тоже время в частном случае, когда $m_l = 2, l = 1, \dots, s$, можно найти точное решение данного уравнения.

Утверждение 5. При $m_l = 2, l = 1, \dots, s$ и $\epsilon < \frac{1}{2}$ максимум функции правдоподобия $L(u_1, \dots u_s|\epsilon)$ достигается при

$$\epsilon^* = \frac{1}{2} - \frac{1}{2}\sqrt{\frac{2}{s}\sum_{l=1}^{s} u_l - 1}.$$

Таким образом, в ситуации, когда для каждого документа имеется только две оценки, возможно явное нахождение оценки вероятности ошибки эксперта. В общем же случае, требуется применение итерационных методов.

Нахождение оценки вероятности ошибки с использованием ЕМ-алгоритма. В данном случае для нахождения оценки вероятности ошибки рассмотрим расширенную функцию правдоподобия $L(c_1, c_1^0, \dots, c_s, c_s^0|\epsilon, \pi)$, в которую входят наблюдаемые признаки $c_l = (c_{l1}, \dots, c_{lm_l})$ и не наблюдаемые признаки $c_l^0, l = 1, \dots, s$, где $\pi = P(c_l^0 = 1)$, ϵ – вероятность ошибки эксперта.

С учетом приведенных обозначений в рамках модели независимых ошибок справедливо следующее равенство

$$logL(c_{11}, \dots, c_{1m_l}, c_1^0, \dots, c_{s1}, \dots, c_{sm_s}, c_s^0|\epsilon) = \sum_{l=1}^{s}\Big(c_l^0 \log\pi + (1-c_l^0)\log(1-\pi) + \sum_{t=1}^{m_l}\big(c_l^0(c_{lt}\log(1-\epsilon) + (1-c_{lt})\log\epsilon) + (1-c_l^0)(c_{lt}\log\epsilon + (1-c_{lt})\log(1-\epsilon))\big)\Big).$$

В соответствии с общей схемой построения ЕМ-алгоритма требуется решение следующих двух задач:

- найти условное математическое ожидание расширенной функции правдоподобия при фиксированных неизвестных параметрах (Е-шаг);

- найти максимум условного математического ожидания расширенной функции правдоподобия по неизвестным параметрам (М-шаг).

В данном случае в рамках Е-шага требуется найти следующее условное математическое ожидание:

$$E(logL(c_1, c_1^0, \dots, c_s, c_s^0, \epsilon, \pi)|c_1, .., c_s, , \epsilon, \pi).$$

Его вычисление сводится к нахождению следующих апостериорных вероятностей

$$g_l = P(c_l^0 = 1|c_{l1}, \dots, c_{lm_l}, \epsilon, \pi), l = 1, \dots, s,$$

которые можно вычислить следующим образом:

$$g_l = \left(1 + \left(\frac{1}{\pi} - 1\right)\left(\frac{1}{\epsilon} - 1\right)^{m_l - 2n_l}\right)^{-1},$$

где $n_l = \sum_{t=1}^{m_l} c_{lt}$ - число единиц в векторе результатов экспертной классификации.

Отсюда получаем выражение для нахождения математического ожидания логарифма функции правдоподобия:

$$E(logL(c_1, c_1^0, \dots, c_s, c_s^0, \epsilon, \pi)|c_1, .., c_s, \epsilon, \pi) = \sum_{l=1}^{s}\Big(g_l \log\pi + (1-g_l)\log(1-\pi) + \sum_{t=1}^{m_l}\big(g_l(c_{lt}\log(1-\epsilon) + (1-c_{lt})\log\epsilon) + (1-g_l)(c_{lt}\log\epsilon + (1-c_{lt})\log(1-\epsilon))\big)\Big).$$

Найдем теперь в рамках М-шага решение следующей задачи:

$$(\epsilon^*, \pi^*) = \arg\max_{\epsilon,\pi} E(logL(c_1, c_1^0, \dots, c_s, c_s^0, \epsilon, \pi)|c_1, .., c_s, \epsilon, \pi).$$

Можно показать, что максимум будет достигаться при

$$\pi^* = \frac{1}{s}\sum_{l=1}^{s} g_l,$$

$$\epsilon^* = \frac{\sum_{l=1}^{s}\sum_{t=1}^{m_l}(g_l + c_{lt} - 2g_l c_{lt})}{\sum_{l=1}^{s} m_l} = \frac{\sum_{l=1}^{s}(m_l g_l + n_l - 2g_l n_l)}{\sum_{l=1}^{s} m_l}$$

Для задания начальных значений параметров ϵ и π можно положить ϵ равной небольшому числу больше 0, например, $\epsilon = 0.01$, а $\pi = \frac{1}{2}$. В качестве критерия завершения работы алгоритма можно использовать два условия: число итераций равно t_{max} – положительное целое число, разница между новым и старым значениями ϵ меньше $\Delta_\epsilon \in (0,1)$.

Теперь можно описать ЕМ-алгоритм оценивания вероятности ошибки экспертов полностью. В качестве входных параметров у него выступают следующие: t_{max}, Δ_ϵ, $c_l = (c_{l1}, \dots, c_{lm_l})$, $l = 1, \dots, s$, - вектора экспертных классификаций.

ЕМ-алгоритм оценивания вероятности ошибки экспертов

1. Инициализация. Положить $t = 0$, $\epsilon^{(t)} = 0.01$, $\pi^{(t)} = \frac{1}{2}$, $n_l = \sum_{j=1}^{m_l} c_{lj}$, $l = 1, \dots, s$.

2. Е-Шаг. Вычислить для $l = 1, \dots, s$ апостериорные вероятности $g_l^{(t)}$ с использованием следующего выражения:

$$g_l^{(t)} = \left(1 + \left(\frac{1}{\pi^{(t)}} - 1\right)\left(\frac{1}{\epsilon^{(t)}} - 1\right)^{m_l - 2n_l}\right)^{-1}.$$

3. М-Шаг. Вычислить оценки параметров $\epsilon^{(t+1)}$ и $\pi^{(t+1)}$ с использованием следующих выражений:

$$\epsilon^{(t+1)} = \frac{\sum_{l=1}^{s}\left(g_l^{(t)}(m_l - 2n_l) + n_l\right)}{\sum_{l=1}^{s} m_l},$$

$$\pi^{(t+1)} = \frac{1}{s}\sum_{l=1}^{s} g_l^{(t)}.$$

4. Критерий завершения работы. Положить $t = t + 1$. Если $t > t_{max}$ или $\left|\epsilon^{(t+1)} - \epsilon^{(t)}\right| < \Delta_\epsilon$, то завершить работу алгоритма, в противном случае, перейти к шагу 2. ■

4.3 Оценка вероятностей ошибок в рамках модели условных ошибок

В рамках модели независимых ошибок справедливы следующие равенства:

$$c_{lt} = c_l^0(1 - z_{lt}^1) + (1 - c_l^0)z_{lt}^2,$$

где $z_{lt}^1, z_{lt}^2 \in \{0,1\}$ – независимые случайные величины, $P(z_{lt}^1 = 1) = \alpha$, $P(z_{lt}^2 = 1) = \beta$, $t = 1, .., m_l$.

Для нахождения оценок значений параметров α и β воспользуемся методом максимального правдоподобия и построим соответствующий ЕМ-алгоритм.

Рассмотрим расширенную функцию правдоподобия $L(c_1, c_1^0, \dots, c_s, c_s^0|\alpha, \beta, \pi)$, в которую входят наблюдаемые признаки $c_l = (c_{l1}, \dots, c_{lm_l})$ и не наблюдаемые признаки $c_l^0, l = 1, \dots, s$, где $\pi = P(c_l^0 = 1)$.

С учетом приведенных обозначений в рамках модели независимых ошибок справедливо следующее равенство

$$logL(c_1, c_1^0, \dots, c_s, c_s^0|\alpha, \beta, \pi) = \sum_{l=1}^{s}\Big(c_l^0 \log\pi + (1 - c_l^0)\log(1 - \pi) + \sum_{t=1}^{m_l}\big(c_l^0(c_{lt}\log(1 - \epsilon) + (1 - c_{lt})\log\epsilon) + (1 - c_l^0)(c_{lt}\log\epsilon + (1 - c_{lt})\log(1 - \epsilon))\big)\Big).$$

В данном случае в рамках Е-шага требуется найти следующее условное математическое ожидание:

$$E(logL(c_1, c_1^0, \dots, c_s, c_s^0|\alpha, \beta, \pi)|c_1, .., c_s, \alpha, \beta, \pi).$$

Несложно заметить, что его вычисление сводится к нахождению следующих апостериорных вероятностей

$$g_l = P(c_l^0 = 1|c_{l1}, \dots, c_{lm_l}, \alpha, \beta, \pi), l = 1, \dots, s,$$

которые можно записать в следующем виде:

$$g_l = \left(1 + \left(\frac{1}{\pi} - 1\right)\left(\frac{\beta}{1 - \alpha}\right)^{n_l}\left(\frac{1 - \beta}{\alpha}\right)^{m_l - n_l}\right)^{-1},$$

где $n_l = \sum_{t=1}^{m_l} c_{lt}$.

Отсюда получаем выражение для нахождения математического ожидания логарифма функции правдоподобия:

$$E(logL(c_1, c_1^0, \dots, c_s, c_s^0|\alpha, \beta, \pi)|c_1, .., c_s, \alpha, \beta, \pi) = \sum_{l=1}^{s}\Big(g_l\log\pi + (1 - g_l)\log(1 - \pi) + \sum_{t=1}^{m_l}\big(g_l(c_{lt}\log(1 - \alpha) + (1 - c_{lt})\log\alpha) + (1 - g_l)(c_{lt}\log\beta + (1 - c_{lt})\log(1 - \beta))\big)\Big).$$

Найдем теперь в рамках М-шага решение следующей задачи:

$$(\alpha^*, \beta^*, \pi^*) = \arg\max_{\alpha,\beta,\pi} E(logL(c_1, c_1^0, \dots, c_s, c_s^0|\alpha, \beta, \pi)|\mathrm{c}, \alpha, \beta, \pi).$$

Можно показать, что максимум будет достигаться при следующих значениях параметров:

$$\pi^* = \frac{1}{s}\sum_{l=1}^{s} g_l,$$

$$\alpha^* = 1 - \frac{\sum_{l=1}^{s} g_l n_l}{\sum_{l=1}^{s} g_l m_l},$$

$$\beta^* = \frac{\sum_{l=1}^{s}(1 - g_l)n_l}{\sum_{l=1}^{s}(1 - g_l)m_l}.$$

Для задания начальных значений параметров α, β и π можно положить $\alpha = \beta = 0.01$, а $\pi = \frac{1}{2}$. В качестве критерия завершения работы алгоритма можно использовать два условия: число итераций равно t_{max} – положительное целое число, разница

между новым и старым значениями α, β меньше $\Delta_\epsilon \in (0,1)$.

Теперь можно описать ЕМ-алгоритм оценивания вероятности ошибки экспертов полностью. В качестве входных параметров у него выступают следующие: t_{max}, Δ_ϵ, $c_l = (c_{l1}, \dots, c_{lm_l})$, $l = 1, \dots, s$, - вектора экспертных классификаций.

ЕМ-алгоритм оценивания условных вероятностей ошибок

1. Инициализация. Положить $t = 0$, $\alpha^{(t)} = \beta^{(t)} = 0.01$, $\pi^{(t)} = \frac{1}{2}$, $n_l = \sum_{j=1}^{m_l} c_{lj}$, $l = 1, \dots, s$.

2. Е-Шаг. Вычислить для $l = 1, \dots, s$ апостериорные вероятности $g_l^{(t)}$ с использованием следующего выражения:

$$g_l^{(t)} = \left(1 + \left(\frac{1}{\pi} - 1\right)\left(\frac{\beta}{1-\alpha}\right)^{n_l}\left(\frac{1-\beta}{\alpha}\right)^{m_l - n_l}\right)^{-1}.$$

3. М-Шаг. Вычислить оценки параметров $\alpha^{(t+1)}, \beta^{(t+1)}$ и $\pi^{(t+1)}$ с использованием следующих выражений:

$$\alpha^{(t+1)} = 1 - \frac{\sum_{l=1}^{s} g_l^{(t)} n_l}{\sum_{l=1}^{s} g_l^{(t)} m_l},$$

$$\beta^{(t+1)} = \frac{\sum_{l=1}^{s} (1 - g_l^{(t)}) n_l}{\sum_{l=1}^{s} (1 - g_l^{(t)}) m_l},$$

$$\pi^{(t+1)} = \frac{1}{s}\sum_{l=1}^{s} g_l^{(t)}.$$

4. Критерий завершения работы. Положить $t = t + 1$. Если $t > t_{max}$ или $\max\left(\left|\alpha^{(t+1)} - \alpha^{(t)}\right|, \left|\beta^{(t+1)} - \beta^{(t)}\right|\right) < \Delta_\epsilon$, то завершить работу алгоритма, в противном случае, перейти к шагу 2. ■

5 Примеры оценивания ошибок экспертов

5.1 Примеры оценивания вероятностей ошибок при наличии нескольких экспертных классификаций

Пример оценивания вероятности ошибки экспертов в рамках модели независимых ошибок. Для иллюстрации оценивания вероятностей ошибок на практике рассмотрим задачу построения классификаторов для оценивания мнений пользователей, которая предлагалась в рамках семинара РОМИП-2012. В рамках РОМИП для оценки качества работы систем вручную были сформированы 3 эталонных массива текстов (массив с оценками книг, массив с оценками фильмов, массив с оценками камер), в которых каждый текст был оценен двумя экспертами по 2-х бальной шкале, 3-х бальной шкале, 5 бальной шкале. В следующей таблице приведены оценки вероятностей ошибок экспертов и оценки вероятностей классов, полученные с помощью ЕМ-алгоритма для массива с оценками книг.

Таблица 1. Вероятности ошибок экспертов для массива с оценками книг

	2 класса	**3 класса**	**5 классов**
Ошибки	0.017 0.017	0.011 0.094 0.081	0.013 0.026 0.070 0.150 0.094
Вероятности классов	0.083 0.918	0.0519 0.294 0.626	0.000 0.030 0.147 0.290 0.357

Приведенный пример показывает, что на практике величины ошибок могут быть достаточно большими и существенно отличаться для различных классов. Знание вероятностей ошибок позволяет получить оценки истинных значений показателей качества классификации, оценить объем исходных данных, необходимых для получения требуемой точности оценивания показателей качества.

Пример оценивания условных ошибок экспертов в рамках дорожки по классификации тональности оценок пользователей РОМИП-2012. Для иллюстрации оценивания вероятностей ошибок рассмотрим опять массив с оценками книг, который был сформирован в рамках РОМИП-2012. В следующей таблице приведены оценки вероятности ошибок первого и второго рода экспертов для различного числа классов, а также оценки вероятности ошибок, которые были получены в рамках модели независимых ошибок.

Таблица 2. Вероятности ошибок экспертов для массива с оценками книг

	$\boldsymbol{\pi}$	$\boldsymbol{\epsilon}$	$\boldsymbol{\alpha}$	$\boldsymbol{\beta}$
2-класса	0.103 0.897	0.0166 0.0166	0.120 0.0061	0.006 0.121
3-класса	0.063 0.302 0.572	0.0110 0.0944 0.0811	0.0998 0.1055 0.0437	0.006 0.090 0.137
5-классов	0.006 0.042 0.155 0.261 0.323	0.013 0.026 0.070 0.150 0.094	0.983 0.174 0.095 0.113 0.054	0.013 0.021 0.066 0.164 0.115

Приведенные данные показывают, что ошибки могут принимать достаточно большие значения и при этом заметно отличаться для различных классов. Это приводит к тому, что оценки качества, получаемые на таком массиве, могут существенно отличаться от истинных значений.

С учетом найденных ошибок можно оценить максимально достижимые значения показателей точности и полноты с использованием следующих соотношений:

$$P^1 \in (\beta, 1-\alpha),$$

$$R^1 \in \left(\frac{\pi^1\beta}{\pi(1-\alpha-\beta)+\beta}, 1-\frac{(1-\pi^1)\beta}{\pi(1-\alpha-\beta)+\beta}\right).$$

Максимальные и минимальные значения показателей для массива с оценками книг приведены в следующей таблице.

Таблица 3. Оценки максимальных и минимальных значений для точности и полноты при классификации на 2 класса

	Класс 1 (отрицательные отзывы)	Класс 2 (положительные отзывы)
Точность	1%-88%	12%-99%
Полнота	1%-91%	11%-98%

Полученные результаты показывают, что при классификации отрицательных отзывов ошибки могут быть значительно выше, чем при классификации положительных отзывов.

5.2 Пример оценивания вероятностей ошибок при наличии одной экспертной классификации

Рассмотрим теперь пример оценивания вероятностей ошибок экспертов в ситуации, когда имеется только одна матрица эталонной классификации. В качестве массива текстов возьмем материалы дорожки классификации нормативно-правовых документов, которая проводилась в рамках РОМИП-2009. Обучающее множество содержит 29943 документа, которые распределены по 721 классу.

Для получения оценивания вероятности ошибок экспертов проведем выявление «дубликатов» документов. При этом будем считать, что документы являются дубликатами, если мера косинусной близости между векторами документов будет больше 0.9. Непосредственный просмотр документов, мера близости между которыми больше данного порога, показал, что они действительно являются почти дубликатами.

В результате оценивания ошибок экспертов первого и второго рода представлены в форме гистограмм распределения значения ошибок по рубрикам на следующих двух рисунках (такая форма выбрана из-за большого числа рубрик).

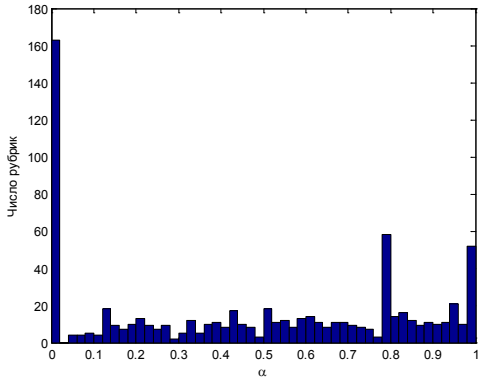


Рис. 5. Гистограмма распределения ошибок первого рода по рубрикам

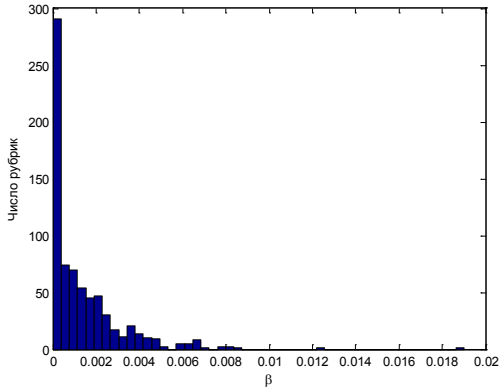


Рис. 6. Гистограмма распределения ошибок второго рода по рубрикам

Анализ полученных результатов показывает, что ошибки первого рода принимают достаточно большие значения и значительно больше ошибок второго рода, что соответствует известному эмпирическому наблюдению, что эксперты при ручной классификации чаще пропускают рубрики, чем добавляют неправильные.

На следующем рисунке также приведены максимальные значения показателей точности и полноты для рубрик, которые вычислены с использованием полученных оценок для ошибок первого и второго рода.

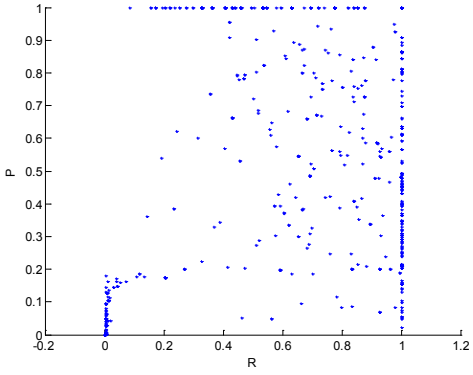


Рис. 7. Значения максимальных значений показателей точности и полноты для рубрик, полученные в рамках модели условных ошибок

Средние значения максимальных значений показателей точности и полноты по всем рубрикам равны следующим значениям:

$$maxP = 0.53,$$
$$maxR = 0.63.$$

Сравнение полученных максимальных значений показателей качества с теми, которые были достигнуты участниками дорожки (максимальное значение точности – 35%, максимальное значение полноты было равно 45%), объясняет получение участниками низких значений показателей качества.

6 Заключение

В работе рассмотрены две модели ошибок экспертов, а также предложен подход к оцениванию их вероятностей, основанный на использовании EM-алгоритма.

Разработанные модели и методы позволяют решать следующие прикладные задачи:

- вычислять значения ошибок экспертов как при наличии нескольких, так и при наличии только одной эталонной экспертной классификации;

- восстанавливать истинные значения показателей качества классификации по наблюдаемым экспертным оценкам значений соответствующих показателей;

- вычислять максимально возможные значения показателей качества классификации при данном уровне ошибок экспертов;

- оценивать величину дисперсии показателей качества и определять размер тестовых выборок, необходимый для обеспечения требуемой точности их оценивания, в зависимости от уровня ошибок экспертов;

- определять рубрики, которые требуют более внимательного оценивания.

Предложенный подход к оценке вероятностей ошибок экспертов является достаточно общим и его можно обобщить и для случая оценивания матриц условных вероятностей, рассматриваемых в работах [4] и [8].

Практическое использование предложенных моделей и методов показано на примерах оценивания ошибок экспертов и максимальных значений показателей качества на материалах дорожек РОМИП-2009 и РОМИП-2012.

Литература

[1] Cohen J. A coefficient of agreement for nominal scales // Educ. Psychol. Measurement. - 1960: Vol. 20. - p. 37-46.

[2] Eye A., Mun E. Y. Analyzing Rater Agreement: Manifest Variable Methods: Taylor and Francis, 2006. – 190 p.

[3] Fleiss J. L. Measuring nominal scale agreement among many raters // Psychological Bulletin. - 1971: Vol. 76. - p. 378-382.

[4] Gulin A., Kuralenok I., Pavlov D. Winning The Transfer Learnin Track of Yahoo!'s Learning To Rank Challenge with YetiRank // Journal of Machine Learning Research, Vol. 14, 2011. - p. 63-76.

[5] Gwet K. L. Handbook of Inter-Rater Reliability: The Definitive Guide to Measuring the Extent of Agreement Among Multiple Raters: Advanced Analytics, LLC, 2010. – 294 p.

[6] Lam C. P., Stork D. G. Evaluating classifiers by means of test data with noisy labels // Proceedings of the International Joint Conference on Artificial Intelligence, AAAI Press, 2003. – p. 513-518.

[7] Lewis D. D., Sebastiani F. Report on the Workshop on Operational Text Classification systems (OTC-01) // SIGIR Forum. - 2001. - 2: Vol. 35. - p. 8-11.

[8] Metricov P., Pavlu V., Aslam J. A. Impact of Assessor Disagreement on Ranking Performance // SIGIR'12. - Portland, Oregon, USA, 2012. - p. 1091-1092.

[9] Reichenheim M. E. Confidence intervals for the kappa statistic // The Stata Journal. - 2004: Vol. 4. - p. 421-428.

[10] Scholer F., Turpin A., Sanderson M. Quantifying Test Collection Quality Based on the Consistency of Relevance Judgements // SIGIR'11, Beijing, China, 2011. - p. 1063-1072.

[11] Sebastiani F. Machine learning in automated text categorization // ACM Comput. Surv. - 2002. - 1: Vol. 34. - p. 1-47.

[12] Webber W., Oard D. W., Scholer F. Assessor Error in Stratified Evaluation // CIKM'10, Toronto, Ontario, 2010. - p. 539-548.

[13] Агеев М. С., Добров Б. В., Лукашевич Н. В. Поддержка системы автоматического рубрицирования для сложных задач классификации текстов // Труды 6-ой Всероссийской научной конференции – RCDL2004, 2004. – 10 с.

[14] Заславский А. А., Пригарина Т. А. Оценка согласованности субъективных классификаций при заданных классах // Социология. - 1994: Vol. 3-4. - с. 84-109.

[15] Лукашевич Н. В. Тезаурусы в задачах информационного поиска. - М.: Издательство Московского Университета, 2011. - 512 с.

Probabilistic models and methods for classifier etalon datasets quality estimation

Vitaly G. Vasilyev

In this paper two probabilistic models of expert errors and special iterative methods of their estimation are proposed. By using this framework the expert errors, size of etalon datasets, maximum values for quality metrics can be calculated. Examples of real calculations are shown on materials of the ROMIP tracks.

Подход к созданию персональной семантической электронной библиотеки

© О. М. Атаева © В. А. Серебряков
Вычислительный центр им. А.А. Дородницына РАН,
Москва
oli@ultimeta.ru serebr@ultimeta.ru

Аннотация

Целью данной работы является разработка информационной системы для создания семантической электронной библиотеки, наполнение которой индивидуально для каждого пользователя системы и выполняется из разнородных источников данных, расположенных на просторах сети и интегрированных в облако LOD. В работе представлена общая схема системы, выделены ее основные модули и дана краткая характеристика каждого из них. Основная задача системы заключается в предоставлении пользователю унифицированного представления для возможности извлечения интересующей пользователя информации по определенной предметной области. Эта предметная область описывается в терминах тезауруса, поддержка которого обеспечивается соответствующим модулем. Обсуждается задача отображения терминов источников данных на этот тезаурус.

1 Введение

Обилие информационных источников в сети вызывает трудности при поиске ресурсов. Поиск ресурсов по определенной тематике отдельно в каждом источнике требует времени, не менее затратно использовать для этого обычные поисковые системы, которые в результатах поиска выдают много «мусора». В связи с этим уже давно возникла насущная необходимость в структурировании информации в сети. Последнее десятилетие наблюдается бурное развитие технологий Semantic Web и активное развитие сообщества, поддерживающего идею Linked Open Data (LOD). Эти события оказали влияние и на электронные библиотеки, которые трансформируются и превращаются центры данных, вокруг которых формируется сообщество заинтересованных экспертов и пользователей, принимающих активное участие в жизни таких динамически развивающихся библиотек. Основной задачей таких центров является интеграция контента различных электронных библиотек, что позволяет увеличить степень повторного использования данных, понизить степень дублирования данных, повысить ценность данных за счет связывания их с другими данными..

Целью данной работы является разработка информационной системы для создания семантической электронной библиотеки, наполнение которой индивидуально для каждого пользователя системы и выполняется из разнородных источников данных, расположенных на просторах сети и интегрированных в облако LOD. Наполнение происходит полуавтоматически, при этом пользователь может не быть осведомлен о структуре данных источника. Система должна быть «проста» в использовании, т.е. не требовать от пользователя специальных знаний. Тематика поисковых запросов по пространству LOD определяется пользователем, с использованием внешних источников данных, в качестве которых, например, могут выступать другие библиотеки. Из результатов поиска пользователь может формировать коллекции, которые доступны также и для внешних систем.

С активным развитием Semantic Web и его относительно нового направления LOD в сети появляются ресурсы, представляющие огромные объемы информации по разным предметным областям. В число этих ресурсов входят и различные электронные библиотеки. Особая ценность их интеграции в LOD обеспечивается возможностью связать данные из различных источников. Возможность использования тезауруса некоторой предметной области в нашей системе позволяет не просто искать и формировать определенные данные в облаке LOD, но и выявлять новые связи между ними, и дополнять уже имеющиеся данные, опираясь на дополнительные возможности системы.

Итак, мы определяем персональную семантическую электронную библиотеку как информационную систему, в качестве ресурсов которой выступают структурированные коллекции разнородных электронных объектов, за формирование которых отвечают пользователи

системы. Эти объекты поступают в систему из различных источников данных зарегистрированных в системе. Для каждого объекта в системе поддерживается набор соответствующей контекстной информации. Средствами системы поддерживается создание и поддержка тезауруса, который представляет знания о предметной области семантической электронной библиотеки. Основная функциональность системы обеспечивает разнообразные средства навигации и поиска по ресурсам и их источникам, доступным через сеть, а также возможность дальнейшей публикации ресурсов библиотеки в LOD. В процессе подключения и описания новых источников данных, тезаурус пополняется новыми понятиями и связями, расширяя тем самым не только область поиска, но и благодаря связям уточнять и конкретизировать тематику поиска..

Проблеме поиска в LOD посвящены различные исследования и существуют поисковые системы, ориентированные на источники, интегрированные в LOD. В работе [9] описывается система поиска в репозиториях LOD на основе высокоуровневой онтологии, на которую отображается схема подключаемого источника данных. Недостаточный уровень концептуализации понятий не позволяет в достаточной мере сконцентрироваться на определенной предметной области. В системах, описанных в работах [10], [11], требуется знание каждого источника данных для задания поисковых запросов. Поисковые системы, такие как Sig.ma, Falcons и SWSE, обеспечивают поиск на основе ключевых слов. Наш подход конкретизирует предметную область, используя тезаурус в рамках семантической электронной бибилиотеки и позволяя связывать результаты поиска с уже имеющимися ресурсами в репозитории бибилиотеки.

2 Семантический подход к электронным библиотекам

В данной работе не затрагиваются задачи построения онтологии библиографических данных для электронных библиотек. Мы рассматриваем задачу построения онтологии электронной библиотеки как информационной системы. Обычно при построении информационных систем на первом этапе выделяют общие понятия, которые не зависят от конкретной предметной области. Далее вводятся определения, характерные для конкретной предметной области, которые соединяются с общими понятиями бинарными отношениями. Наиболее полная онтология для описания информационных систем (онтология BWW) была представлена в работе [2], которая фокусируется на модели представления, определяет набор понятий, их связей и характеристик, достаточных для описания структуры и поведения информационных систем. Основное преимущество этого подхода - это гибкость и расширяемость систем. Концептуальная модель электронных библиотек, с определениями важнейших представлений об архитектуре, ресурсах и функциональности электронных библиотек была определена в программном документ DELOS Digital Library Reference Model [1].

2.1 Онтология электронной библиотеки

На основе изучения «стандартов» [1], [3] в области электронных библиотек можно сказать, что эта область плохо формализована. Некоторые вопросы хорошо исследованы, но моделей, описывающих компоненты построения электронной библиотеки в целом, не существует, что осложняет построение подобного рода систем. На основе понятий концептуальной модели электронных библиотек предложена онтология системы управления персональной семантической электронной библиотекой. Такая библиотека поддерживает различные профили пользователей (эксперты, администраторы, операторы, простые пользователи) с учетом их прав, предоставляет различные сервисы для работы с контентом (формирование коллекций, рубрикация, активация и деактивация источников данных), поддерживает различные процессы управления подсистемами библиотеки (управление классификаторами, наполнение тезауруса по описанию источников данных и т.д.).

Онтология электронной библиотеки, основные понятия которой представлены ниже, разработана в общем виде без привязки к конкретным методам и способам реализации семантических цифровых библиотек. На рисунке 1 приведен фрагмент UML-диаграммы классов основных сущностей разработанной онтологической модели. В верхней части представлены понятия онтологии BWW, нижняя часть представляет их отображение на понятия электронных библиотек. Этот фрагмент взят из работы [13] и представляет понятия онтологии, связанные с подсистемой управления доступом. Работа [13], основывается на подходе также ориентированном на онтологию BWW, что позволяет нам использовать эту модель для электронной библиотеки для описания роли и прав пользователей при взаимодействии с электронной библиотекой, а также определить правила работы.

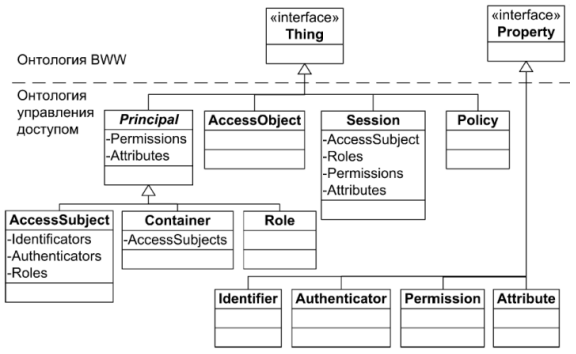


Рисунок 1.

На рисунке 2 приводится фрагмент онтологии подсистемы управления контентом

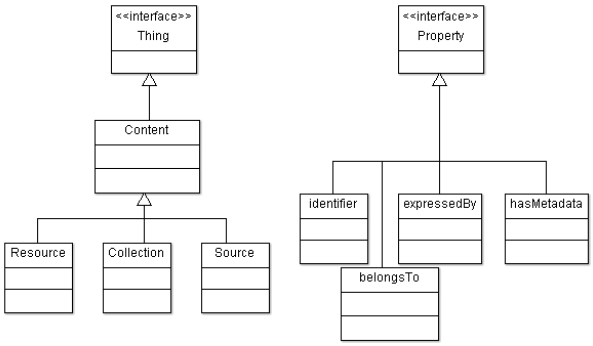


Рисунок 2.

Content (Контент) – это суперкласс объектов электронной библиотеки, задает общие характеристики объектов

Resource (Ресурс) – это информационный объект, множество которых и образует основной контент библиотеки, описание которого дается определенным набором метаданных, представленным соответствующим источником данных

Collection (Коллекция) – подмножество произвольных типов ресурсов

Source (Источник) – представляет собой «параметрическое» описание внешнего по отношению к конкретной библиотеке источника ресурсов (данных) поступающих в систему, где ресурсы могут быть представлены в различных форматах

На рисунке 3 приводится фрагмент онтологии подсистемы управления словарями / классификаторами / тезаурусами

Vocablurary (Словарь) – линейный список терминов

Classifier (Классификатор) – иерархически связанные термины

Taxonomy (Таксономия) - общее представление справочников и словарей

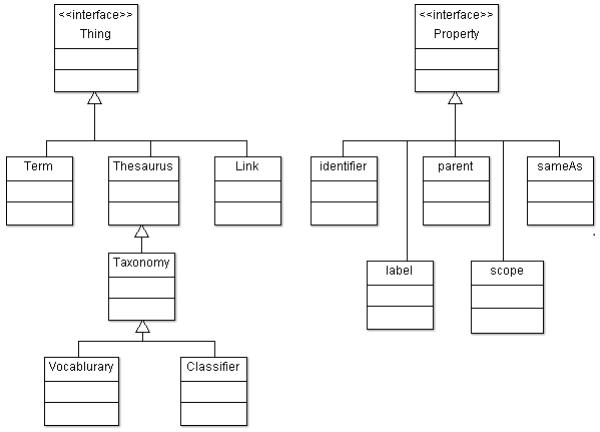


Рисунок 3

Thesaurus (Тезаурус) – является наиболее общей формой таксономии. совокупность словарей/классификаторов, с вертикальными и горизонтальными связями, концепты (элементы) тезауруса могут использоваться как для классификации контента, так и для (описания) источников (ресурсов) данных.

На рисунке 4 приводится фрагмент онтологии подсистемы автоматического мониторинга источников

SavedQuery (Сохраненный запрос) – пользователь определяет запрос к источникам данным, который запрашивает новые объекты за определенный период времени. Запрос определяется с помощью графического интерфейса, то есть от пользователя не требуется специальных знаний, далее системой запрос транслируется в SPARQL и сохраняется в таком виде.

SavedQueryCollection (Коллекция сохраненного запроса) – последняя коллекция новых объектов полученных в результате автоматического мониторинга источников данных

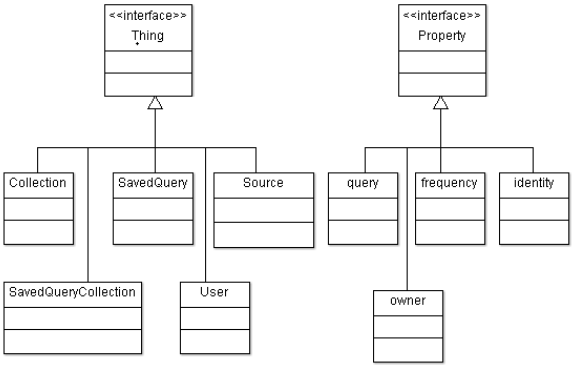


Рисунок 4

3 Общая схема. Основные модули

На рисунке 5 представлена общая схема системы, выделены ее основные модули и дана краткая характеристика каждого.

3.1 Модуль управления доступом

Важной подсистемой любой электронной библиотеки является система управления доступом пользователей к сервисам библиотеки. Пользователь управляет, использует и редактирует контент библиотеки, используя соответствующие доступные сервисы системы. Пользователь должен обладать правами - совокупностью ограничений, накладываемых на него при использовании сервисов системы для работы со своей электронной библиотекой.

3.2 Модуль навигации по ресурсам библиотеки

Подсистема навигации определяет представление данных в различных форматах, обеспечивает навигацию по структуре данных, поддержку тематических подборок, работу с коллекциями объектов, атрибутный поиск, выделение неявных связей между ресурсами по их описаниям.

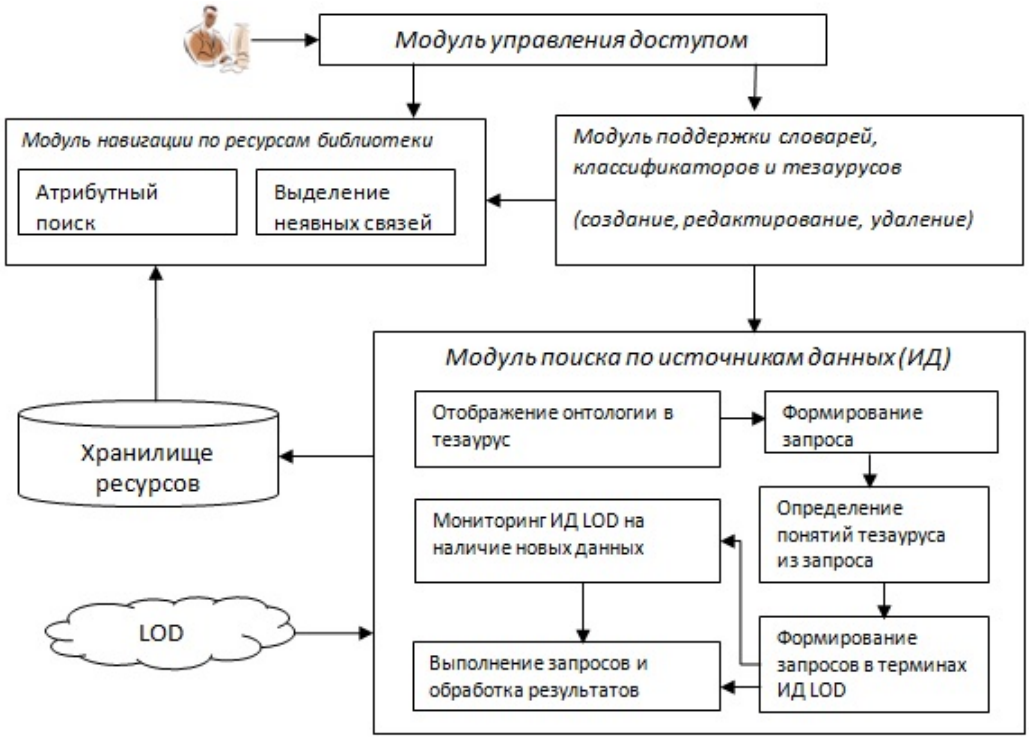


Рисунок 5

3.3 Модуль поиска по источникам данных

Этот модуль позволяет интегрировать данные из внешних систем. Таким образом, контент библиотеки включает не только собственно информационные ресурсы из своего электронного каталога, но и источники данных. Для интеграции данных используется тезаурус, в который отображается описание структуры данных из источников, но при этом тезаурус содержит понятия только той предметной области, в которой заинтересован пользователь. Функция, выполняющая поиск данных во внешних источниках данных, запрашивает описание структур данных из этого тезауруса и отправляет запрос. После этого полученные данные отображаются пользователю. Система может предоставить поиск на естественном языке, при этом пользовательские запросы могут быть проанализированы с помощью функции для трансляции их в запросы для конкретных источников данных. Данный модуль также содержит функцию автоматического мониторинга источников данных, которая информируют пользователей о поступлении новых или изменении существующих информационных объектов в источнике или во внутреннем хранилище, в соответствии с их интересами.

3.4 Модуль поддержки словарей, классификаторов и тезаурусов

Как было отмечено выше, тезаурус обеспечивает представление знания о предметной области семантической электронной библиотеки. Контролируемые словари и классификаторы в системе используются для структуризации данных.

Этот модуль позволяет создавать и наполнять словари, классификаторы, тезаурус, а также позволяет осуществлять просмотр (навигацию) и атрибутивный поиск терминов, чем обеспечивается эффективное выполнение необходимых для этого запросов. Также обеспечиваются функции администрирования тезауруса, при необходимости допускается детализация некоторых связей, а также добавление новых типов связей через интерфейс редактирования плоского словаря этих связей.

Этот модуль поддерживает редактирование набора классификаторов, их структуры и элементов с помощью пользовательских интерфейсов, а также возможность через интерфейсы системы указать перечень классифицируемых типов ресурсов и разорвать связь между некоторым типом ресурсов и классификатором. Каждый классификатор может быть подключен к любому типу ресурсов, и каждый тип ресурсов может классифицироваться несколькими классификаторами.

4 Постороение тезауруса предметной области на основе источников данных

Основная задача системы заключается в предоставлении пользователю унифицированного представления для возможности извлечения интересующей пользователя информации по определенной предметной области. Эта предметная область описывается в терминах тезауруса, поддержка которого обеспечивается соответствующим модулем. При отображении терминов источников данных на этот тезаурус неизбежно возникает основная проблема – неоднородность:

- структурная неоднородность – данные в различных источниках могут быть по-разному представлены и организованы в структуру;
- семантическая неоднородность – данные могут быть представлены в различных системах понятий, схожие понятия могут по-разному интерпретироваться в разных источниках данных.

Для преодоления неоднородности при расширении тезауруса терминами из источников данных необходимо проводить некоторый предварительный анализ понятий источника данных и понятий из тезауруса нашей системы

- на сходство символических имен терминов;
- структурного положения понятия,
- степень сходства множеств атрибутов, достаточных для идентификации объекта и необходимых для его описания, где необходимыми и достаточными атрибутами понятия являются
  - идентифицирующие атрибуты,
  - обязательные описательные атрибуты.

На этом этапе надо уделять внимание анализу связей [4], [6], которые позволяют выявить

- эквивалентные классы,
- ранее не определенные связи между разными источниками данных,
- новые источники данных.

На этапе предварительного анализа актуальны проблемы, классификация которых представлена в работах [7], в которых выделяются следующие группы: лексические, синтаксические, семантические, структурные.

Отображение понятий источников данных в тезаурус производится методом частичного соответствия, где **с**оответствие есть отображение понятий и отношений источника данных на тезаурус [7]. Соответствие может быть определено не полностью и является частичным, если, может существовать несколько понятий в источнике, не имеющих своих эквивалентов в тезаурусе. Для семантического поиска в разных источниках по исследуемой предметной области достаточно такой интеграции на уровне частичного соответствия, которое позволяет избежать изменения в источниках.

При расширении или изменении тезауруса (например, при подключении пользователем нового источника данных для электронной библиотеки) основными операциями являются:

- поиск в тезаурусе понятий эквивалентных понятиям из источника,
- добавление новых понятий из источника в существующий тезаурус,
- привязка к суперпонятиям понятий из источника (если суперпонятия присутствуют в тезаурусе),
- привязка подпонятий к понятиям из источника (если подпонятия присутствуют в тезаурусе).

Основные операции над свойствами:

- добавление новых свойств и связей к понятиям из тезауруса,
- поиск эквивалентных свойств и связей для понятий из источников.

4.1 Источники данных

Источники могут представлять данные трех видов [9]:

- структурированные - предоставляют стандартизированное описание метаданных своих информационных ресурсов, например, в виде онтологий на OWL,
- неструктурированные - не существует каких-либо общепринятых стандартов их представления, но содержат не меньшее количество полезной информации,
- полуструктурированные - обладают некоторой структурой, но не являются жестко структурированными, например, XML.

Недостаточная степень гранулярности структурированных на первый взгляд данных может вызвать затруднения для их обработки, не говоря уже о проблемах в случае с неструктурированными данными. Для преодоления большинства проблем могут использоваться методы text mining. Основная задача text mining - переход от неструктурированного текста к структурированному через последовательность преобразований. Некоторые задачи, которые могут быть решены с их помощью, это:

- выявление связей между ресурсами,
- уточнение тематической направленности,
- выделение ключевых слов.

5 Заключение и дальнейшее направление работы

Сейчас реализовано ядро системы, которое проходит тестовую эксплуатацию. Дальнейшее направление работ планируется в области использования возможностей text mining для анализа сопутствующей текстовой информации и

выявления неявных связей между различными объектами. Эти методы позволят также решать задачи уточнения терминов онтологии предметной области и обработки текстовых документов для более точной их классификации. Реализуется возможность задания запросов на естественном языке к полуструктурированным источникам данных на основе «улучшенной» методами text mining онтологии. Использование методов text mining для уточнения методов построения онтологии предметной области позволит существенно улучшить качество онтологии и позволит соответственно обогащать интегрируемые в системе данные из различных источников, используя более точные понятия и термины, и связи между ними. Также планируется реализовать подсистему позволяющую отслеживать во времени изменение и развитие состояния данных, что позволит оценить «эволюцию» и «распространение» информации по заданной тематике.

Литература

[1] Candela L., Castelli D., Dobreva M., Ferro N., Ioannidis Y., Katifori H., Koutrika G., Meghini C., Pagano P., Ross S., Agosti M., Schuldt H., Soergel D. The DELOS Digital Library Reference Model Foundations for Digital Libraries. IST-2002-2.3.1.12. Technology-enhanced Learning and Access to Cultural Heritage. Version 0.98, December 2007. http://www.delos.info/files/pdf/ReferenceModel/DELOS_DLReferenceModel_0.98.pdf

[2] Weber, R. Ontological Foundations of Information Systems , Queensland, Australia, Coopers & Lybrand. 1997.

[3] Functional Requirements for Bibliographic Records, Final Report / IFLA Study Group on the Functional Requirements for Bibliographic Records. – München: K.G. Saur, 1998. (UBCIM Publications, New Series; v. 19) http://archive.ifla.org/VII/s13/frbr/frbr.htm

[4] Lihua Zhao, Ryutaro Ichise. Integrating Heterogeneous Ontology Schema from LOD.

[5] Proceedings of the 26th Annual Conference of the Japanese Society for Artificial Intelligence (JSAI2012), Yamaguchi, June 12-15, 2012.

[6] Ding, L., Shinavier, J., Finin, T., McGuinness, D.L.: An Empirical Study of owl:sameAs Use in Linked Data. In: Web Science 2010.

[7] Erhard Rahm, Hong Hai Do. Data Cleaning: Problems and Current Approaches. 2000

[8] Isabel F. Cruz and Huiyong Xiao, The Role of Ontologies in Data Integration, Journal Of Engineering Intelligent Systems, 2005, volume 13, pages 245—252.

[9] М. Р. Когаловский. Метаданные, их свойства, функции, классификация и средства представления RCDL, 2012. 3-14

[10] Jain, P., Verma, K., Yeh, P.Z., Hitzler, P., Sheth, A.P.: LOQUS: Linked Open Data SPARQL Querying System. Technical report, Tech. rep., Kno. e. sis Center, Wright State University, Dayton, Ohio, 2010. Available from http://www.pascal-hitzler.de/resources/publications/loqus-tr-2010. pdf (2010)

[11] Hartig, O.; Bizer, C.; and Freytag, J.-C. 2009. Executing SPARQL Queries over the Web of Linked Data. In ISWC 2009 , volume 5823 of LNCS , 293–309

[12] Quilitz, B., and Leser, U. 2008. Querying Distributed RDF Data Sources with SPARQL. In ESWC 2008 , volume 5021 of LNCS, 524–538.

[13] Созыкин А.В. Семантическая интеграция управления доступом к сервисам. Интернет ресурс. – Режим доступа : www/URL: http://asozykin.ru/sites/default/files/sozykin.pdf

An Approach to Creating a Personal Semantic Digital Library

O.M. Atayeva, V.A. Serebryakov

The aim of this work is to develop an information system for creation of semantic digital library which content is individual for each user and is populated from various data sources located on the Web and integrated into LOD cloud. We specify its main modules and provide brief characteristic of each of the modules. The system gives to the user a unified presentation in order to enable retrieval of the information on a certain subject area the user is interested in. This subject area is described in terms of the thesaurus supported by a respective module. The data source terms mapping onto this thesaurus is discussed.

Разработка семантических электронных библиотек на основе онтологических моделей

© Ле Хоай, © А.Ф Тузовский

Национальный исследовательский Томский политехнический университет
Оптимизация систем управления, Институт кибернетики.
lehotomsk@yahoo.com, tuzovskyaf@tpu.ru

Аннотация. *Рассматриваются основные функции электронных библиотек на основе использования семантических технологий, делается постановка задач их реализации. Предлагаются методы их решения и демонстрируются примеры реализации функций семантической электронной библиотеки в разработанной системе.*

1 Введение

Под электронными библиотеками (ЭБ) понимаются информационные системы, позволяющие автоматизировать работу пользователей с различными электронными ресурсами: документами, изображениями, мультимедиа-файлами. Реализация подобных систем сталкивается с рядом ключевых проблем [1]:

- интеграция разнородной информации (электронных ресурсов, пользовательских профилей, структур знаний предметных областей), представленной разными способами на основе использования различных метаданных;
- обеспечение надёжности результатов поиска и навигации по ресурсам;
- обеспечение точности категоризации электронных ресурсов.

Основной причиной возникновения этих проблем является описание электронных ресурсов (ЭР) в системе в виде набора терминов естественных языков (ключевых слов) и формирование логических выводов об их смысловом содержании без учёта синонимии, полисемии и омонимии. Это влечёт за собой снижение качества функций, предоставляемых ЭБ.

Для решения перечисленных проблем и повышения эффективности работы с ЭР требуется выполнять работу с их семантикой, для чего используются модели описания семантики (таксономии, тезаурусы, онтологии) и технологии Semantic Web (семантические технологии). Использование таких технологий позволяет реализовать работу с ЭР на новом уровне, с учётом содержащегося в них смысла. Электронные библиотеки, разработанные с использованием подобных технологий, обычно называются семантическими ЭБ (СЭБ).

В данной статье предлагается способ реализации основных функций СЭБ, таких как: семантический поиск, формирование рекомендаций, выполнение навигации по ресурсам и автоматическая категоризация.

2 Онтологический поход к описанию ЭР

Основной идеей данного подхода является использование онтологий предметных областей для аннотирования содержания электронных ресурсов [2]–[4]. В СЭБ описание электронных ресурсов, содержащих знания из разных предметных областей, выполняется с использованием специально разработанных онтологий этих областей, описанных с помощью таких языков, как RDFS или OWL [5],[6].

Под описанием (аннотированием) ресурса понимается его семантическое метаописание, в виде набора простых высказываний (триплетов) на основе онтологической модели, в состав которых могут входить контекстные и контентные семантические метаданные.

Сделаем вначале несколько базовых определений.

Определение 1. *Под онтологической моделью (онтологией) O понимается знаковая система* $\langle C, P, I, L, T \rangle$*, где, C – множество элементов, которые называются понятиями; P – множество элементов, называемые свойствами (двуместными предикатами); I – множество экземпляров понятий; L – множество текстовых меток или значений понятий и свойств; T – частичный порядок на множестве C и P.*

С помощью набора предикатов P онтологии могут описываться различные отношения между понятиями и экземплярами. Эти отношения задаются с использованием простых утверждений (триплетов) $\langle s, p, o \rangle$, где s и o – это субъект и объект высказывания, а $p \in P$ – это предикат онтологии O.

Считаем, что любому свойству $p \in P$ может быть задан весовой коэффициент (семантический вес) $pv \in [0, 1]$, задающий смысловую близость между субъектом и объектом утверждения (1 – субъект и объект считаются сходными по семантике, а 0 – не похожими), составленного с использованием данного свойства.

Определение 2. *Контекстными метаданными ресурса s* (*заданного с помощью идентификатора*

URI) называется *набор простых утверждений* (*триплетов*) $Мк = \{t_i = \langle s, p_i, o_i \rangle \mid i = 1, n\}$, *где* $s \in I$ *– это аннотируемый ресурс* (*субъект*), $o \in C \cup I \cup L$ *– объект утверждения*, $p \in P$ *– отношение между субъектом и объектом*.

Таким образом, под набором *Мк* ресурса *s* в СЭБ понимаются утверждения о его связи с другими объектами, понятиями из общих онтологий библиотеки, таких как онтологии пользователей, онтологии ресурсов и онтологии предметных областей.

Определение 3. *Контентные метаданные ресурса – это набор простых утверждений (*кортежей*)* $Мс = \{t_j = \langle s_j, p_j, o_j, v_j \rangle \mid j = 1, m\}$, *где* $s \in C \cup I$ *– это субъект утверждения*, $o \in C \cup I$ *– объект утверждения*, $p \in$ P *– отношение между субъектом и объектом*, *а* v *– весовой коэффициент, который оценивает значимость данного утверждения*.

Под набором *Мс* ресурса *s* понимаются утверждения о знаниях, содержащихся в самом аннотируемом ресурсе.

В СЭБ набор M(s) = *Мк*(s) ∪ *Мс*(s) будет называться семантическими метаданными ресурса *s*. Таким образом, все компоненты библиотеки (электронные ресурсы, пользователи, категории и т.п.) описываются метаданными, представленными с помощью RDF-триплетов на основе использования элементов некоторых онтологий. Множество триплетов, описывающих онтологию *O* и их экземпляры *s*, формирует базу знаний (БЗ, Knowledge Base) СЭБ. К этой базе знаний могут выполняться запросы, описанные на некотором языке (например, SPARQL или SERQL [8], [9]), и на основе их обработки могут решаться различные задачи, позволяющие реализовать услуги, предоставляемые СЭБ.

3 Постановка задач и их решение

Доступные функции СЭБ (семантический поиск, категоризация и формирование рекомендаций) реализуются с учетом семантики описаний ЭР. В связи с этим необходима некоторая оценка (метрика) семантической близости различных объектов.

3.1 Семантическая близость

Семантическая близость (смысловое сходство) может определяться между *разными компонентами триплетов, триплетами и наборами триплетов*. Существуют различные методы вычисления таких метрик [4], которые имеют свои сильные и слабые стороны. В рамках данной работы они не анализируются. В ходе выполнения исследований по созданию СЭБ авторами были разработаны новые методы оценки семантической близости, учитывающие специфику разрабатываемой системы.

3.1.1 Метод вычисления семантической близости между компонентами триплетов

Пусть $\mathrm{Sim}(\alpha, \beta)$ – семантическая близость между элементами α и β, где $\alpha, \beta \in C \cup I \cup P \cup$ T. Для вычисления $\mathrm{Sim}(\alpha, \beta)$ необходимо построить неориентированный граф *GO* из всех имеющих триплетов в БЗ. Граф *GO* создаётся в соответствии со следующими правилами:

- *используются только триплеты, у которых значения весовых коэффициентов предикатов не равны нулю* ($pv \neq 0$)*;*
- *вершинами графа являются субъекты и объекты триплетов, а ребра графа, соединяющие субъекты с объектами имеют веса, равные значению pv предиката того триплета, с использованием которого они были сформированы;*
- *инверсное отношение* (на основе предиката *owl:inverseOf*) между *предикатами p1* (*pv*1) и *p2* (*pv*2) *добавляет в граф два ребра с весами pv*1 *и pv*2, например: для триплета $\langle s, p1, o \rangle$ добавляются следующие два ребра: $\langle s, pv1, o \rangle, \langle o, pv2, s \rangle$*;*
- *симметричное отношение добавляет в граф два ребра с равными весами, например*: <*owl:sameAs*> добавляет два ребра со значениями *pv = 1.0*.

Под путём $PATH(\alpha, \beta)$ между двумя вершинами α и β графа *GO* понимается набор рёбер (предикатов) ведущих от вершины α до вершины β, с учётом их направленности. В этом случае значение $\mathrm{Sim}(\alpha, \beta)$ между этими вершинами вычисляется следующим образом:

$$\mathrm{Sim}(\alpha, \beta) = \max_{i=1 \to k} \left(\mathrm{Sim}_{PATH_i}(\alpha, \beta) \right), \qquad (1)$$

где k – число возможных путей графа *GO* от вершины α до вершины β. Значение семантической близости между элементами α и β по направлению пути i $\mathrm{Sim}_{PATH_i}(\alpha, \beta)$ определяется по следующей формуле:

$$\mathrm{Sim}_{PATH_i}(\alpha, \beta) = \prod_{j=1}^{h_i} pv_{i,j}, \qquad (2)$$

где h_i – число семантических отношений между элементами α и β на пути i, $pv_{i,j}$ – значение веса ребра на основе j-ого семантического предиката на пути i. На основе формул 1 и 2 можно получить окончательную формулу для определения семантической близости между вершинами α и β:

$$\mathrm{Sim}(\alpha, \beta) = \max_{i=1 \to k} \left(\mathrm{Sim}_{PATH_i}(\alpha, \beta) \right) = \max_{i=1 \to k} \left(\prod_{j=1}^{h_i} pv_{i,j} \right) \qquad (3)$$

Величина $\mathrm{Sim}(\alpha, \beta)$ удовлетворяет следующим свойствам: $\mathrm{Sim}(\alpha, \beta) \in [0,1]$; $\mathrm{Sim}(\alpha, \beta) = 0$ при отсутствии пути от α к β; $\mathrm{Sim}(\alpha, \alpha) = \mathrm{Sim}(\beta, \beta) = 1$. В исключительном случае $\mathrm{Sim}(\alpha, \beta)$ может равняться 1, при условии существовании инверсного отношения между элементами α, β.

3.1.2 Метод вычисления семантической близости между триплетами

Пусть $\mathrm{Sim}(t_1, t_2)$ – семантическая близость между триплетами t_1 и t_2. Близость между триплетами вычисляется на основе близостей между их компонентами. В данной работе учитываются свойства инверсного отношения между предикатами (Если два предиката p_1 и p_2 имеют отношение $\langle p_1, owl{:}inverseOf, p_2\rangle$, то при наличии триплета $\langle s, p_1, o\rangle$ подразумевается триплет $\langle o, p_2, s\rangle$) [7].

Имеются следующие две ситуации:

- *если* $t_1, t_2 \in Mк$: то $\mathrm{Sim}(t_1, t_2)$ вычисляется по следующей формуле:

$$\mathrm{Sim}(t_1,t_2) = \mathrm{Sim}(p_1, p_2) \times \mathrm{Sim}(o_1, o_2) \quad (4)$$

- *если* $t_1, t_2 \in Mc$: то $\mathrm{Sim}(t_1, t_2)$ определяется следующим образом:

$$\mathrm{Sim}(t_1,t_2) = \begin{cases} |k| \dfrac{\mathrm{Sim}(s_1, s_2) + \mathrm{Sim}(o_1, o_2)}{2} \omega(t_1,t_2), \forall k > 0 \\ |k| \dfrac{\mathrm{Sim}(s_1, o_2) + \mathrm{Sim}(o_1, s_2)}{2} \omega(t_1,t_2), \forall k \le 0 \end{cases}, \quad (5)$$

где $\omega(t_1, t_2) = v_1 \times v_2$ – функция весовых коэффициентов значимости двух триплетов; $k = \mathrm{Sim}(p_1, p_2)$ и $\mathrm{Sim}(t_1, t_2) \in [0, 1]$; $\mathrm{Sim}(t_1, t_2) = 0$ при $k = 0$. $\mathrm{Sim}(p_1, p_2)$ также вычисляется по формуле 3.

3.1.3 Метод вычисления семантической близости между наборами триплетов

Семантическая близость между наборами триплетов может быть вычислена с использованием двух методов: на основе метода предложенного в работе [10] и метода косинусной меры в модели векторного пространства [11]. Для вычисления семантической близости между наборами триплетов $T_1 = \{t_i = \langle s_i, p_i, o_i\rangle \mid i = 1, k\}$ и $T_2 = \{t_j = \langle s_j, p_j, o_j\rangle \mid j = 1, h\}$ используются следующие формулы:

$$\mathrm{Sim}(T_1,T_2) = \frac{\sum_{t_i \in T_1} \max\left(\mathrm{Sim}(t_i, T_2)\right)}{|T_1|} = \frac{\sum_{i=1}^{k} \max_{j=1 \to h}\left(\mathrm{Sim}(t_i, t_j)\right)}{k} \text{ и} \quad (6)$$

$$\mathrm{Sim}(T_1,T_2) = \frac{T_1 \times T_2}{\sqrt{T_1^2} \times \sqrt{T_2^2}} = \frac{\sum_{i=1}^{k}\sum_{j=1}^{h} \mathrm{Sim}(t_i, t_j)}{\sqrt{\sum_{i=1}^{k}\sum_{j=1}^{k} \mathrm{Sim}(t_i, t_j) \times \sum_{i=1}^{h}\sum_{j=1}^{h} \mathrm{Sim}(t_i, t_j)}}, \quad (7)$$

где $\mathrm{Sim}(T_1, T_2) \in [0, 1]$ и $\mathrm{Sim}(t_i, t_j)$ – семантическая близость между триплетами, вычисляемая по формуле 4 (если $\forall t_i, t_j \in Mк$) или формуле 5 (если $\forall t_i, t_j \in Mc$).

3.2 Семантический поиск

Как отмечалось ранее, метаописание ЭР на основе онтологической модели O рассматривается с точки зрения его контекста и контента. На основе контекстных метаданных и контентных метаданных, запрос семантического поиска q представляется в виде набора $M(q) = (Mк(q), Mc(q))$, а объект кандидата d (возможный результат на данный запрос) – $M(d) = (Mк(d), Mc(d))$. Тогда, задачу семантического поиска можно переформулировать следующим образом: Имеется запрос q с его набором $M(q)$, тогда результатом выполнения данного запроса q будет конечный набор из k объектов знаний (ЭР) $D = \{d_i \mid i = 1, k\}$, где каждый d с набором M(d) удовлетворяет следующему условию:

$$\mathrm{Sim}(M(q), M(d) = \alpha \times \mathrm{Sim}(Mк(q), Mк(d)) + \beta \times \mathrm{Sim}(Mc(q), Mc(d)) > \varepsilon, \quad (8)$$

где $\varepsilon \in (0, 1)$ – пороговое значение, α и β – коэффициенты близости по контексту и по контенту, соответственно и $\alpha + \beta = 1.0$. Значения α и β настраиваются так, чтобы значение семантической близости корректно определяло важность контента и контекста искомого объекта. Если более важным является содержание искомого объекта (контента), тогда значение β превышает значение α и наоборот.

Кроме семантического поиска, в СЭБ также доступен и поиск по графу, который получает широкое применение в социальных сетях (например: *Facebook*) [13].

3.3 Формирование рекомендаций

В данной работе под формированием рекомендаций понимается предоставление пользователям набора ресурсов, релевантных рассматриваемому ЭР. Пусть в СЭБ имеется ЭР d с метаданными $M(d) = (Mк(d), Mc(d))$. Задача формирования рекомендаций для документа d может рассматриваться как выполнение семантического поиска с использованием запроса $q = d$ с наборами $Mк(d)$ и $Mc(d)$. Набор кандидатов $\mathrm{R} = \{dr_i \mid dr_i \neq d \wedge i = 1, h\}$ на запрос q считаются рекомендуемыми для документа d если для любого $dr \in \mathrm{R}$ выполняется условие:

$$\alpha \times \mathrm{Sim}(Mк(d), Mк(dr)) + \beta \times \mathrm{Sim}(Mc(d), Mc(dr)) > \varepsilon \quad (12)$$

Выборы значений α, β и порогового значения ε делаются вручную или автоматически на основе количества триплетов метаописаний ресурса d.

3.4 Автоматическая категоризация

В СЭБ пользователь может создавать свои категории (рубрики), а система автоматически будет относить релевантные ресурсы к заданным категориям. Одна категория может включить в себя другие категории, и они структурируются в виде какой-либо иерархии (например: категория – подкатегория).

Категория k может описываться конечным набором шаблонных ресурсов $Dk = \{tr_i \mid i = 1, h\}$ и каждый tr_i имеет своё метаописание $M(tr_i)$. Любой ресурс $dr \notin Dk$ считается релевантным к заданной категории k при выполнении следующего условия:

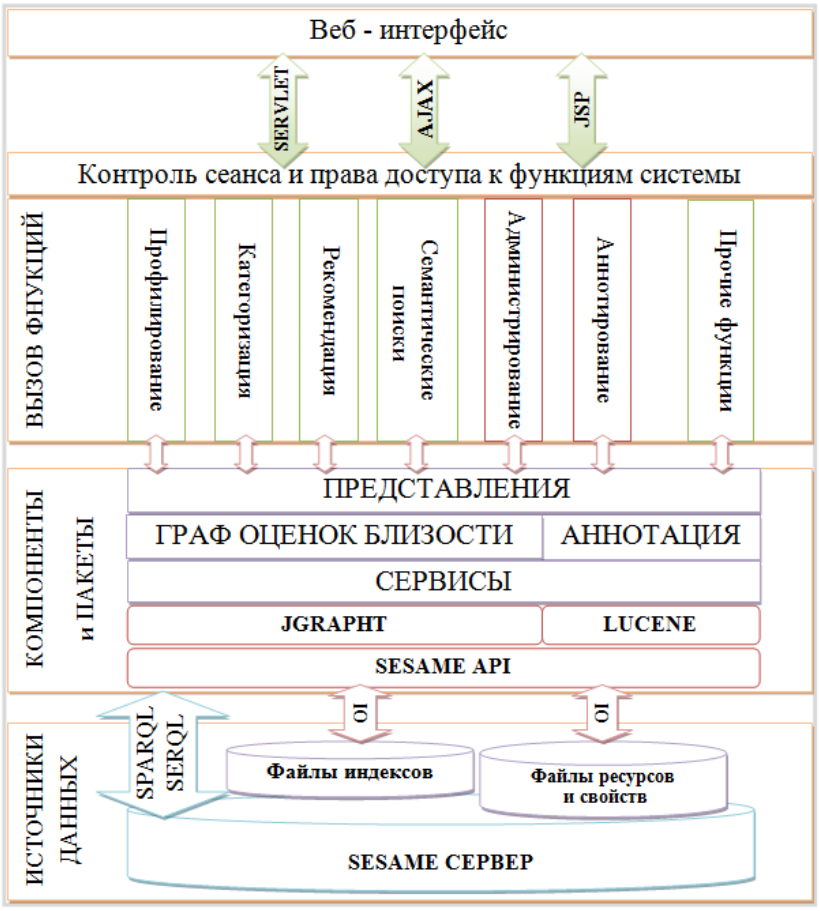


Рис. 1. Программная архитектура системы SemDL

$\alpha \times \mathrm{Sim}(M_к(Dk), M_к(dr)) + \beta \times \mathrm{Sim}(Mc(Dk), Mc(dr)) > \varepsilon$, (13)

где $M_к(Dk) = M_к(tr1) \cup \ldots \cup M_к(trh)$, $Mc(Dk) = Mc(tr1) \cup \ldots \cup Mc(trh)$. В качестве шаблонных ресурсов могут служить существующие ЭР или наборы триплетов (контекст и контент), созданные вручную для описания созданной категории. Кроме метода категоризации по набору шаблонных ресурсов в СЭБ выполняется категоризация по элементам используемых в СЭБ онтологий, описывающих предметные области.

3.5 Метод просмотра

В СЭБ все ЭР (экземпляры), понятия (элементы) онтологии (кроме лексических и числовых данных) имеют уникальные идентификаторы (*URIs*). Такими идентификаторами могут быть субъекты, объекты и предикаты триплетов в БЗ. В связи с этим имеется возможность просмотра содержания СЭБ по субъектам или объектам описаний ЭР. Реализация функции просмотра позволяет пользователям находить и просматривать все допустимые отношения (триплеты), связанные с рассматриваемым ресурсом. Например: по ссылке автора ресурса, можно найти все его публикации, его интересы и т.п., а далее по этим публикациям и интересам могут быть выполнены следующие переходы.

4 Программная реализация в SemDL

Предложенные методы решения задач по реализации описываемых функций СЭБ применены в разработанной системе *SemDL* [10].

4.1 Программная архитектура SemDL

Программная архитектура системы *SemDL* показана на рис. 1. Она логически разделена на четыре уровня. В соответствии с этим разделением, пользователи взаимодействуют с системой с помощью веб-интерфейса и под контролем системы могут получить доступ к интересующим их функциям системы. Системные компоненты (пакеты) группируются по четырём категориям на основе их функциональности:

- ПРЕДСТАВЛЕНИЯ (*VIEWS*): включают в себя различные разработанные теги (например, HTML-теги) и сервлеты (*servlets*), которые непосредственно обрабатывают полученные запросы и возвращают ответы системы.
- АННОТАЦИЯ (*ANNOTATION*): включает функции для работы с текстовыми документами (извлечение, индексирование). В основном, данный пакет выполняет поиска кандидатов для аннотирования ресурсов (создание метаописания ЭР). Индексирование и поиск ключевых слов возлагается на библиотеку с открытым исходным

кодом *LUCENE ENGINE* [14], [15].

• ГРАФ ОЦЕНОК (*WG*): данный пакет предназначен для создания и индексирования графа семантических оценок *GO*, и вычисляет семантическую близость между компонентами триплетов, наборами триплетов по контенту и контексту. Элементарные операции с графом осуществляется с помощью библиотеки с открытым исходным кодом *JGRAPH* [16]-[18].

• СЕРВИСЫ *SEMDL*: данный пакет осуществляет доступ к базе знаний (*SESAME*) и выполняют различные операции с хранимыми данными, а также обеспечивают фильтрацию ресурсов по наборам триплетов контента и контекста. Для работы с *SESAME* используется библиотека с открытым исходным кодом *OPENRDF-SESAME-API* [19], [20].

4.2 Реализация

Примеры интерфейса системы SemDL показаны на рис. 2–3. Составление запроса с использованием набора триплетов по контексту или контенту для семантического поиска выполняется с помощью рекомендации субъектов, предикатов и объектов на основе вводимых терминов (слов). Результаты поиска, рекомендации или категоризации ранжируются в порядке уменьшения оценок семантической близости. Ресурсы в результатах кратко показываются с основными свойствами (авторы, ключевые слова), по которым можно выполнять переходы к другим ресурсам. При работе с категориями и рекомендациями для получения желаемых результатов можно настраивать значения параметров α, β и ε. Значения параметров α и β по умолчанию определяются на основе количества триплетов для описания контекста и контента ресурса.

4.3 Обоснование полученного решения

Процесс семантического поиска. Схема процесса выполнения семантического поиска показана на рис. 4. Пользователь выполняет семантический поиск с помощью составленных триплетов контекста и контента $M(q)$. На основе $M(q)$ выполняется фильтрация возможных кандидатов (ЭР) d, которые могут быть ответом на запрос q. В результате чего получаются наборы: S_{PRK} – набор возможных кандидатов по контексту и S_{PRC} – набор по контенту. В дальнейшем между метаописаниями M(q) и M(d) вычисляются близости по формуле 8. Ресурсы d, которые удовлетворяют заданному запросу q, ранжируются по степени убывания значений их близости обрабатываемому запросу.

Фильтрация возможных кандидатов. Для повышения эффективности работы данного алгоритма имеется возможность выполнять фильтрацию возможных кандидатов с использованием описанных ниже методов.

Метод фильтрации по набору триплетов контекста. Пусть задан набор триплетов контекста некоторого объекта s: $T_K = \{t_i = \langle s, p_i, o_i\rangle \mid i = 1, k\}$. Для любого элемента e триплета имеется непустой список связанных элементов по семантике Exs: $Exs(e) = \{e, e_i \mid i = 0, h \wedge \mathrm{Sim}(e, e_i) > \varepsilon\}$, где ε - пороговое значение близости и $\mathrm{Sim}(e, e_i)$ вычисляется по формуле 3.

Пусть $Exs_P(T_K)$ – список всех предикатов из набора T_K и их связанных элементов по семантике, $Exs_P(T_K) = Exs(p_1) \cup \ldots \cup Exs(p_k)$; $Exs_O(T_K) = Exs(o_1) \cup \ldots \cup Exs(o_k)$ – список всех объектов из набора T_K и их связанных элементов по семантике.

На основе списков связанных элементов по семантике предложенный метод фильтрации по набору T_K допускает только тот ресурс $prk \in S_{PRK}$ с набором триплетов его контекста $T_{PRK} = \{t_j = \langle prk, p_j, o_j\rangle \mid j \in [1, h]\}$, который удовлетворяет следующему условию:

$$(\exists t_j \in T_{PRK}) \wedge (p_j \in Exs_P(T_K)) \wedge (o_j \in Exs_O(T_K)) \quad (14)$$

Метод фильтрации по набору триплетов контента. Пусть заданный набор триплетов контекста некоторого объекта s: $T_C = \{t_i = \langle s_i, p_i, o_i\rangle \mid i = 1, k\}$. Для набора T_C имеются следующие списки связанных элементов по семантике для компонентов всех триплетов: $Exs_S(T_C) = Exs(s_1) \cup \ldots \cup Exs(s_k)$; $Exs(p_1) \cup \ldots \cup Exs(p_k)$; $Exs_O(T_K) = Exs(o_1) \cup \ldots \cup Exs(o_k)$.

На основе списков связанных элементов по семантике предложенный метод фильтрации по набору T_C допускает только тот ресурс $prc \in S_{PRC}$ с набором триплетов его контента $T_{PRC} = \{t_j = \langle s_j, p_j, o_j\rangle \mid j = 1, h\}$, который удовлетворяет следующему условию:

$$(\exists t_j \in T_{PRC}) \wedge (s_j \in Exs_S(T_C)) \wedge (p_j \in Exs_P(T_C)) \wedge (o_j \in Exs_O(T_C)) \quad (15)$$

Условия фильтрации (14 и 15) могут описываться на языках запросов *SPARQL* или *SERQL*, которые эффективно обрабатываются сервером БЗ.

Пример шаблона запроса на языке *SERQL* для фильтрации возможных кандидатов некоторого объекта S по заданным наборам триплетов показан ниже.

```
/*Запрос на фильтрацию возможных кандидатов по набору T_K */
SELECT S FROM
{S} P {O}
WHERE P IN Exs_P(T_K) AND O IN Exs_O(T_K)
USING NAMESPACE
…
/*Запрос на фильтрацию возможных кандидатов по набору T_C */
/* CONTEXT S – триплеты контента объекта S */
SELECT S FROM CONTEXT S
{Sc} Pc {Oc}
WHERE Sc IN Exs_S(T_C) AND Pc IN Exs_P(T_C) AND Oc
IN Exs_O(T_C) USING NAMESPACE …
```

Рис. 2. Составление триплетов для поиска

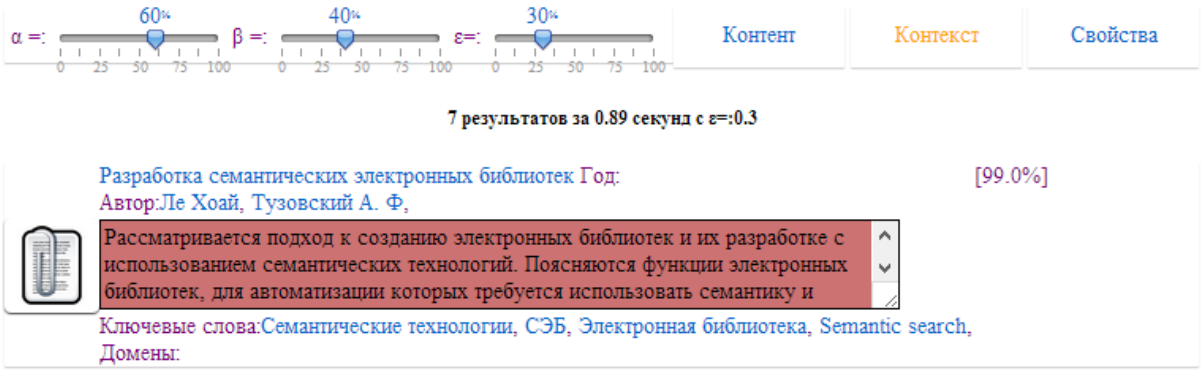


Рис. 3. Интерфейс категоризации или рекомендации на основе метаописаний

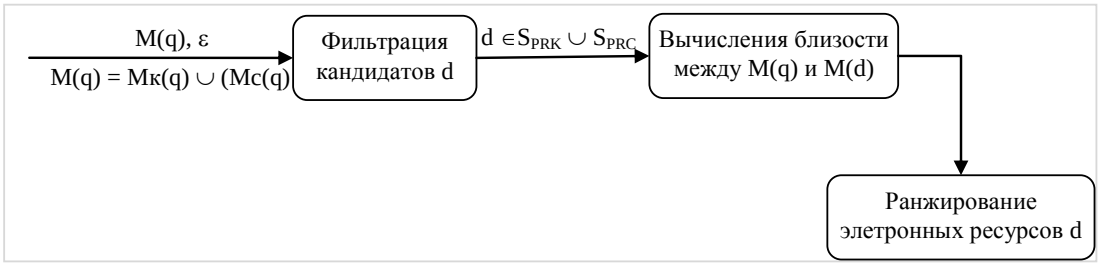


Рис. 4. Схема процесса семантического поиска

	S	P	O	№
Мк(q)	?S	_:type	_:Document	1
	?S	_:hasKeyword	_:Semantic_Technology	2
	?S	_:hasKeyword	_:Semantic_Digital_Library	3
	?S	_:hasAuthor	_:Ле_Хоай	4
	?S	_:hasAuthor	_:Тузовский	5
	Количество триплетов контекста (k): 5			
Mc(d)	_:2013051402540193	_:title	РАЗРАБОТКА ЭЛЕКТРОННЫХ БИБЛИОТЕК НА ОСНОВЕ СЕМАНТИЧЕСКИХ ТЕХНОЛОГИЙ	1
	_:2013051402540193	rdf:type	_:Article	2
	_:2013051402540193	_:hasKeyword	_:Semantic_Technology	3
	_:2013051402540193	_:hasKeyword	_:Semantic_Digital_Library	4
	_:2013051402540193	_:hasKeyword	_:Digital_Library	5
	_:2013051402540193	_:hasKeyword	_:Semantic_Search	6
	_:2013051402540193	_:hasKeyword	_:Knowledge_System	7
	_:2013051402540193	_:hasAuthor	_:Ле_Хоай	8
	_:2013051402540193	_:hasAuthor	_:Тузовский	9
	Количество триплетов контекста (h): 9			

Рис. 5. Пример метаописаний q и d

Пример численных вычислений. Пример вычисления делается для семантического поискового запроса, показанного на рис. 2 и его первого результата. Их метаописания показаны на рис. 5.

Вначале вычисляются близости между $M(q)$ и $M(d)$ по формуле 8 где ε = 0.1 $\alpha = 1$ и $\beta = 0$, так как семантический поиск выполняется по контексту.

На следующем шаге значение Sim(M(q),M(d)) = Sim($M\kappa(q)$, $M\kappa(d)$) вычисляется по формуле 6 или 7. и с целью простоты пояснения используется формулы 6. Для заданных $M\kappa(q)$ и $M\kappa(d)$ значение k = 5 (число триплетов $M\kappa(q)$), а значение h = 9 (число триплетов $M\kappa(d)$). Тогда значение Sim(Мк(q), Мк(d)) вычисляется следующим образом:

$$\mathrm{Sim}(M\kappa(q), M\kappa(d)) = \frac{\sum_{t_i \in M\kappa(q)} \max\left(\mathrm{Sim}(t_i, M\kappa(d))\right)}{\left|M\kappa(q)\right|} =$$

$$\frac{\sum_{i=1}^{5} \max_{j=1\to 9}\left(\mathrm{Sim}(t_i, t_j)\right)}{5} = \frac{\sum_{i=1}^{5} \max_{j=1\to 8}\left(\mathrm{Sim}(p_i, p_j) \times \mathrm{Sim}(o_i, o_j)\right)}{5}$$

Для $i = 1$ и $j = 1\to 8$ величина $p(q)_1$ = rdf:type, а $p(d)_{1\to 8} \in$ {_:title,rdf:type,_:hasKeyword, _:hasAuthor}. Так как в используемой онтологической модели нет семантических отношений между предикатами $p(q)_1$ и $p(d)_{1\to 8}$ (но могут быть семантические отношения между другими предикатами), то max(Sim(t_1, $M\kappa(d)$) = $\max_{i=1,\, j=1\to 8}$ (Sim(p_i, p_j) × Sim(o_i, o_j) =(Sim(p_1, p_2) × Sim(o_1, o_2), так как Sim(p_i, p_j) = 0 при $j \neq 2$.

Тогда в данном случае получается следующая окончательная формула: max(t_1, $M\kappa(d)$) = Sim(rdf:type, rdf:type) × Sim(:_Document, _:Article) = Sim(:_Document, _:Article).

Для $i = 2$ и $j = 1\to 8$ получаем:

max(Sim(t_2, $M\kappa(d)$) = Sim(_:Semantic_Technology, _:Semantic_Technology) = 1.

Аналогичным образом выполняются вычисления и для $i = 3\to 5$.

В конечном результате значение Sim(Мк(q), Мк(d)) определяется следующим образом:

$$\mathrm{Sim}(M\kappa(q), M\kappa(d)) = \frac{Sim(\_:Document,\ \_:Article) + 1 + 1 + 1 + 1}{5},$$

$$= \frac{Sim(\_:Document,\ \_:Article) + 4}{5}$$

Для определения величины Sim(_:Document, _:Article) используется граф семантических отношений GO (рис. 6). Здесь символы «_:» обозначают некоторое пространство имен понятий.

В показанном фрагменте графа *GO* используются четыре семантических отношений:

subClassOf (*подКласс*), *subPropertyOf* (*ПодСвойство*), *hasBroader* (Уже) и *hasNarrower* (*Шире*). Отметим, что все эти предикаты отображают таксономические отношения (на практике могут использоваться и другие предикаты). В таксономической иерархии имеются два отношения, которые либо обобщают (*subClassOf*, *subPropertyOf*, *hasBroader*), либо детализируют (*hasNarrower*) понятия.

Предпочтение всегда отдаётся детализации, это означает, что при поиске некоторого понятия более важными являются те понятия, которые детализуют рассматриваемое понятие. В связи с этим можно задать значения предикатом следующим образом:

- Если предикат p описывает отношение в направлении детализации, то $pv > 0.5$. Например, в данной работе pv(*hasBroader*) = 0.8.
- Если предикат p описывает отношение в направлении обобщения, то $pv < 0.5$. Например, в данной работе pv(*hasNarrower*) = 0.4.

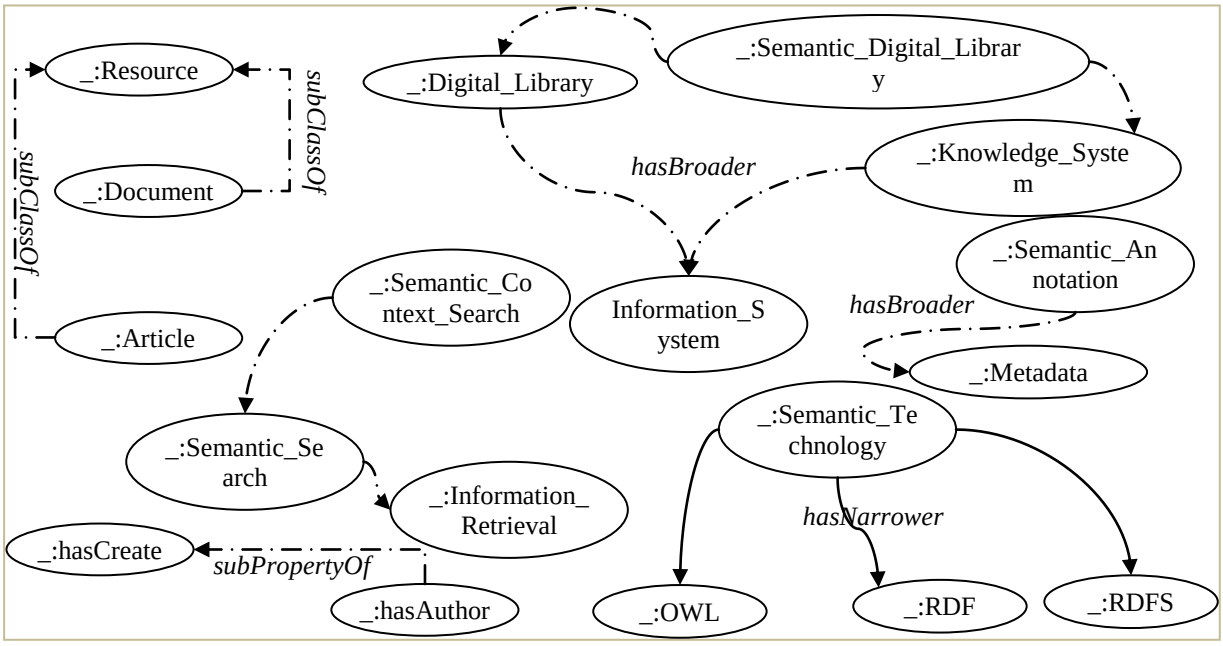


Рис. 6. Фрагмент графа семантических отношений GO

Все значения веса предикатов могут быть заданы специалистом, поддерживающим работу СЭБ, в соответствии с его пониманием онтологии предметной области.

В рассматриваемом случае *pv*(subClassOf) = 0.7 в направлении детализации, а в направления обобщения *pv*(*subClassOf*) = 0.3, а это значит, что Sim(_:*Resource*, _:*Article*) = 0.7, а Sim (_:Resource; _:Article)=0.3 и тогда

$$\mathrm{Sim}(M\kappa(q), M\kappa(d)) = \frac{Sim(\_:Document,\ \_:Article) + 4}{5}$$

$$= \frac{0,3*0,7+4}{5} = 0,842.$$

Для контентного семантического поиска процессы вычисления выполняются аналогично. Однако при контентном поиске для вычисления близости между триплетами контента учитывается ещё и близость между субъектами триплетов.

5 Заключение

Повышение эффективности электронных библиотек в значительной степени связано с использованием в их работе описания семантики электронных ресурсов. Для создания таких ЭБ требуется решение целого комплекса новых задач. В данной статье предложены методы описания семантики ЭР и вычисления семантической близости, рассмотрена реализация стандартных функций электронных библиотек с их использованием. Выполнение тестирования показало высокие результаты со средними значениями критерий: Точность = 100%, Полнота = 94%.

Литература

[1] Тузовский, А. Ф. Разработка семантических электронных библиотек / А. Ф. Тузовский, Х. Х. Ле // Доклады Томского государственного университета систем управления и радиоэлектроники. – 2011 – №. 2 (24) – С. 195–199.

[2] Ле Х. Х. Разработка электронных библиотек на основе семантических технологий // Научно–технический вестник Поволжья. – Казань, 2012. №. 3. С. 138–145.

[3] Тузовский А.Ф. Формирование семантических метаданных для объектов системы управления знаниями. //Известия Томского политехнического университета. 2007. Т. 310. №. 3. С. 108 – 112.

[4] Нгуен, Б. Н. Модели и методы поиска информационных ресурсов с использованием семантических технологий: дис. канд. техн. наук / Нгуен Ба Нгок. – Томск, 2012. – 198с.

[5] Hendler, A. J. Handbook of Semantic Web Technologies. – Springer, 2011.

[6] OWL Web Ontology Language Overview // Доступ осуществлен 03.04.2013 по адресу http://www.w3.org/TR/owl-features/.

[7] OWL:inverseOf // Доступ осуществлен 03.04.2013 по адресу http://www.infowebml.ws/rdf-owl/inverseOf.htm.

[8] SPARQL 1.1 Federated Query // Доступ осуществлен 03.04.2013 по адресу *http://www.w3.org/TR/sparql11-federated-query/*.

[9] System documentation for Sesame 2.x // Доступ осуществлен 03.04.2013 по адресу http://www.openrdf.org/doc/sesame2/system/.

[10] Тузовский, А. Ф. Онтолого-семантические модели в корпоративных системах управления знаниями: дис. д-тр. тех. наук / А. Ф.

Тузовский. – Томск, 2007. – 342с.
[11] Vector space model // Доступ осуществлен 26.07.2013 по адресу http://en.wikipedia.org/wiki/Vector_space_model.
[12] Хоай, Л. Программная система «SemDL – система управления хранилищем электронных ресурсов с использованием семантических технологий» / Ле Хоай, А.Ф. Тузовский // Свидетельство о государственной регистрации программы для ЭВМ № 2013613266. М.: Федеральная служба по интеллектуальной собственности (Роспатент). – 2013.
[13] Facebook graph search // Доступ осуществлен 03.04.2013 по адресу http://en.wikipedia.org/wiki/Facebook_Graph_Search.
[14] Thomas Paul. The Lucene Search Engine // Доступ осуществлен 12.05.2013 по адресу http://www.javaranch.com/journal/2004/04/Lucene.html.
[15] Welcome to Apache Lucene // Доступ осуществлен 12.05.2013 по адресу http://lucene.apache.org/ .
[16] Java Graph Visualization Library // Доступ осуществлен 12.05.2013 по адресу http://www.jgraph.com/jgraph.html.
[17] JGraph Diagram Component // Доступ осуществлен 12.05.2013 по адресу http://sourceforge.net/projects/jgraph/.
[18] JGraphT Visualizations via JGraph // Доступ осуществлен 12.05.2013 по адресу http://jgrapht.org/visualizations.html.
[19] The Sesame API // Доступ осуществлен 12.05.2013 по адресу http://www.openrdf.org/doc/sesame/users/ch07.html.
[20] Sesame distribution // Доступ осуществлен 12.05.2013 по адресу http://sourceforge.net/projects/sesame/files/Sesame%202/.

Проблемы использования данных из облака LOD для обогащения контента научных баз данных и знаний

© З. В. Апанович © А. Г. Марчук
Институт систем информатики им. А.П. Ершова СО РАН,
Новосибирск
apanovich@iis.nsk.su mag@iis.nsk.su

Аннотация

В данной работе описаны проблемы, возникающие в процессе использования данных из облака LOD для обогащения контента научных баз данных и знаний и подходы к их решению. Эксперименты выполнялись при помощи набора инструментов, разработанного для упрощения анализа данных из разных наборов. В качестве тестовых примеров использовались данные открытого Архива СО РАН, и его онтология ОНС, а также различные наборы библиографических данных, структурированные при помощи онтологии AKT Reference ontology.

Введение

В связи с бурно развивающимся направлением Semantic Web и его новой ветвью LOD (Связанные Открытые Данные) в Интернете становятся доступными большие объемы информации, посвященной различным научным направлениям. Облако LOD содержит в настоящий момент более 28 миллиардов троек RDF. С одной стороны, эти данные могут быть использованы для обогащения имеющихся семантических баз данных, с другой стороны, имеющиеся базы данные могут быть также полезны для уточнения информации, хранящейся в облаке LOD.

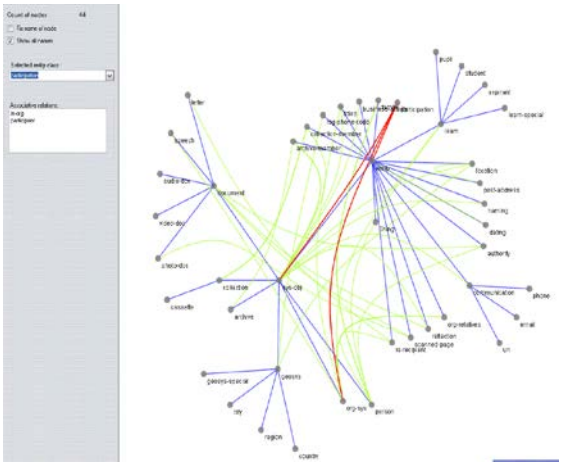
В работе [18] предложена четырехшаговая стратегия интеграции Связанных Данных в приложения. Помимо проблем, специфических для конкретного приложения, требуется решить проблему доступа к связанным данным (1), проблему нормализации словарей (2), установления идентичности сущностей (3) и фильтрации данных (4). Способы решения этих проблем варьируют в диапазоне от полностью ручных до автоматизированных [4, 8, 10, 13, 14, 19]. При этом многие проблемы, такие, как проблема обработки данных большого объема, проблема установления соответствия между онтологиями, а также проблема объединения данных из разных наборов «еще находятся в детском состоянии» [19]. С другой стороны, проблема (1) может быть решена при помощи запросов SPARQL 1.1 [8]. Проблема (2) может быть решена как при помощи сложных запросов SPARQL 1.1. Проблема (3) может быть частично решена при помощи таких полуавтоматических инструментов, как SILK [10] или LIMES [13] совместно с использованием запросов SPARQL. Наконец, проблема (4) также может быть решена при помощи запросов SPARQL 1.1. Поскольку практически каждая из проблем может быть решена при помощи подходящего набора SPARQL-запросов, мы расширили разработанную ранее программу визуализации онтологий средствами построения SPARQL-запросов и генерации результатов как в текстовом виде, так и в виде графа. SPARQL- запросы могут быть сгенерированы также на основе нашей визуализации одной или двух онтологий. В качестве тестовых примеров использовались онтология ОНС и данные открытого Архива СО РАН [20], а также AKT Reference ontology [1], при помощи которой структурированы различные наборы библиографических данных. В работе сравнивается их структура, и обсуждается стратегия установления связей между наборами данных, описанных при помощи этих онтологий.

1 Визуализация онтологий для исследования семантических систем

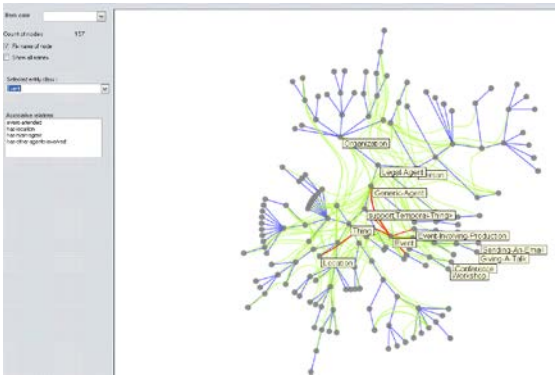
В настоящий момент в ИСИ СО РАН выполняется проект, направленный на интеграцию баз данных, разработанных в ИСИ СО РАН с данными мирового сообщества. С этой целью изучаются базы данных облака Linked Open Data [4] и выясняются возможности интеграции с ними систем, разработанных в ИСИ СО РАН [20], в

Труды 15-й Всероссийской научной конференции «Электронные библиотеки: перспективные методы и технологии, электронные коллекции» — RCDL-2013, Ярославль, Россия, 14-17 октября 2013 г.

частности, научной информации из открытого архива и фотоархива СО РАН [http://soran1957.iis.nsk.su/pa2/Home/Portrait?id=c_do 1000663]. Ее основное наполнение составляют документы, посвященные различным событиям СО РАН, начиная с 1957 года. В базе имеется также структурированная информация о людях, отраженных в документах, научных организациях, и важнейших событиях в жизни СО РАН, в частности, о научных конференциях. Структура Открытого архива организована при помощи Онтологии Неспецифических Сущностей (ОНС), описанной в OWL-формате и содержащей 44 класса.



(а)



(б)

Рис. 1. (а) Классы и отношения онтологии ОНС, (б) классы и отношения AKT reference ontology.

На Рис. 1(а) показано изображение онтология ОНС открытого архива СО РАН, построенное нашей программой визуализации [21]. Прямолинейные ребра изображают таксономию, задаваемую отношениями класс-подкласс. Криволинейные ребра изображают отношения типа *owl:ObjectProperty*. При выборе одного из классов в поле “Selected entity class” на панели визуализации графа высвечиваются все отношения, описанные в онтологии как *owl:ObjectProperty*, а в нижнем поле выдается список отношений этого класса (*owl:DatatypeProperty* и *owl:ObjectProperty*). При выборе элемента этого списка соответствующие ребра высвечиваются в окне визуализации. Это свойство визуализации весьма существенно для понимания незнакомой онтологии. Например, при выборе в онтологии такого класса как *онс:participation* высвечиваются ребра, соединяющие классы *онс:person, онс: participatio*n и *онс:org-sys*. Эта визуализация демонстрирует специфическую особенность онтологии ОНС, состоящую в том, что многие сущности, обычно описываемые как отношения, в данной онтологии описаны как классы, компенсируя отсутствие атрибутов у отношений в формате RDF.

Для сравнения, на Рис. 1(б) показаны классы и отношения AKT reference ontology [1], которая в облаке LOD используется для описания многих библиографических порталов, таких как DBLP, Citeseer, ACM, IEEE и др. Часть данных этих порталов представлены в облаке Open Linked Data[5, 7]. Она содержит 157 классов.

Попытки установления соответствия между этими онтологиями при помощи одной из лучших программ выравнивания AgreementMaker [6] оказались неудачными как из-за существенных лексических и структурных различий между рассматриваемыми онтологиями, так и из-за специфических особенностей онтологии ОНС, обсуждаемых ниже. Единственное очевидное соответствие наблюдается между классами *онс:person* и *akt:Person*. Остальные связи гораздо менее очевидны. Рассмотрим, например, класс *онс:participation*. В онтологии открытого архива этот класс используется для описания фактов работы персон в различных организациях, а также фактов участия в различных мероприятиях, например, конференциях. Этот класс связан отношением *онс:participant* с классом *онс:person* и отношением *онс:in-org* с классом *онс:org-sys*. Класс *онс:org-sys* используется как для описания различных организаций, так и для описания мероприятий, например, конференций. В AKT Reference ontology эти же самые факты могут быть описаны несколькими способами. Это может быть отношение *akt:works-for* между экземпляром класса *akt:Employee* и экземпляром класса *akt:Organization* , отношение *akt:has-affiliation* между экземпляром класса *akt:Person* и экземпляром класса *akt: Organization* , а также отношения *akt:has-main-agent, akt:has-other-agents-involved* между экземплярами классов *akt:Event* и *akt:Generic-Agent*. Из-за наличия в онтологии ОНС таких классов как *онс:participation*, при установлении соответствия между онтологиями возникает систематическая потребность в генерации экземпляров классов, которых до этого не было ни в одной из онтологий.

2 Эксперименты по выравниванию онтологий

Рассмотрим для определенности случай генерации экземпляра класса *онс:participation* по отношению *akt:has-affiliation* между экземпляром класса *akt:Employee* и экземпляром класса *akt:Organization*. Для пополнения открытого архива информацией о местах работы из одного библиографических порталов нам потребуется сначала установить соответствие между экземплярами классов *akt:Person* и *онс:person*, *akt-Organization* и *онс:org-sys*, а затем для каждого факта наличия отношения *akt:has-affiliation* между экземплярами классов *akt:Person* и *akt-Organization* потребуется сгенерировать новый экземпляр класса *онс:participation*, а также связать его отношением *онс:in-org* с соответствующим экземпляром класса *онс:org-sys*, и отношением *онс:participant* с соответствующим экземпляром класса *онс:person*. При обратной трансляции нам потребуется генерировать отношения akt:works-for соответствующие экземплярам класса *онс:participation.*

Поскольку в онтологии ОНС имеется достаточно много классов, аналогичных классу *онс:participation*, систематически возникает необходимость устанавливать соответствие между различными группами классов и отношений этих двух онтологий. А именно, необходимо установить соответствие между группой вида "Class 1-relation1-Class2" онтологии AKT Reference ontology и одной или несколькими группами вида "Class3- relation2-Class4-relation3-Class5" онтологии ОНС. При этом между объектами классов Class1 и Class3 следует установить связи типа *owl:sameAs* также как и для классов Class 2 и Class5. Помимо этого, необходимо сгенерировать экземпляр класса Class4 для каждой тройки <Class1:instance1, relation1, Class2:instance2>. Такая трансляция может быть осуществлена при помощи запроса SPARQL 1.1. Упрощенная версия этого запроса имеет следующий вид:

```
PREFIX iis:<http://iis.nsk.su#>
PREFIX akt:<http://www.aktors.org/ontology/portal#>
PREFIX
akts:<http://www.aktors.org/ontology/support#>
CONSTRUCT {
_:p a iis:Class4.
_:p iis:relation2 ?instance1.
_:p iis:relation3 ?instance2.
}
WHERE {
        ?instance1 akt:relation1  ?instance2.
        ?instance1 a akt:Class1.
        ?instance2 a akt:Class2.
         }
```

Для упрощения задачи нами разработана программа, которая позволяет генерировать SPARQL-запросы на основе визуализации онтологии. Пример установления такого соответствия показан на Рис. 2. Сначала в интерактивном режиме устанавливается соответствие между двумя наборами классов и отношений, а затем автоматически генерируется шаблон SPARQL-запроса, осуществляющий трансляцию данных.

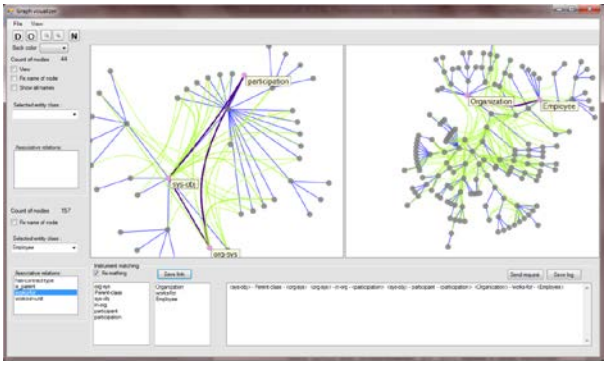


Рис. 2 Интерактивное установление соответствия между классами и отношениями двух онтологий.

3 Проблема установления идентичности сущностей

Как уже было сказано выше, существенным моментом обогащения одной базы знаний при помощи другой является этап установления идентичности сущностей в наборах данных LOD и данных открытого архива, то есть, генерация отношений вида *owl:sameAs.* Рассмотрим следующий пример. В Открытом архиве имеется экземпляр класса *онс:person*, описывающий бывшего директора ИСИ СО РАН В.Е Котова:

```
<person rdf:about="piu_200809052136">
    <name xml:lang="ru">Котов Вадим
Евгеньевич</name>
    <name xml:lang="en">Kotov, Vadim
Yevgenievich</name>
    <from-date>1938-07-23</from-date>
    <sex>m</sex>
  </person>
```

Также, в Открытом архиве СО РАН имеется достаточно подробная информация о его местах работы как в различных организациях СО РАН, так и в США. При этом отсутствует информация о его публикациях. С другой стороны, достаточно много информации о публикациях В.Е. Котова содержится в различных наборах данных облака LOD таких как: acm.rkbexplorer.com, dblp.rkbexplorer.com, citeseer.rkbexplorer.com. Но в этих наборах данных нет информации о местах работы В.Е. Котова, присутствующей в Открытом архиве. Для взаимовыгодного обмена данными надо, прежде

всего, связать отношением *owl:sameAs* экземпляры наборов данных из Открытого архива и LOD. Первая проблема связана с тем, что хоть имена персон и описаны при помощи одного и того же атрибута *akt:full-name* , этот атрибут может иметь разные значения не только в разных наборах облака LOD, но и в одном и том же наборе. Например, в наборе данных http://acm.rkbexplorer.com , в качестве свойства *akt:full-name* используются следующие идентификаторы: Vadim E. Kotov, V. Kotov, Vadim Kotov, V.E. Kotov. При этом каждому из этих имен соответствует отдельный идентификатор персоны, поэтому если мы строим запрос о публикациях, выдается по одной-две публикации соответствующей каждой из этих (РАЗНЫХ!) персон.

В настоящий момент все эти данные собираются в полу-автоматическом режиме при помощи программы SILK[10], на основе регулярных выражений и сравнения лексической близости соответствующих идентификаторов. Понятно, что эта процедура не гарантирует нам того, что мы не объединили вместе информацию об однофамильцах.

Наши эксперименты с полнотекстовыми версиями документов показали, что, во-первых, в них часто указываются места работы авторов, которые можно было бы сравнить с имеющимися в Открытом архиве списком мест и дат работы для каждой персоны. Во-вторых, авторы часто ссылаются на свои прежние публикации, что позволяет связывать в одну цепочку работы одного и того же автора. К сожалению, на сайтах указанного набора эта информация представлена неполно. Редко указаны места работы персон, а их временные границы не указаны совсем. Что же касается списков цитирования, то эта информация на данный момент тоже отражена неполно. Для многих персон из Открытого Архива, редко имеется информация более чем о двух ссылках из списка публикаций каждой из статей. Виду неполноты имеющейся информации планируется в дальнейшем извлекать информацию из полнотекстовых версий публикаций.

Для включения данных о публикациях Котова В.Е. в контент открытого архива, необходимо выполнить следующие трансформации (что опять же связано с различиями в онтологическом строении этих наборов данных). Во-первых, для каждого экземпляра класса *akt: publication-reference* следует создать экземпляр класса *онс:document,* а затем для каждого отношения *akt:has-author* AKT Reference ontology надо сгенерировать объект класса *онс:authorship* онтологии ОНС, после чего сгенерировать отношения *онс:adoc* и *онс:author,* связывающие индивид класса *онс:authorship* с соответствующими индивидами классов *онс:person* и *онс:document*. Эти трансформации выполняются при помощи SPARQL-запроса, аналогичного описанному выше.

4 Визуализация результатов Sparql-запросов для трансформации и анализа наборов данных

Основным инструментом исследования наполнения семантических систем в нашей системе является построение SPARQL-запросов и визуализация их результатов при помощи либо стандартного, либо специализированного алгоритма визуализации. Для этого нами реализована программа, позволяющая генерировать SPARQL-запросы к любой исследуемой семантической системе и получать результаты запроса, как текстовом виде, так и в виде графа. На Рис. 3 показано окно для ввода SPARQL-запросов и вывода результатов запроса в текстовом виде. Окно состоит из трех панелей. Панель справа показывает список основных классов и отношений исследуемой системы, верхняя панель предназначена для ввода SPARQL-запроса. В данный момент там имеется sparql-запрос на генерацию графа, ребрами которого являются отношения «коллега», соответствующие тому, что люди работают в одной организации. Для упрощения понимания запрос генерирует также ребро к вершине, соответствующей организации, в которой работают коллеги. Нижняя панель выдает результаты запроса в текстовом виде. Помимо этого, в верхней правом углу есть две кнопки, позволяющие либо сгенерировать результат запроса в виде графа, либо дополнительно осуществить кластеризацию одной из связных компонент полученного результата. Поскольку данная возможность визуализации предусмотрена для графов произвольной структуры, используется силовой алгоритм[8].

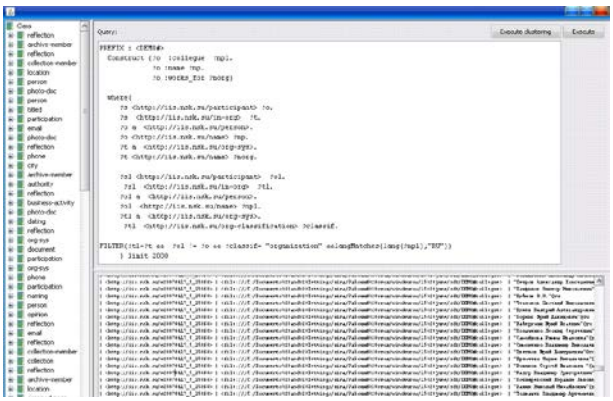
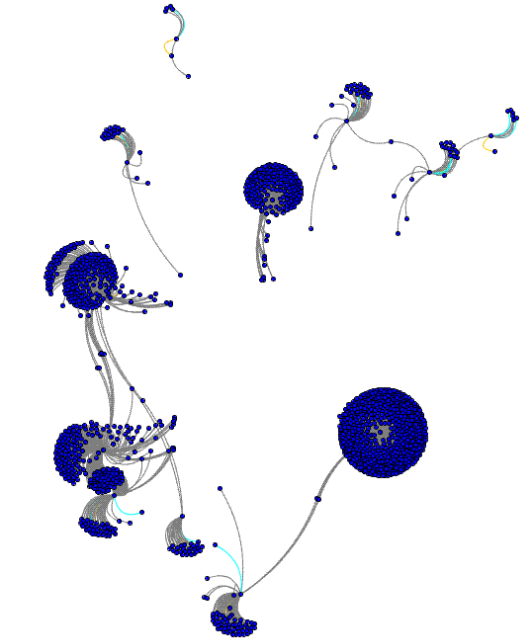
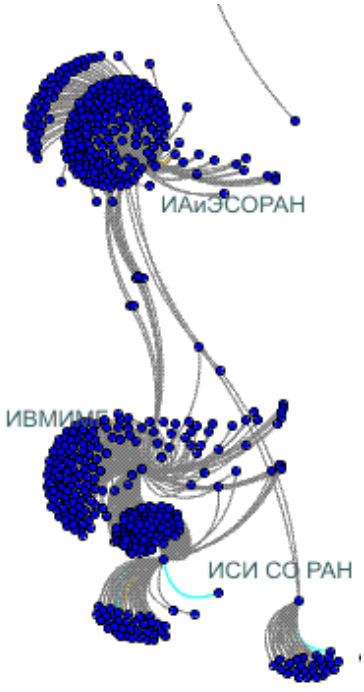


Рис. 3. Окно ввода SPARQL-запросов к исследуемой системе.

На Рис. 4(а) показан результат данного запроса в виде графа.



(а)



(б)

Рис. 4. Сети, выдаваемые в результате запросов к семантической системе открытого архива СО РАН.

Граф состоит из нескольких компонент связности. Люди сгруппированы вокруг организаций, в которых они работают или работали. Следует отметить, что поскольку указанный запрос не использовал фильтрацию по дате работы, некоторых людей ребра связывают с несколькими организациями. Так на Рис. 4(б) показан фрагмент изображения с Рис. 4(а), на котором видно, что часть коллег связано с ИСИ СО РАН, часть - с ИВМ и МГ, а часть – с обеими организациями. Эта промежуточная часть достаточно велика, поскольку ИСИ СО РАН был создан на базе одного из отделов ИВМ и МГ. Помимо просмотра данных из исследуемых баз данных эта компонента дает нам возможность визуализации и исследования сетей цитирования и соавторства и их кластеризацию, что важно для данного приложения.

Заключение

В данной работе рассмотрены проблемы обогащения научных баз знаний при помощи контента библиографических порталов из облака LOD и подходы к их решению. Сравниваются онтология ОНС и AKT Reference ontology и соответствие между наборами данных, основанных на этих онтологиях, устанавливается при помощи SPARQL-запросов, которые могут быть сгенерированы на основе визуализации онтологий.

Эксперименты показали, что для выравнивания онтологий недостаточно установления простых соответствий и могут потребоваться более сложные шаблоны. Также продемонстрировано, что обычные инструменты, применяемые для установления идентичности сущностей на основе метрик сходства, не позволяют различать авторов публикаций, являющихся тезками или даже однофамильцами. Для решения этой проблемы планируется использовать информацию о временных границах мест работы персон, описанных в Открытом Архиве СО РАН, а также методы изучения сетей самоцитирования.

Благодарности

Работа выполнена при финансовой поддержке РФФИ (проект № 11-07-00388).

Литература

[1] AKT ontology description: http://www.aktors.org/ontology.

[2] Alani, H. TGVizTab: An Ontology Visualization Extension for Protege. // Proceedings of Knowledge Capture (K-Cap'03),Workshop on Visualization Information in Knowledge Engineering, Sanibel Island, Florida, USA. 2003.

[3] Apanovich Z. V., Vinokurov P. S. An extension of a visualization component of ontology based portals with visual analytics facilities. // Bulletin of NCC .— Issue 31.— 2010.— pp. 17-28.

[4] Bizer, C., Heath, T. , Berners-Lee, T. Linked Data - The Story So Far. //Int. J. Semantic Web Inf. Syst., 5 (3). 2009. P. 1-22

[5] CiteSeer dataset : http://citeseer.rkbexplorer.com/.

[6] Cruz I. F., Stroe C., Caimi F., Fabiani A., Pesquita C., Couto F. M., Palmonari M. Using AgreementMaker to Align Ontologies for OAEI

2011. http://ceur-ws.org/Vol-814/oaei11_paper1.pdf

[7] DBLP dataset: http://dblp.rkbexplorer.com/.

[8] Erling O. How Virtuoso uses Relational Technology in its RDF Triple Store and SPARQL implementation. http://virtuoso.openlinksw.com/whitepapers/SPARQL%20RDF%20Store%20using%20SQL-ORDBMS.html

[9] Fruchterman T. M. J., Reingold E. M. Graph Drawing by Force-Directed Placement//Software - Practice and Experience, 1991, Vol. 21, N11, P. 1129-1164.

[10] Isele R., Jentzsch A., Bizer Ch. Silk Server - Adding missing Links while consuming Linked Data// 1st International Workshop on Consuming Linked Data (COLD 2010), Shanghai, November 2010.

[11] Katifori, A., Halatsis, C., Lepouras, G., Vassilakis, C., and Giannopoulou, E. (2007). Ontology Visualization Methods - a Survey. ACM Computing Surveys, 39(4).

[12] B. Kernighan and S. Lin, An efficient heuristic procedure for partitioning graphs, Bell System Technical Journal, 49 (1970), pp. 291- 307.

[13] Ngomo A.-C. N., Auer S.: LIMES - A Time-Efficient Approach for Large-Scale Link Discovery on the Web of Data. //IJCAI 2011: Proceedings of the 22nd International Joint Conference on Artificial Intelligence, Barcelona, Catalonia, Spain, July 16-22, 2011 pp. 2312-2317 .

[14] Oren, E., Delbru, R., Catasta, M., Cyganiak, R., Stenzhorn, H. and Tummarello, G.(2008). Sindice.com: a document-oriented lookup index for open linked data.// Int. J. Metadata, Semantics and Ontologies, Vol. 3, No. 1, pp. 37–52 (2008

[15] Pietriga, E. IsaViz. http://www.w3.org/2001/11/IsaViz .

[16] Sintek, M. Ontoviz tab: Visualizing Protégé ontologies. 2003 http://protegewiki.stanford.edu/wiki/OntoViz.

[17] Storey, M.-A. D. , Muller, H. A. Manipulating and documenting software structures using shrimp views. // Proc. of the Intl. Conf. on Software Mainten. — 1995.

[18] Schultz A. et al. How to integrate LINKED DATA into your application //Semantic technology & Business Conference, San Francisco, June 5, 2012. http://mes-semantics.com/wp-content/uploads/2012/09/Becker-etal-LDIF SemTechSanFrancisco.pdf.

[19] Tramp S., Williams H., Eck K., Creating Knowledge out of Interlinked Data: The LOD2 Tool Stack http://lod2.eu/Event/ESWC2012-Tutorial.html.

[20] Марчук А.Г., Марчук П.А. Особенности построения цифровых библиотек со связанным контекстом //Труды RCDL'2010- Двенадцатая Всероссийская научная конференция «Электронные библиотеки: перспективные методы и технологии, электронные коллекции» Казань, Казанский университет , 2010. — С. 19-23.

[21] Апанович З.В., Винокуров П.С., Кислицина Т.А. Гибкая подсистема визуализации онтологии и информационного наполнения порталов знаний на протяжении их жизненного цикла // Труды RCDL'2010 - Двенадцатая Всероссийская научная конференция "Электронные библиотеки: перспективные методы и технологии, электронные коллекции" Казань, Казанский университет, 2010.— С. 265-272.

Problems of using the LOD cloud datasets to enrich the content of scientific data and knowledge bases

Zinaida V. Apanovich, Alexander. G. Marchuk

This paper describes some problems arising during the use of the LOD cloud datasets to enrich the content of scientific knowledge bases and approaches to their solution. The experiments are carried out with the help of a toolkit intended to simplify analysis and integration of data from different datasets. The dataset of the Open Archive of the Russian Academy of Sciences, based on the ONS ontology, as well as various bibliographic datasets , structured by AKT Reference ontology, are used as test examples.

Использование графов горизонтальной видимости для выявления слов, определяющих информационную структуру текста

© Д. В. Ландэ
Институт проблем регистрации информации НАН Украины, НТУУ «Киевский политехнический институт», Украина, Киев
dwlande@gmail.com

© А. А. Снарский
asnarskii@gmail.com

© Е. В. Ягунова
С.-Петербургский гос. унив., С.-Петербург, Россия
iagounova.elena@gmail.com

Аннотация

Предлагается методика компактифицированного графа горизонтальной видимости для создания сети слов и выявления тех слов в тексте, которые определяют его информационную структуру. Исследованы свойства таких сетей слов, показано, что они являются безмасштабными, а также, что среди узлов с наибольшими степенями имеются слова, определяющие не только структуру связности текста, но и его информационную структуру.

Наряду с последовательным, «линейным» анализом текстов, построение сетей, узлами которых являются их элементы – слова или словосочетания, фрагменты естественного языка, позволяет выявлять структурные элементы текста, без которых он теряет свою связность. При этом актуальной является задача определения того, какие из важных структурных элементов оказываются также информационно-значимыми, определяющими информационную структуру текста. Такие элементы могут использоваться также для идентификации еще не достаточно четко определенных компонент текста, таких как коллокации, сверхфразовые единства [1], например, при поиске подобных фрагментов в различных текстах [2].

Известно несколько подходов к построению сетей из текстов, так называемых сетей слов (Language Network), и различные способы интерпретации узлов и связей, что приводит, соответственно, к различным видам представления таких сетей. Узлы могут быть соединены между собой, если соответствующие им слова стоят рядом в тексте [3, 4], принадлежат одному предложению или абзацу [5], соединены синтаксически [6, 7] или семантически [8, 9].

Труды 15-й Всероссийской научной конференции «Электронные библиотеки: перспективные методы и технологии, электронные коллекции» — RCDL-2013, Ярославль, Россия, 14-17 октября 2013 г.

В рамках теорий цифровой обработки сигналов (Digital Signal Processing) и сложных сетей (Complex Network) [10, 11] предложено несколько методов построения сетей на основе временных рядов, среди которых можно назвать несколько методов построения графов видимости (см. обзор [12]), в частности, так называемый граф горизонтальной видимости (Horizontal Visibility Graph – HVG) [13,14]. Эти подходы также позволяют строить сетевые структуры на основании текстов, в которых отдельным словам или словосочетаниям некоторым специальным образом поставлены в соответствие числовые весовые значения. В качестве функции, ставящей в соответствие слову число, можно рассматривать, например, порядковый номер уникального слова в тексте, длину слова, «вес» слов в текстах, общепринятую оценку TFIDF (в каноническом виде, равную произведению частоты слова в фрагменте текста – term frequency – на двоичный логарифм от величины, обратной количеству фрагментов текста, в которых это слово встретилось – inverse document frequency) или ее варианты [15, 16], а также другие весовые оценки.

В качестве весовой оценки TFIDF из полного текста, состоящего из N слов, текст разбивается на фрагменты, содержащие заданное количество слов M (например, $M = 500$). Затем для каждого слова i, входящего в текст, подсчитывается количество фрагментов $df(i)$, в которые это слово входит, а также общее количество вхождений данного слова i в текст – $n(i)$. После этого по формуле

$$tfidf(i) = \frac{n(i)}{N} \log\left(\frac{N}{M \times df(i)}\right)$$

рассчитывается среднее значение TFIDF весовой оценки каждого слова.

При построении сетей слов в данной работе также будет использована дисперсионная оценка важности слов [17], которая реализуется следующим образом: пусть текст состоит из N слов ($n = 1,...N$, n – порядковый номер слова в тексте, позиция слова). Некоторое слово, например A, обозначается как A_k^n, где индекс $k = 1,2,...,K$ – номер появления данного слова в тесте, а n – позиция данного слова в тексте. Например, A_3^{50}

означает, что на 50-й позиции текста находится слово A, которое встретилось третий раз.

Интервал между последовательными появлениями слова при таких обозначениях будет величина $\Delta A_k = A_{k+1}^m - A_k^n = m - n$, где на m-м и n-м позициях в тесте находится слово A, которое встретилось $k+1$-й и k-й разы.

Предложенная в [27] дисперсионная оценка рассчитывается как

$$\sigma_A = \frac{\sqrt{\langle \Delta A^2 \rangle - \langle \Delta A \rangle^2}}{\langle \Delta A \rangle},$$

где: $\langle \Delta A \rangle$ – среднее значение последовательности $\Delta A_1, \Delta A_2, ..., \Delta A_K$, $\langle \Delta A^2 \rangle$ – последовательности $\Delta A_1^2, \Delta A_2^2, ..., \Delta A_K^2$, K –количество появления слова A в тексте.

По сути, дисперсионная оценка позволяет отделить слова, встречающиеся в тексте относительно равномерно (для равномерно распределенных слов эта оценка равна нулю), от слов, распределенных неравномерно. Т.е. это оценка различительной, дискриминантной силы слов, в частности, для информационного поиска. Идея дисперсионной оценки очень близка к TFIDF, при этом менее распространена, однако более корректно применима к полным единичным текстам, а не к массивам текстов, как TFIDF.

В отличие от остальных рядов, изучаемых в рамках цифровой обработки сигналов, ряды из цифровых значений, соответствующих словам, преобразуются в графы горизонтальной видимости, в которых узлам соответствуют не только цифровые значения, но сами слова, выражающие определенное смысловое значение.

Сеть слов с использованием алгоритма горизонтальной видимости строится в три этапа. На первом на горизонтальной оси отмечается ряд узлов, каждый из которых соответствует словам в порядке появления в тексте, а по вертикальной оси откладываются весовые численные оценки (визуально – набор вертикальных линий, см. рис.1).

На втором этапе строится традиционный граф горизонтальной видимости [21]. Для этого между узлами существует связь, если они находятся в «прямой видимости», т.е. если их можно соединить горизонтальной линией, не пересекающей никакую другую вертикальную линию. Этот (геометрический) критерий можно записать, согласно [15,16] следующим образом: два узла (слова) слова, например, B_3^n и C_7^m $(m = n+5)$ соединены связью, если (см. рис. 1) $\sigma_n, \sigma_m > \sigma_p$ для всех $n < p < m$.

Алгоритм построения можно представить удобным для вычисления способом. Так например, на рис. 1 для узла-слова A_1^{n+2} смежными в сети считаются слова B_3^n и C_1^{n+5} (и устанавливаются ребра-связи), такие что B_3^n – ближайшее слева от A_1^{n+2} слово, с дисперсионной оценкой $\sigma_n = \sigma_B$, превышающей дисперсионную оценку слова A $\sigma_{n+2} = \sigma_A$, а $C_7\,(m = n+5)$ – ближайшее справа от A_1^{n+2} слово, для которого $\sigma_{105} > \sigma_{102}$.

На третьем, заключительном этапе, полученная на предыдущем этапе сеть компактифицируется. Все узлы с данным словом, например словом A, объединяются в один узел (естественно, индекс и номер положения слова при этом исчезают). Все связи таких узлов также объединяются. Важно отметить, что между любыми двумя узлами при этом остается не более одной связи – кратные связи изымаются. В частности это означает, что степень (число связей) узла A не превышает суммы степеней $\sum_k A_k^n$. В результате получается новая сеть слов – *компактифицированный граф горизонтальной видимости* (КГГВ) – рис.2.

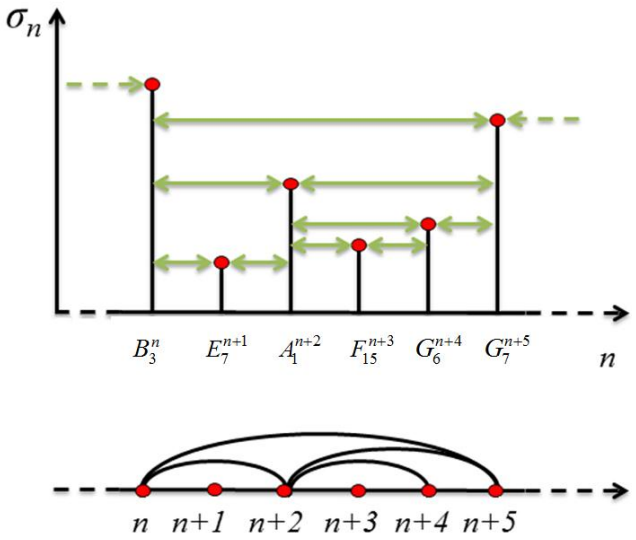


Рис. 1. Пример построения графа горизонтальной видимости

В качестве текстов при построении сетей слов в рамках данной статьи рассматриваются рассказы В. Астафьева «Ловля пескарей в Грузии», Ю. Бондарева «Река», И. Грековой «Без улыбок», Л. Петрушевской «Свой круг» и В. Пелевина «Проблема верволка в средней полосе». Следует, отметить, что авторами проводились подобные исследования на базе десятков других произведений, в том числе, значительно более объемных. Анализировались также законодательные акты Украины и России. Концептуальные результаты анализа при этом совпадали с приведенными ниже, поэтому остановимся на предложенных произведениях, как примерах.

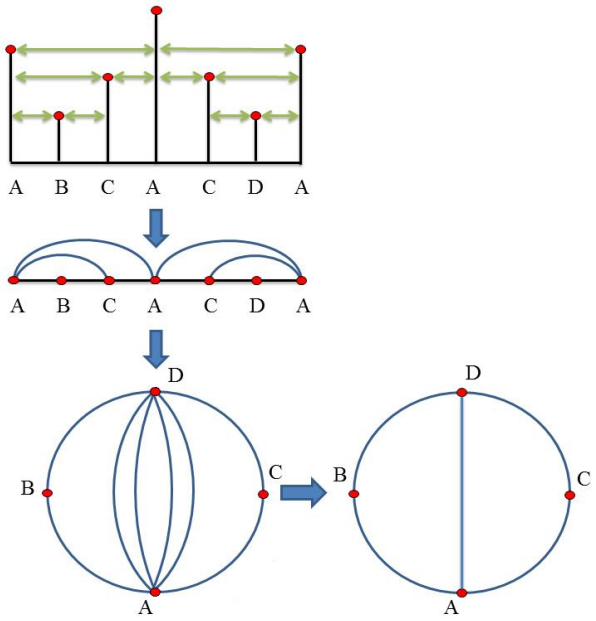


Рис. 2. Этапы построения компактификационного графа горизонтальной видимости

Для всех построенных КГГВ-сетей слов было определено распределение степеней узлов, которое оказалось близким к степенному ($p(k) = Ck^{\alpha}$), т.е. эти сети являются безмасштабными. Были проведены расчеты параметров сетей для всех рассмотренных литературных произведений. В результате оказалось, что для всех из них коэффициент α изменялся в диапазоне от –1 до –0,97 при относительно небольшой точности аппроксимации R^2 степенного распределения, которая повышается при увеличении длины текста. Для рассказов эта точность составила 0,5-0,7, а для сравнительно больших произведений, исследованных авторами, например, для романа М. Булгакова «Мастер и Маргарита» – 0,95.

В состав узлов с наибольшими степенями в для КГГВ-сетей, наряду с личными местоимениями и другими служебными словами (частицы, предлоги, союзы и т.д.), попали слова, определяющие информационную структуру текста [18, 19].

Для сравнения исследовано поведение простейших сетей языка, когда не первом этапе построения сети связываются соседние слова, входящие в текст, а на втором происходит компактификация сети. Очевидно, вес узлов в этой сети соответствует частоте встречаемости слов, а их распределение – закону Ципфа [20]. При этом самые большие степени имеют узлы, соответствующие словам с наибольшей частотой – союзам, предлогами и т.п., имеющим большое значение для связности текста, но малоинтересным с точки зрения информационной структуры.

Если обозначить Ψ – множество из N различных слов, соответствующих наиболее весомым узлам приведенной простейшей сети языка, а Λ – множество из слов, соответствующих наиболее весомым узлам КГГВ, то множество $\Omega = \Lambda \setminus \Psi$ соответствует информативным словам, имеющим, кроме того, важное значение и для связности текста. В Приложении приведены сопоставления 100 наиболее весомых узлов для трех рассматриваемых типов сетей слов по рассказам В. Астафьева «Ловля пескарей в Грузии», Л. Петрушевской «Свой круг» и В. Пелевина «Проблема верволка в средней полосе». Рассматривался случай N = 100, что было выбрано достаточно произвольно, с учетом того, что для рассматриваемых небольших по объему произведений важнейшие с точки зрения смысла слова попали в данный диапазон.

В частности, в КГГВ-сети по весовым значениям TFIDF, по рассказу В. Астафьева ««Ловля пескарей в Грузии» в состав множества Ω попали такие слова, как «Дядя», «Вася», «Собора», «Хозяин», «Грузии». В КГГВ-сети для этого же рассказа по весовым значениям, соответствующим дисперсионным оценкам, в состав множества Ω попали дополнительно такие слова, как «Пескаря», «Рыбы», «Храм», «Горы», «Витязь» и др.

При анализе рассказа Л. Петрушевской «Свой круг» в множество Ω попали такие слова, как «Алешка», «Отец», «Время», «Жизни», «Улице». В КГГВ-сети для этого рассказа по весовым значениям, соответствующим дисперсионным оценкам, в состав множества Ω попали дополнительно такие слова, как «Любви», «Ребенка», «Глаз», «Андрея».

В случае рассказа В. Пелевина «Проблема верволка в средней полосе» в состав множества Ω попали такие слова, как «Поляны», «Лапы», «Декан», «Дороги», «Машины», «Девочка». В КГГВ-сети для этого же рассказа по весовым значениям, соответствующим дисперсионным оценкам, в состав множества Ω попали те же слова, что и в предыдущем случае, и, кроме того, слова «Волки» и «Волков», играющие особую информационную роль в данном произведении.

Представления об информационной значимости рассматриваемых наборов слов, степени их важности для понимания смысла литературного произведения были подтверждены в ходе экспериментов с информантами. Так, для всех текстов были проведены эксперименты со стандартной инструкцией «Прочитайте текст. Подумайте над его содержанием. Выпишите 10-15 слов, наиболее важных для его содержания» (более 20 информантов для каждого текста) [21].

В результате проведенных исследований сетей:

1. Предложен алгоритм построения компактифицированного графа горизонтальной видимости (КГГВ).
2. На основе последовательности дисперсионных оценок слов текста и TFIDF, с помощью метода КГГВ, построены сети слов различных текстов.

3. Для литературных текстов среди узлов соответствующих КГГВ с наибольшими степенями присутствуют слова, не только обеспечивающие связность структуры текста, но и определяющие его информационную структуру, отражают семантику литературных произведений.
4. Алгоритм определения веса слов, базирующийся на дисперсионной оценке оказался более эффективным для определения информационно-значимых слов, играющих важное значение для структурной связности в литературных текстах, чем алгоритм TFIDF.

Литература

[1] Солганик Г. Я. Синтаксическая стилистика. Сложное синтаксическое целое. – 2-е изд., испр. и доп. – М.: Высш. шк. – 182 с. (1991).

[2] Broder A. Identifying and Filtering Near-Duplicate Documents, COM'00 // Proceedings of the 11th Annual Symposium on Combinatorial Pattern Matching. – P. 1-10 (2000).

[3] Ferrer-i-Cancho R., Sole R. V. The small world of human language // Proc. R. Soc. Lond. – B 268, 2261 (2001).

[4] Dorogovtsev S.N., Mendes J. F. F. Language as an evolving word web // Proc. R. Soc. Lond. – B 268, 2603 (2001).

[5] Caldeira S. M. G., Petit Lobao T. C., Andrade R. F. S., Neme A., Miranda J. G. V. The network of concepts in written texts // Preprint physics/0508066 (2005).

[6] Ferrer-i-Cancho R., Sole R.V., Kohler R. Patterns in syntactic dependency networks // Phys. Rev. E 69, 051915 (2004).

[7] Ferrer-i-Cancho R. The variation of Zipf's law in human language. // Phys. Rev. E 70, 056135 (2005).

[8] Motter A. E., de Moura A. P. S., Lai Y.-C., Dasgupta P. Topology of the conceptual network of language // Phys. Rev. E 65, 065102(R) (2002).

[9] Sigman M., Cecchi G. A. Global Properties of the Wordnet Lexicon // Proc. Natl. Acad. Sci. USA, 99, 1742 (2002).

[10] Strogatz S. H. Exploring Complex Networks // Nature. – 410. – P. 268-276 (2001).

[11] Albert R., Barabasi A.-L. Statistical mechanics of complex networks // Reviews of Modern Physics. – 74. – P. 47 (2002).

[12] Nunez A. M., Lacasa L., Gomez J. P., Luque B. Visibility algorithms: A short review // New Frontiers in Graph Theory, Y. G. Zhang, Ed. Intech Press, ch. 6. – P. 119 – 152 (2012).

[13] Luque B., Lacasa L., Ballesteros F., Luque J. Horizontal visibility graphs: Exact results for random time series // Physical Review E,– P. 046103-1–046103-11 (2009).

[14] Gutin G., Mansour T., Severini S. A characterization of horizontal visibility graphs and combinatoris on words // Physica A, – 390 – P. 2421-2428 (2011).

[15] Jones K.S. A statistical interpretation of term specificity and its application in retrieval // Journal of Documentation. – 28 (1). – P. 11–21 (1972).

[16] 16. Salton G., McGill M. J. Introduction to Modern Information Retrieval. – New York: McGraw-Hill. – 448 p. (1983).

[17] Ortuño M., Carpena P., Bernaola P., Muñoz E., Somoza A.M. Keyword detection in natural languages and DNA // Europhys. Lett, – 57(5). – P. 759-764 (2002).

[18] Черняховская Л.А. Смысловая структура текста и ее единицы // Вопросы языкознания. – № 6. – С. 118–126. (1983).

[19] Giora R. Segmentation and Segment Cohesion: On the Thematic Organization of the Text // Text. An Interdisciplinary Journal for the Study of Discourse Amsterdam. – 3. – № 2. – P. 155-181 (1983).

[20] Zipf G.K. Human Behavior and the Principle of Least Effort. – Cambridge, MA: Addison-Wesley Press. – 573 p. (1949).

[21] Ягунова Е.В. Эксперимент и вычисления в анализе ключевых слов художественного текста // Сборник научных трудов кафедры иностранных языков и философии ПНЦ УрО РАН. Философия языка. Лингвистика. Лингводидактика / Отв. ред. В.Т. Юнгблюд.– Пермь, 2010. – Вып. 1. – С. 85- 91.

The Use Of Horizontal Visibility Graphs To Identify The Words That Define The Information Structure Of The Text

D.V. Lande, A.A. Snarskii, E.V. Yagunova

A compactified horizontal visibility graph for the language network and identify the words that define the information structure of the text is proposed. It was found that the networks constructed in such way are scale free, and have a property that among the nodes with largest degrees there are words that determine not only a text structure communication, but also its informational structure.

Приложение. Сопоставление 100 наиболее весомых узлов сетей по рассказам*

1. В. Астафьев, «Ловля пескарей в Грузии»

Простейшая сеть

1	И	21	ОТАР	41	ТОЖЕ	61	СЕБЯ	81	ЛЮДЕЙ
2	В	22	ПОД	42	ТОЛЬКО	62	ЛИ	82	КОТОРЫЕ
3	НА	23	БЫЛО	43	ОТАРА	63	КУДА	83	ЕСТЬ
4	С	24	ОНИ	44	МНЕ	64	ЕМУ	84	ЕСЛИ
5	НЕ	25	ЕЩЕ	45	ВОЗЛЕ	65	БРАТЬЯ	85	ДОМА
6	ЧТО	26	ТАК	46	ВО	66	СЕРДЦЕ	86	ЧЕЛОВЕК
7	Я	27	МЫ	47	ТЫ	67	НАД	87	ВСЕГДА
8	ПО	28	ИЛИ	48	ДЛЯ	68	МЕНЯ	88	СОВСЕМ
9	ЗА	29	ЖЕ	49	ТУТ	69	ГЕЛАТИ	89	ПОТОМ
10	ИЗ	30	ДО	50	ГДЕ	70	БЕЗ	90	НАМ
11	ТО	31	КОГДА	51	РАЗ	71	ЧЕМ	91	ИМ
12	НО	32	ЭТО	52	ВРЕМЯ	72	НАС	92	ДАЖЕ
13	ОТ	33	БЫ	53	О	73	БЫЛА	93	БЫЛИ
14	ОН	34	НИ	54	ЧТОБЫ	74	ЗДЕСЬ	94	ЗЕМЛИ
15	ВСЕ	35	ДА	55	ВСЕХ	75	ВСЕГО	95	ВСЕМ
16	КАК	36	БЫЛ	56	ПОЧТИ	76	УЖ	96	ТОГО
17	ЕГО	37	МОЖЕТ	57	ЧТОБ	77	СРЕДИ	97	СТОЛОМ
18	А	38	ИХ	58	ЭТОТ	78	РЕЧКИ	98	ПРО
19	У	39	ВОТ	59	СО	79	НЕТ	99	ОДНАКО
20	К	40	УЖЕ	60	ШАЛВА	80	НАШЕЙ	100	ОДИН

КГГВ–TFIDF

1	И	21	ПОД	41	НАД	61	ЭТОТ	81	ДАЖЕ
2	В	22	БЫ	42	ТЫ	62	БЕЗ	82	ПОТОМ
3	Я	23	ЧТО	43	ВРЕМЯ	63	ВСЕХ	83	*РЕЧКИ*
4	ЗА	24	ВО	44	УЖЕ	64	ТУТ	84	**ДОМ**
5	НА	25	КОГДА	45	МНЕ	65	ЛИ	85	ЧТОБ
6	У	26	ТОЛЬКО	46	ЭТО	66	**ХОЗЯИН**	86	ПРО
7	ПО	27	О	47	НО	67	ВОТ	87	СРЕДИ
8	НЕ	28	НИ	48	ТО	68	НАШЕЙ	88	ТАКОЙ
9	ТАК	29	*ОТАРА*	49	**ДОМА**	69	СЕБЯ	89	СОВСЕМ
10	К	30	МЫ	50	**ДЯДЯ**	70	ГДЕ	90	РАЗ
11	С	31	БЫЛО	51	**ВАСЯ**	71	ТОГДА	91	НЕТ
12	ЕЩЕ	32	*БРАТЬЯ*	52	**СОБОРА**	72	КУДА	92	**ДОЖДЬ**
13	ОТ	33	ДО	53	СО	73	МЕНЯ	93	НАМ
14	МОЖЕТ	34	ИХ	54	КАК	74	КОТОРЫЕ	94	**ГРУЗИИ**
15	*ОТАР*	35	ОНИ	55	ДЛЯ	75	**ЗЕМЛИ**	95	МОЕГО
16	ИЗ	36	*ГЕЛАТИ*	56	БЫЛ	76	ЗДЕСЬ	96	**СЕРДЦЕ**
17	ЕГО	37	ВОЗЛЕ	57	ДА	77	ТОЖЕ	97	**ГОР**
18	ВСЕ	38	*ШАЛВА*	58	НАС	78	ЧТОБЫ	98	ЕСЛИ
19	ИЛИ	39	ЖЕ	59	ПОЧТИ	79	ЛИШЬ	99	БЫЛА
20	А	40	*СТОЛОМ*	60	ОН	80	ЧЕМ	100	**ЧЕЛОВЕК**

КГГВ–дисперсионная оценка

1	И	21	*ОТАР*	41	*ГЕЛАТИ*	61	БЕЗ	81	**ДРУГ**
2	В	22	ПОД	42	О	62	ВОЗЛЕ	82	**ДЕТЕЙ**
3	НА	23	НИ	43	МЕНЯ	63	ЛИ	83	НАМ
4	С	24	ЕЩЕ	44	ОНИ	64	СО	84	ТУТ
5	НЕ	25	КОГДА	45	НАД	65	ДА	85	ТОЖЕ
6	Я	26	КАК	46	ЭТОТ	66	СОВСЕМ	86	ЧЕМ
7	ЗА	27	ИЛИ	47	ЖЕ	67	**ДОМ**	87	**ПЕСКАРЯ**
8	ЧТО	28	ТЫ	48	**СОБОРА**	68	**ДОЖДЬ**	88	**ГОРЫ**
9	ПО	29	ВРЕМЯ	49	СЕБЯ	69	БЫЛ	89	РАЗ
10	ОТ	30	БЫЛО	50	ДО	70	ПРО	90	ПОТОМ
11	ВСЕ	31	**ДОМА**	51	*СТОЛОМ*	71	ТАКОЙ	91	ДАЖЕ
12	ЕГО	32	ЭТО	52	**ХОЗЯИН**	72	**ПЕСКАРЕЙ**	92	ГДЕ
13	ОН	33	ВО	53	*ШАЛВА*	73	НЕТ	93	СРЕДИ
14	У	34	**ДЯДЯ**	54	ТОЛЬКО	74	НАШЕЙ	94	ПРОТИВ
15	ТО	35	БЫ	55	ДЛЯ	75	ЗДЕСЬ	95	ЧТОБЫ
16	ИЗ	36	МЫ	56	ПОЧТИ	76	*РЕЧКИ*	96	ВСЕГО
17	К	37	МОЖЕТ	57	МНЕ	77	**ХРАМ**	97	**ВИТЯЗЬ**
18	А	38	**ВАСЯ**	58	ИХ	78	УЖЕ	98	ВСЕХ
19	ТАК	39	*ОТАРА*	59	**РЫБА**	79	**ТВОРЧЕСТВА**	99	ВОТ
20	НО	40	*БРАТЬЯ*	60	НАС	80	КОТОРЫЕ	100	КУДА

* Слова, присутсвующие в списке узлов КГГВ, но отсутствующие в списке узлов простейшей сети выделены жирным шрифтом. Наиболее информационно-значимые слова, также присутствующие и в топ-100 простейшей сети, выделены курсивом.

2. Л. Петрушевская, «Свой круг»

Простейшая сеть

1	И	21	СЕРЖ	41	ЖОРА	61	ЛИ	81	МАРИШУ
2	В	22	ЖЕ	42	БЫ	62	ВООБЩЕ	82	МАРИШЕ
3	НЕ	23	ТАК	43	ТУТ	63	ТЕПЕРЬ	83	ИХ
4	НА	24	МАРИША	44	СЕРЖА	64	СВОЕЙ	84	РАЗ
5	А	25	ИЗ	45	ДО	65	ОДИН	85	ПОСЛЕ
6	С	26	АНДРЕЙ	46	ВОТ	66	НАДЯ	86	МАРИШИ
7	ВСЕ	27	КОЛЯ	47	БЫЛИ	67	ЛЕТ	87	ИМ
8	КАК	28	КОГДА	48	ОТ	68	ГДЕ	88	ЭТОТ
9	Я	29	ВАЛЕРА	49	МЫ	69	ВСЕГДА	89	ТЫ
10	ЧТО	30	БЫЛ	50	ЛЕНКА	70	ТАМ	90	ТАНЯ
11	ОН	31	ТОЛЬКО	51	ПОТОМ	71	ОЧЕНЬ	91	ПОСКОЛЬКУ
12	У	32	УЖЕ	52	ВСЕХ	72	О	92	НИЧЕГО
13	ТО	33	ЕЩЕ	53	СО	73	АЛЕША	93	НАДО
14	ЭТО	34	ЕЕ	54	МНЕ	74	ВРЕМЯ	94	ДАЖЕ
15	НО	35	ОНИ	55	ДЛЯ	75	ТОЖЕ	95	БЕЗ
16	ЕГО	36	НИ	56	БУДЕТ	76	ТОГО	96	ТОТ
17	ПО	37	ЕМУ	57	СКАЗАЛА	77	СВОЮ	97	СКАЗАЛ
18	К	38	КОТОРЫЙ	58	МЕНЯ	78	СТАЛ	98	НАД
19	ЗА	39	ОНА	59	ЧТОБЫ	79	ПОД	99	ДОМА
20	БЫЛО	40	БЫЛА	60	СЕБЯ	80	МОЙ	100	АЛЕШУ

КГГВ-TFIDF

1	И	21	Я	41	ИЗ	61	ВСЕХ	81	**НОЧЬ**
2	В	22	ТАК	42	БЫЛА	62	СВОЕЙ	82	**ДВЕРЬ**
3	ОН	23	НИ	43	БУДЕТ	63	*МАРИШУ*	83	ЭТОТ
4	*АНДРЕЙ*	24	ЕМУ	44	*АЛЕША*	64	*АЛЕШУ*	84	ЧТОБЫ
5	*ВАЛЕРА*	25	К	45	ТО	65	ПРИ	85	СТАЛ
6	*КОЛЯ*	26	БЫЛИ	46	ТУТ	66	ВООБЩЕ	86	СПРОСИЛА
7	НЕ	27	ЖЕ	47	ЛИ	67	МОЙ	87	ЛЕТ
8	НА	28	БЫЛ	48	ВСЕГДА	68	ТОТ	88	ИМ
9	ЭТО	29	МЫ	49	ОТ	69	ЖИТЬ	89	БЕЗ
10	С	30	*ЖОРА*	50	*НАДЯ*	70	ТОГО	90	**АЛЕШКА**
11	*СЕРЖ*	31	КАК	51	ДО	71	ГДЕ	91	**УЛИЦЕ**
12	ПО	32	БЫ	52	ПОТОМ	72	ТАМ	92	ПОД
13	ОНА	33	У	53	ОДИН	73	СЕБЯ	93	**ОТЕЦ**
14	А	34	ЕЩЕ	54	НАС	74	МНЕ	94	**ВРЕМЯ**
15	ОНИ	35	БЫЛО	55	О	75	СО	95	КОТОРЫЙ
16	ЕЕ	36	СКАЗАЛА	56	МЕНЯ	76	ЗА	96	ТОЛЬКО
17	*ЛЕНКА*	37	*МАРИША*	57	*МАРИШИ*	77	НЕЕ	97	**ЖИЗНИ**
18	ЧТО	38	ВОТ	58	НО	78	ДЛЯ	98	СКАЗАЛ
19	КОГДА	39	*СЕРЖА*	59	ЕГО	79	ИЛИ	99	ТОЖЕ
20	ВСЕ	40	УЖЕ	60	ОЧЕНЬ	80	*ТАНЯ*	100	ИХ

КГГВ-дисперсионная оценка

1	И	21	ЭТО	41	БУДЕТ	61	СО	81	ОДИН
2	В	22	ЖЕ	42	ТЕ	62	*МАРИШУ*	82	СТАЛ
3	НЕ	23	ТАК	43	ДО	63	БЫЛА	83	МОЙ
4	А	24	К	44	ЕМУ	64	СВОЕЙ	84	**ЛЮБВИ**
5	Я	25	ЕГО	45	ВСЕГДА	65	ТОЛЬКО	85	ИМ
6	С	26	ЗА	46	*АЛЕШУ*	66	ВОТ	86	ГДЕ
7	НА	27	*МАРИША*	47	ЛИ	67	МНЕ	87	ДЛЯ
8	ВСЕ	28	*ЛЕНКА*	48	ОЧЕНЬ	68	БЫТЬ	88	ДАВНО
9	ТО	29	ЕЕ	49	**ОТЕЦ**	69	КОТОРЫЙ	89	ЧЕМ
10	ОН	30	ИЗ	50	УЖЕ	70	ПЕРЕД	90	**ВРЕМЯ**
11	*АНДРЕЙ*	31	НО	51	ТУТ	71	НИЧЕГО	91	СПРОСИЛА
12	У	32	*ЖОРА*	52	БЫЛ	72	**ГЛАЗ**	92	СКАЗАЛ
13	ЧТО	33	НИ	53	ОТ	73	ТЫ	93	**РЕБЕНКА**
14	*СЕРЖ*	34	МЫ	54	БЫ	74	ВСЕХ	94	**АЛЕШКА**
15	КАК	35	ОНА	55	*НАДЯ*	75	ПОТОМ	95	ПОД
16	*ВАЛЕРА*	36	*АЛЕША*	56	БЫЛИ	76	НАД	96	ТОГО
17	*КОЛЯ*	37	*СЕРЖА*	57	*МАРИШИ*	77	КТО	97	*ТАНЯ*
18	ОНИ	38	ЕЩЕ	58	О	78	СЕБЯ	98	**АНДРЕЯ**
19	ПО	39	КОГДА	59	РАЗ	79	ПРИ	99	**УЛИЦЕ**
20	БЫЛО	40	СКАЗАЛА	60	ДАЖЕ	80	ЖИТЬ	100	ПОЧЕМУ

3. В. Пелевин, «Проблема верволка в средней полосе»

Простейшая сеть

1	И	21	ЗА	41	ЕСЛИ	61	ПОД	81	ПЕРЕД
2	В	22	ТЫ	42	ГЛАЗА	62	НЕСКОЛЬКО	82	МОРДУ
3	НА	23	ОНА	43	ДО	63	ЕЕ	83	ЧУТЬ
4	ОН	24	ОТ	44	ЧТОБЫ	64	МНЕ	84	ЭТОТ
5	САША	25	ТАК	45	СЕЙЧАС	65	КАКОЙ	85	ЗДЕСЬ
6	НЕ	26	ЕЩЕ	46	О	66	СТАЛ	86	ПЕРЕД
7	ЧТО	27	КОГДА	47	ГДЕ	67	НЕГО	87	В
8	А	28	ТОЛЬКО	48	БЫЛ	68	ПОЧЕМУ	88	ТЕБЯ
9	ТО	29	ВОЖАК	49	ВОКРУГ	69	КТО	89	РЯДОМ
10	С	30	НИКОЛАЙ	50	ИЛИ	70	ЭТОГО	90	ДОРОГЕ
11	ПО	31	ПОТОМ	51	ВОТ	71	ОТВЕТИЛ	91	ЧЕГО
12	КАК	32	ОНИ	52	БЫЛИ	72	МОЖЕТ	92	ВВЕРХ
13	ЭТО	33	У	53	БЫЛА	73	ДА	93	ВРЕМЯ
14	ЕГО	34	СКАЗАЛ	54	ДЛЯ	74	БУДЕТ	94	ВО
15	К	35	ЖЕ	55	ТОЖЕ	75	ЧЕМ	95	СТАЯ
16	НО	36	ЕМУ	56	СЕБЯ	76	ПОДУМАЛ	96	ОЧЕНЬ
17	ИЗ	37	ВДРУГ	57	ЛЕНА	77	ЛЕС	97	ОПЯТЬ
18	БЫЛО	38	УЖЕ	58	ЧЕРЕЗ	78	ИХ	98	ОДНА
19	Я	39	БЫ	59	ТЕПЕРЬ	79	УВИДЕЛ	99	НИБУДЬ
20	ВСЕ	40	ВЫ	60	РАЗ	80	ПОСЛЕ	100	НЕТ

КГГВ-TFIDF

1	И	21	ТОЛЬКО	41	ЕМУ	61	КТО	81	ВРЕМЯ
2	Я	22	ВДРУГ	42	БЫЛ	62	*КОСТРА*	82	РЯДОМ
3	В	23	К	43	ЧТОБЫ	63	МНЕ	83	**МАШИНЫ**
4	БЫЛО	24	*ЛЕНА*	44	ЖЕ	64	ЕСЛИ	84	ПОГЛЯДЕЛ
5	ЭТО	25	БЫ	45	ГДЕ	65	ДЛЯ	85	СРАЗУ
6	С	26	ПОТОМ	46	ИЗ	66	**ЛАПЫ**	86	УВИДЕЛ
7	*ВОЖАК*	27	ТЕПЕРЬ	47	РАЗ	67	ИЛИ	87	*ЛЕС*
8	НА	28	ВСЕ	48	ТОЖЕ	68	ЖИЗНИ	88	ПОЧУВСТВОВАЛ
9	ТЫ	29	КАК	49	СЕЙЧАС	69	*МОРДУ*	89	ЧУТЬ
10	ЕГО	30	КАКОЙ	50	НО	70	ПОЧЕМУ	90	СТАЛ
11	ЧТО	31	ПО	51	ПОД	71	ПОДУМАЛ	91	**ДЕВОЧКА**
12	*НИКОЛАЙ*	32	КОГДА	52	ПОНЯЛ	72	ЗА	92	ПЕРЕД
13	ОНА	33	*ГЛАЗА*	53	НЕГО	73	ВОТ	93	БУДЕТ
14	ОН	34	У	54	ЕЩЕ	74	ОНИ	94	ИДТИ
15	СКАЗАЛ	35	ДО	55	ТАК	75	**ДЕКАН**	95	ВВЕРХ
16	*САША*	36	УЖЕ	56	*ДОРОГЕ*	76	**ДОРОГИ**	96	НАЗАД
17	НЕ	37	ОТ	57	СЕБЯ	77	ВО	97	ЕЕ
18	ТО	38	О	58	**ПОЛЯНЫ**	78	ОДНА	98	ЗАМЕТИЛ
19	ВЫ	39	БЫЛИ	59	НЕСКОЛЬКО	79	ЧЕРЕЗ	99	ТЕБЯ
20	А	40	БЫЛА	60	ОТВЕТИЛ	80	ЧЕМ	100	ЗДЕСЬ

КГГВ-дисперсионная оценка

1	И	21	ИЗ	41	НЕГО	61	*МОРДУ*	81	**ВОЛКОВ**
2	В	22	ОНА	42	ЕСЛИ	62	ОТВЕТИЛ	82	*СТАЯ*
3	ОН	23	УЖЕ	43	*ДОРОГА*	63	ЕМУ	83	РАЗ
4	*САША*	24	*КОСТРА*	44	ЧЕРЕЗ	64	ДЛЯ	84	МЫ
5	НА	25	СКАЗАЛ	45	БЫЛ	65	**ЛАПЫ**	85	ВВЕРХ
6	ТО	26	*ЛЕНА*	46	ОНИ	66	*ГЛАЗА*	86	ПРИ
7	НЕ	27	ЗА	47	*ЛЕС*	67	*ДОРОГЕ*	87	ПОД
8	ЭТО	28	ДО	48	ЖЕ	68	**ДЕВОЧКА**	88	ПОЧУВСТВОВАЛ
9	ЧТО	29	НО	49	У	69	ПОЧЕМУ	89	НАЗАД
10	С	30	ТОЛЬКО	50	БЫЛА	70	ИЛИ	90	ИХ
11	БЫЛО	31	ВЫ	51	ВО	71	**ДЕКАН**	91	ВАМ
12	Я	32	ВСЕ	52	О	72	ГДЕ	92	СЛОВО
13	ЕГО	33	ЕЩЕ	53	БУДЕТ	73	ТЕПЕРЬ	93	СЕЙЧАС
14	К	34	КОГДА	54	ОДНА	74	**ПОЛЯНЫ**	94	КТО
15	ПО	35	ПОТОМ	55	ЧТОБЫ	75	МИМО	95	ДРУГ
16	А	36	БЫ	56	БЫЛИ	76	ВОКРУГ	96	ВРЕМЯ
17	*ВОЖАК*	37	КАКОЙ	57	**ДОРОГИ**	77	ТАКОЕ	97	БУДТО
18	ТЫ	38	ОТ	58	ВОТ	78	НЕСКОЛЬКО	98	ЭТОТ
19	*НИКОЛАЙ*	39	МНЕ	59	ТОЖЕ	79	**МАШИНА**	99	**ВОЛКИ**
20	КАК	40	ВДРУГ	60	ТАК	80	НАОБОРОТ	100	**САШЕ**

Методы решения задачи автоматического выявления заимствований в структурированных научно-технических документах на основе их семантического анализа

© В.Н. Захаров
ИПИ РАН

© А. А. Хорошилов
ЦИТиС

Москва

vzakharov@ipiran.ru

a.a.horoshilov@mail.ru

Аннотация

В работе излагаются методы выявления заимствований в структурированных научно-технических документах, базирующиеся на семантическом анализе текстов. Существующие системы позволяют устанавливать заимствования только в случаях, если эти заимствования производятся путем копирования фрагментов текста без его изменения или с незначительными изменениями его структуры или лексического состава. Использование семантических методов анализа текстов позволяет выявить смысловую структуру текста и распознавать более сложные случаи преднамеренного изменения заимствованных текстов, например, установить случаи замены слов или словосочетаний их смысловыми инвариантами, изменения разбиения текста на предложения, перестановка фрагментов текста. Также в данной работе обосновывается необходимость выявлять в текстах и исключать из рассмотрения текстовые фрагменты, относящиеся к описанию структурных элементов документов (например, стандартные для всех документов заголовки).

1 Проблема выявления заимствований в текстах документов

1.1 Введение

В соответствии с Гражданским кодексом Российской Федерации государство обязано защищать объекты авторских прав от различных нарушений в этой сфере. Одним из наиболее частых нарушений является умышленное присвоение авторства на чужие результаты интеллектуальной деятельности, представленные в виде произведений науки, литературы или искусства. Часто такое присвоение авторства осуществляется путем заимствования чужого произведения или его части. Борьба с этими нарушениями авторских прав ведется постоянно, но только сейчас предпринимаются попытки использования средств автоматизации для установления фактов таких нарушений.

1.2 Анализ существующих средств установления заимствований в документах

В качестве одного из средств поиска заимствований можно рассматривать Интернет-сервис AntiPlagiat.ru [14-16], который предлагает набор услуг, реализующих технологию проверки текстовых документов на наличие заимствований. Проверка документа выполняется путем его загрузки на сервер системы, сопоставления текста этого документа с базой данных системы и определения степени уникальности текста. При этом система выдаёт все ссылки на источники, из которых были заимствованы тексты. База данных системы «Антиплагиат» включает как открытые источники сети Интернет, так и закрытые, например полнотекстовую базу Электронной библиотеки диссертаций Российской государственной библиотеки (ЭБД РГБ).

Анализируя эту систему и ряд подобных систем [13-16] нельзя не отметить присущие им органические недостатки. Так, в частности, такие системы в значительной степени ориентированы на установление фактов прямого заимствования. При этом, системой могут выявляться заимствования, в которых была произведена незначительная замена отдельных слов на их синонимы, а также были произведены различные преднамеренные форматные искажения. Но более существенные изменения заимствованного текста, заключающиеся в расширенном использовании синонимов слов и словосочетаний, добавлении или удалении слов,

Труды 15-й Всероссийской научной конференции «Электронные библиотеки: перспективные методы и технологии, электронные коллекции» — RCDL-2013, Ярославль, Россия, 14-17 октября 2013 г.

разбиении или объединении предложений, система не устанавливает, и такие тексты часто определяется как оригинальные.

Такая ситуация обусловлена использованием упрощенной модели представления смыслового содержания анализируемых текстов предложения или более крупного фрагмента текста. Этого вполне достаточно для выявления близких по лексическому составу и синтаксической структуре предложений или фрагментов текста.

Между тем, как утверждается в работах [3,4], связный текст это не набор отдельных предложений. Предложения выступают в тексте не изолированно друг от друга, а в тесной смысловой связи. В основе этой связи лежат мыслительные образы тех конкретных или абстрактных объектов (ситуаций, явлений), которые человек имеет в виду, когда он порождает текст. Образы этих объектов имеют определенную структуру. Кроме того, они дополнительно структурируются человеком при их описании на естественном языке. Соответственно этому структурируется и текст. При прочтении текста у читателя, как и у автора текста, возникнет определенный мыслительный образ. Описание этого мыслительный образа выполнено на основе применения различных языковых средств его выражения. Иными словами, одни и те же результаты интеллектуальной деятельности могут быть описаны с использованием разных форм представления смыслового содержания этого текста. При этом лексический состав и количество предложений текста может быть различными. В этом случае для выявления более сложных изменений заимствованного текста необходимо использовать более адекватную модель представления смысловой структуры текстов.

В последнее время в работах отечественных и зарубежных авторов получают широкое распространение семантические методы сравнения текстов. В работе [20] в контексте описания оригинального решения задачи кластеризации рассматривается метод определение пар текстов с максимальной тематической близостью. Данный способ занимает промежуточное положение между синтаксическими и геометрическими методами. Каждый текст представляется набором лексем, которым поставлена в соответствие числовая характеристика их тематической важности (вес) в этом тексте. В дальнейшем определяется не геометрическое расстояние между текстами, а асимметричная близость между i-ым j-ым документами, определяемая как сумма несимметричных сумм весов лексем пересечения: веса сначала суммируются отдельно по i-ому и j-ому документам.

Таким образом, в работе основной акцент делается на правильность определения тематической роли лексемы.

В работе [21] для выявления близких по смыслу документов (дубликатов) используется так называемый глубокий семантически ориентированный подход. В основе данного метода лежит использование семантических сетей, которые получаются при помощи семантико-синтаксического анализатора. При этом учитываются, как лексические, так и семантические отношения в тексте. При использовании этого метода были выявлены сложности при обработке неправильных и омонимичных фраз, а также отрицательных фраз.

Похожий подход используется и в работе [22]. В качестве инструмента для установления семантических отношений авторы используют электронный тезаурус WordNet. Одной из оригинальных идей, изложенных в данной работе, является то, что семантические профили слов выражаются в терминах явных (LSA), неявных (ESA) и характерных (SSA) понятий. Это решение позволяет перейти от разряженного пространства слов к более богатому и понятному пространству понятий. Это позволяет устанавливать отношения смысловой близости понятий. Для определения меры сходства текстов используется стандартный метод косинусов.

1.3 Теоретическое обоснование необходимости создания нового поколения систем автоматического выявления заимствований

В соответствии с современными теоретическими представлениями в языке и речи наиболее информативными и наиболее устойчивыми единицами смысла являются понятия [3-4,7]. С их помощью описывается смысловое содержание текстов, и именно они являются теми базовыми строительными блоками, на основе которых формируются смысловые единицы более высоких уровней, в частности, предложения. При установлении смысловой близости документов нужно сопоставлять, прежде всего, смысловые единицы текста – понятия, выраженные словосочетаниями. При этом, необходимо учитывать такое явление как вариативность форм представления в тексте одного и того же смысла. Именно это явление в вышеупомянутых системах полностью игнорировалось. Поэтому алгоритмы установления смысловой близости документов должны базироваться на современных процедурах семантико-синтаксического и концептуального анализа. Использование таких процедур позволит выявить не только прямое заимствование, но и установить семантическое тождество документов, отличающихся лексическим составом текста и семантико-синтаксической структурой, но имеющих тождественное смысловое содержание[1-6,8].

Такие системы, базирующиеся на предлагаемой модели представления смыслового содержания текстов, можно отнести к системам выявления заимствований следующего поколения. Эти системы смогут выполнять наряду с функцией выявления смыслового заимствования, также функцию автоматической экспертизы документов на их научную новизну.

В процессе разработки таких систем необходимо разработать концептуальную модель предметных областей, выявить их понятийный и терминологический состав, установить систему смысловых связей между наименованиями понятий и разработать комплекс программных средств автоматического анализа смысловой структуры документов.

В наших исследованиях основной упор делается на семантический анализ содержания научно-технических документов, поэтому необходимо еще учитывать то, что такие документы имеют сложную структуру, некоторые элементы которой повторяются в большинстве исследуемых текстов. Эти элементы имеют различную степень значимости с точки зрения установления заимствований, и многие из них необходимо исключить из рассмотрения или придать меньшую степень значимости.

2 Семантические методы выявления заимствований в текстах документов

2.1 Принципы построения процедур автоматического выявления заимствований

Для проведения исследований с целью разработки семантических методов выявления заимствований было создано программное обеспечение, позволяющее сравнивать тексты документов между собой. Это программное обеспечение базировалось на ранее созданных нами процедурах упрощенного семантико-синтаксического и концептуального анализа [17-18], отличающихся более высокой скоростью обработки текстов.

В общем случае процесс выявления заимствований с использованием упрощенных семантических методов делится на несколько этапов:

1. На первом этапе в заранее обусловленной выборке текстов необходимо установить близкие по их смысловому содержанию тексты. Установление смысловой близости документов выполняется путем сравнения их формализованных смысловых описаний (ФСОД).
2. На втором этапе необходимо установить какие именно фрагменты текстов в наибольшей степени совпадают по своему содержанию с фрагментами других текстов. Для этого в анализируемых текстах устанавливаются местоположения всех совпавших элементов ФСОД (слов и словосочетаний) и выбираются те текстовые фрагменты, в которых содержится максимальное число элементов ФСОД. Таких текстовых фрагментов может быть несколько.
3. На третьем этапе необходимо установить совпадения более мелких фрагментов текстов – одного или контактно расположенных предложений. Для этого полученные фрагменты с помощью процедур семантико-синтаксического и концептуального анализа расчленяются на предложения и в них выделяются наименования понятий и устанавливаются связи между ними. Результаты этого анализа можно представить в виде таблицы связей наименований понятий или в виде графа.
4. В случае необходимости выявления преднамеренной замены наименований понятий на их смысловые инварианты, необходимо по словарю синонимичных фразеологических словосочетаний произвести во всех текстовых фрагментах автоматическую замену исходных слов и словосочетаний на их заранее установленные канонические инварианты. Далее анализ выполняется в соответствие с п.3.
5. На четвертом этапе выполняется процесс выявления заимствований, который заключается в сопоставлении полученных представлений смысловой структуры предложений и принятия решения по каждому случаю в зависимости от полученных результатов совпадения этих представлений, как в пределах конкретного предложения, так и в пределах смыслового фрагмента.

2.2 Программное обеспечение системы автоматического выявления заимствований

На основе приведённых выше принципов была создана процедура сравнения документов (см. Рис. 1). Данная процедура позволяет получить следующие характеристики для проверяемых текстов:

1. Процент совпадения первого текста со вторым.
2. Количество предложений в первом и во втором текстах.
3. Количество совпавших предложений в текстах.

Также для каждого совпавшего предложения мы получаем следующие параметры (см. таблицу 1):

1. Номер каждого из сравниваемых предложений в соответствующих текстах.
2. Вектор соответствий слов в предложениях, в котором “1” показывает, что слово входит в состав совпавшего словосочетания, а “0” означает, что слово не входит в состав совпавших словосочетаний, или оно не является значимым.
3. Процент совпавших слов.
4. Текст предложения.

Таблица 1

Параметры сравнения двух предложений

Наименование текста	Номер предложения в тексте	Вектор соответствий в сравниваемых предложениях	Процент совпадения текстовых фрагментов	Текст предложения
0220xxxxx84	505	111111110110101110	77	Изготовленный корпус оклеивается несколькими слоями стеклоткани суммарной толщиной до 5 мм, зачищается, шпаклюется синтетической шпаклёвкой.
0220xxxxx04	725	11111111011010111000000 00	58	Изготовленный корпус оклеивается несколькими слоями стеклоткани суммарной толщиной до 5 мм, зачищается, шпаклюется синтетической шпаклёвкой (рисунки 4.15, 4.16).

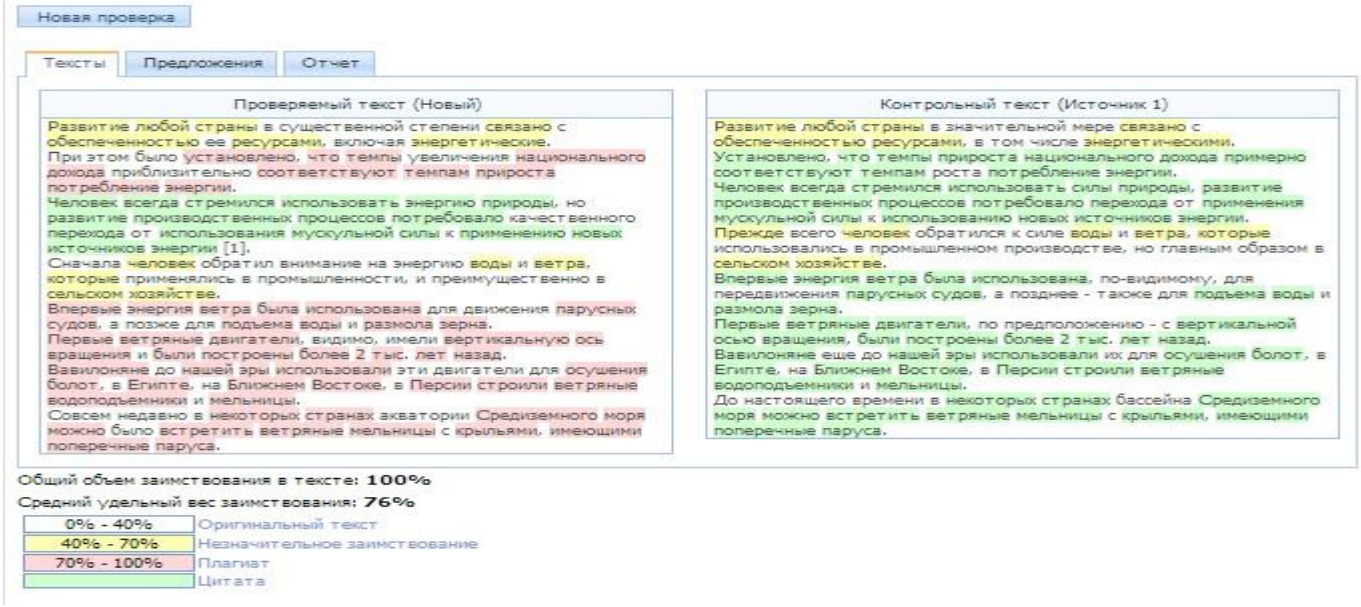


Рис. 1 Пример работы программы

3 Установление заимствований в структурированных научно-технических документах

3.1 Исходные данные

Для проведения исследования была подготовлена подборка отчетов по НИОКР, содержащихся в фонде ЕСУ НИОКР. Данные документы принадлежали к рубрике 55.45 “Судостроение” ГРНТИ за период с 2007 г. по настоящее время, всего 187 документов. Полные тексты найденных документов были обработаны программным обеспечением, описанным выше. Полученные результаты позволили выявить группы документов, в которых есть совпадающие фрагменты текста. Результаты сравнения документов, членов одной из таких групп, представлены в таблице 2. В таблицу вносились документы, в которых было больше 1.5 % совпавших предложений. Более подробные результаты сравнения данной группы документов с документом 0220xxxxx84 представлены в таблице 3. В данном случае брались только предложения, совпавшие на 90 и более процентов.

3.2 Анализ результатов эксперимента

При подробном рассмотрении результатов было установлено, что анализируемые отчеты являются различными этапами одной и той же работы и содержат общие для всей этой работы фрагменты текстов, например в текстах 0220xxxxx84 и 0220xxxxx05 есть фрагмент: «Цель работы в целом: разработка теоретических основ и экспериментальная проверка вопросов проектирования…». Естественно это не является заимствованием. Большинство таких повторов текстовых фрагментов вызвано специфической структурой документа-отчета. Такие случаи необходимо исключать из рассмотрения. В связи с этим возникает необходимость производить анализ структуры документов на основе ранее

Таблица 2

Сравнение группы из четырех документов (результатом является % совпадения)

Наименование документа	0220xxxxx84	0220xxxxx04	0220xxxxx05	0220xxxxx18
0220xxxxx84	-	8.1	4.6	10.5
0220xxxxx04	3.4	-	6.5	1.6
0220xxxxx05	2.6	8.6	-	3.3
0220xxxxx18	8.2	2.9	4.7	-

разработанного словаря структурных элементов документов.

3.3 Членение документа на смысловые фрагменты

Для обеспечения задачи установления смысловой структуры текста необходимо решить задачу разделения текста на его смысловые фрагменты, т.е. текстовые фрагменты, в которых описывается одна и та же ситуация или объект (описание образа объекта). Эта задача очень сложная, поскольку в тексте границы таких фрагментов формально не обозначены. Средствами семантико-синтаксического анализа возможно в тексте выделить только предложения. Но они, как правило, являются только частью смысловых фрагментов. Но если исходить, как было выше сказано, из того, что предложения выступают в тексте не изолированно друг от друга, а в тесной смысловой связи и в основе этой связи лежат мыслительные образы тех конкретных или абстрактных объектов (ситуаций, явлений), которые человек имеет в виду, когда он порождает текст, а также принимать во внимание, что тексты дополнительно структурируются человеком при их описании на естественном языке, то можно предположить, что описания образов этих объектов имеют определенную структуру и в тексте эта структура обозначена в виде абзаца. В тех случаях, когда текст не структурирован, такое членения необходимо выполнить автоматически, исходя из вышеприведенных рассуждений.

Для решения задачи автоматического выявления в тексте смысловых фрагментов, необходимо выполнить его семантико-синтаксический и концептуальный анализ. При этом в тексты будут установлены границы предложений, выявлены наименования понятий и установлены смысловые связи между ними. После построения таблицы связей можно по ней выявить все контактно расположенные предложения, в которых содержатся наименования понятий, связанные с ключевыми наименованиями понятий. Границами таких смысловых блоков будут границы предложений, в которых происходит переход от одного ключевого понятия к другому.

4 Методы установления смыслового тождества текстовых фрагментов

4.1 Модель представления текста

После деления текста на смысловые фрагменты можно приступить к сравнению текстов между собой, для этого необходимо представить их в формализованном виде. Мы решили выбрать модель представления текста, описанную в статьях [17-18], где смысловая структура текста была представлена в виде совокупности нормализованных наименований понятий и связей между ними. Такая смысловая структура текста была названа в этих работах его формализованным смысловым описанием.

В состав формализованного смыслового описания документа включены наименования понятий, сопровождаемые коэффициентом, определяющим степень их смысловой значимости в тексте. Этот весовой коэффициент будет зависеть от их синтаксической роли в предложении и частоты появления в тексте. Подробнее о критериях отнесения слов и словосочетаний к ФСОД изложено в работе [17].

В наших исследованиях мы будем использовать представленное в статьях [17-18] определение формализованного смыслового описания документа (ФСОД), под которым будем называть упорядоченное множество $F = \{Su_i \mid i \in [1, n_F]\}$, где

n_F - количество элементов в формализованном смысловом описании документа;

$Su_i = (Nc_i, w_i, R_i)$ - i-ый элемент ФСОД;

Nc_i — наименование понятия;

w_i - весовой коэффициент, соответствующий наименованию понятия;

R_i - множество связей, относящихся к данному элементу ФСОД.

Таблица 3

Проверка на заимствования документа с именем 0220xxxxx84

Название сравниваемого документа	Всего предложений	Совпало (с вероятностью > 90%)	% совпавших предложений (в тексте)	% совпавших предложений (в фрагментах)
0220xxxxx04	2193	75	8.1	78.6
0220xxxxx05	1650	43	4.6	86.3
0220xxxxx18	1179	97	10.5	88.2

4.2 Отождествление наименований понятий

Для того чтобы решить проблему правильного построения формализованного смыслового описания необходимо решить проблему отождествления наименований понятий. Наименования понятий в текстах могут быть представлены словами и словосочетаниями. При этом описание одинаковых понятий или ситуаций часто может выполняться в терминах различной степени общности и с помощью различных языковых средств. Например, в различных контекстных окружениях наименования понятий могут описываться с использованием явлений словоизменения и словообразования, а также явлений синонимии и гипонимии. Все эти явления существенно затрудняют распознавание и сравнение между собой текстовых форм представления наименований понятий. В связи с этим для отождествления различных форм представления наименований необходимо их приводить к канонической форме. Такое приведение может выполняться на различных уровнях обобщения их смысла: словоизменения, словообразования и синонимии.

Обобщение на уровне словоизменения производится путем последовательного приведения каждого слова к его канонической форме [3-4].

Обобщение на уровне словообразования производится путем выделения у опорных слов их словообразовательных основ и пословной нормализации определяющих их слов. При этом определяющие слова сортируются в их лексикографическом порядке.

Обобщение на уровне синонимии производится по словарю синонимичных слов и фразеологических словосочетаний объемом 450 тыс. словарных статей (Словарь составлен при участии авторов путем обработки двуязычных словарей общим объемом 4,5 млн. словарных статей). В этом словаре представлены смысловые инварианты слов или словосочетаний, один из которых выступает в качестве канонической формы представления смысла наименований понятий словарной статьи. Процедура сведения различных форм представления наименования понятий выполняется путем замены наименования понятия на его каноническую форму представления. Отождествление наименований производится путем сравнение полученных канонических форм двух словосочетаний. В случае успешного сравнения их текстовые формы считаются формами представления одного и того же понятия.

4.3 Процесс установления смыслового тождества текстовых фрагментов

После формирования формализованных смысловых описаний документов, можно перейти к задаче установления смыслового тождества текстовых фрагментов документа. Для этого требуется сопоставить полученные формализованные смысловые описания этих двух текстов. Поскольку фактически мы сопоставляем два графа, нами было решено использовать метод сравнения, который используется для поиска изоморфных пересечений двух графов [19]. Как пишут авторы, этот метод основан на построении графов, по своей структуре сходных с нейронными сетями и названных пирамидами. При данном подходе сначала строится пирамида на основе одного графа, затем на основе второго графа строится пирамида, сходная по структуре с первой пирамидой. Каждой вершине второй пирамиды соответствует подграф второго графа, изоморфный (гомоморфный) подграфу первого графа. Построение пирамид проводится за полиномиальное время, что делает данный метод применимым в данной задаче.

5 Заключение

Описанные в первой части работы методы были реализованы, и полученное программное обеспечение позволило произвести исследования, целью которого являлось выявление общих элементов в научно-технических документах. Результатом данного исследования стало создание алгоритма выявления смысловых фрагментов текста с последующим построением их формализованных смысловых описаний и сравнения методом, представленным в работе [19]. Реализация данного метода будет большим шагом вперед в решении задачи установления заимствований в

структурированных научно-технических документах, ведь до сих пор в данной области семантические методы не получили широкого распространения в связи со сложностью их реализации. В данный момент идет работа по созданию полноценного программного комплекса, реализующего предложенные методы и включающего полный цикл решения задачи нахождения заимствований в текстах документов. Дальнейшим развитием данной системы станет использование данных методов для автоматического проведения экспертизы научно-технических документов на научную новизну, что позволит выявлять заимствования не только текста, но и смысла.

Литература

[1] Кузнецов И.П. Механизмы обработки семантической информации. - М.: Наука, 1978. - 175с

[2] Осипов Г.С. Приобретение знаний интеллектуальными системами: Основы теории и технологии.- М.: Наука. Физматлит, 1997.- 112с

[3] Белоногов Г.Г., Калинин Ю.П., Хорошилов А.А. Компьютерная лингвистика и перспективные информационные технологии. Теория и практика построения систем автоматической обработки текстовой информации — М.: Русский мир, 2004. – 264 с.

[4] Белоногов Г.Г. Теоретические проблемы информатики, Том 2.Семантическиепроблемы информатики. Под общей редакцией К.И. Курбакова. — М.: РЭА им. Г.В. Плеханова. 2008 г. – 342с.

[5] Васильев В.Г., Кривенко М.П. Методы автоматизированной обработки текстов. — М.: ИПИ РАН. 2008г. – 301с.

[6] Крейнес М.Г. Обеспечение активности содержания многоязычия текстовых документов: технология КЛЮЧИ ОТ ТЕКСТА.- Информационное общество. 2000, вып. 2, 241с.

[7] Соссюр Фердинанд де. Курс общей лингвистики. — М.: Прогресс,. 1977. -370с.

[8] Чугреев В.Л. Модель структурного представления текстовой информации и метод ее тематического анализа на основе частотно-контекстной классификации. Диссертация на соискание ученой степени кандидат технических наук. -Санкт-Петербург, 2003. – 185 с.

[9] Зеленков Ю.Г., Сегалович И.В. Сравнительный анализ методов определения нечетких дубликатов для Web-документов // Труды 9 ой Всероссийской научной конференции «Электронные библиотеки: перспективные методы и технологии, электронные коллекции» - RCDL'2007, Переславль, Россия, 2007. – Том 1, С. 166-174.

[10] U. Manber. Finding Similar Files in a Large File System. Winter USENIX Technical Conference, 1994.

[11] A. Broder, S. Glassman, M. Manasse and G. Zweig. Syntactic clustering of the Web. Proc. of the 6th International World Wide Web Conference, April 1997.

[12] S.-T. Park, D. Pennock, C. Lee Giles, R. Krovetz, Analysis of Lexical Signatures for Finding Lost or Related Documents, SIGIR'02, August 11-15, 2002, Tampere, Finland

[13] Р.В. Шарапов, Е.В. Шарапова Система проверки текстов на заимствования из других источников // Труды 13 ой Всероссийской научной конференции «Электронные библиотеки: перспективные методы и технологии, электронные коллекции» - RCDL'2011, Воронеж, Россия, 2011. – Том 1.

[14] 2. Авдеева Н.В., Ботов П.Ю., Букаев А.С., Вислый А.И., Груздев И.А., Житлухин Д.А., Романов М.Ю., Чехович Ю.В. Внедрение системы «Антиплагиат» в Российской государственной библиотеке // Материалы конференции «Интеллектуализация обработки информации» – октябрь, 2010. – С. 499–503.

[15] Никитов, А. В. Плагиат в работах студентов и аспирантов: проблема и методы противодействия / А. В. Никитов, О. А. Орчаков, Ю. В. Чехович // Университет. управление: практика и анализ. - 2012. - № 5. - С. 61 - 69.

[16] Антиплагиат [Электронный ресурс]. — Режим доступа: http://www.antiplagiat.ru

[17] Борзых А.И., Брагина Г.А., Хорошилов А.А. / Методы автоматической кластеризации документов в хранилищах научно-технической информации для решения задачи поиска плагиата в текстах документов // Информатизация и связь, вып.8, 2012

[18] Захаров В. Н., Хорошилов А.А. / Автоматическая оценка подобия тематического содержания текстов на основе сравнения их формализованных смысловых описаний / // Труды XIV-ой Всероссийской научной конференции «Электронные библиотеки: перспективные методы и технологии, электронные коллекции» - RCDL'2012, г. Переславль-Залесский, Россия, 15-18 октября 2012 г.

[19] Агарков А.В. Метод сравнения двух графов за полиномиальное время. // Научно-теоретический журнал "Искусственный интеллект" №.4'2003.

[20] Чанышев О. Г. Метод кластеризации-классификации на основе бинарных классифицирующих таксонов // Труды II Всероссийской конференции «ЗНАНИЯ – ОНТОЛОГИИ – ТЕОРИИ» с международным

участием, г. Новосибирск, 20–22 октября 2009 г.

[21] Hartrumpf, Sven; Tim vor der Brück; and Christian Eichhorn (2010a). Detecting duplicates with shallow and parser-based methods. In Proceedings of the 6th International Conference on Natural Language Processing and Knowledge Engineering (NLPKE), pp. 142-149. Beijing, China

[22] Carmen Banea, Samer Hassan, Michael Mohler, and Rada Mihalcea, UNT: A Supervised Synergistic Approach to Semantic Text Similarity, Proceedings of the Sixth International Workshop on Semantic Evaluation SemEval 2012

Semantic methods for solving a problem of automatic detection of plagiarism in structured scientific and technical documents

Victor N. Zakharov, Alexey A. Khoroshilov

This paper presents the semantic methods for plagiarism detection in structured scientific and technical documents. The existing systems allow to detect plagiarism only when the plagiarism is made by copying text fragments without changes or with minor changes of its structure or lexical composition. The use of semantic methods for text analysis makes it possible to reveal the conceptual structure of the text and recognize more sophisticated cases of intended changes in the plagiarized texts, for example, to determine the cases of substitution of words or word combinations by their semantic invariants, changes in text splitting into sentences, replacement of the text fragments. This paper also grounds the need to detect text fragments, relating to description of structural document elements (for example, standard headings for all documents) and to exclude them from consideration.

Тематическое представление новостного кластера как основа для автоматического аннотирования

© А.А. Алексеев
Московский государственный университет имени М.В. Ломоносова,
г. Москва
a.a.alekseevv@gmail.com

Аннотация

В работе предложен метод извлечения цепочек семантически близких слов и выражений, описывающих различных участников сюжета – тематических узлов. Метод основан на объединении различных факторов схожести, включающих структурную организацию новостных кластеров, анализ контекстов вхождения языковых выражений, а также информацию из предопределенных ресурсов. Контексты слов используются в качестве базиса для извлечения многословных выражений и построения тематических узлов. Оценка предложенного алгоритма произведена в задаче построения обзорных рефератов новостных кластеров.

1 Введение

Современные технологии автоматической обработки новостных потоков основаны на тематической кластеризации новостных сообщений, т. е. выделении совокупностей новостей, посвященных одному и тому же событию – новостных кластеров [17].

Кластер документов должен соответствовать ситуации или совокупности связанных ситуаций (основная тема кластера) [2, 17]. В описываемой ситуации есть набор участников, которые в исходном кластере:

- могут быть выражены не только словами, но и словосочетаниями,
- могут выражаться не одним, а совокупностью различных выражений; так, *акции некоторой компании* могут выражаться в текстах одного новостного кластера как собственно *акции компании, контрольный пакет акций, контрольный пакет, акционер компании, владелец компании, состав владельцев* и др.

Можно предположить, что качественное выделение участников ситуации, включая различные варианты их наименования в различных документах кластера, может помочь лучше определять основную тему новостного кластера, и, значит, повысить качество различных операций с новостными кластерами, например, автоматическое аннотирование, определение новизны информации и др.

В данной работе предлагается модель представления содержания новостного кластера, описывающая основных участников ситуации с учетом вариативности их именования – тематическое представление новостного кластера. Мы рассмотрим методы улучшения качества извлечения основных участников новостного события, что включает нахождение совокупности слов и выражений, с помощью которых тот или иной значимый участник события именовался в документах новостного кластера. Метод основан на совместном использовании совокупности факторов, в том числе разного рода контекстов употребления слов в документах кластера, информации из предопределенных источников (тезаурус русского языка), а также особенностях построения текстов на естественном языке.

Статья организована следующим образом: после обзора существующих подходов, приведенного в разделе 2, в разделе 3 обсуждается теоретическая основа предлагаемого алгоритма, в частности, модель связанного текста. Подробное описание предлагаемого алгоритма и интеграция результатов работы алгоритма в методы автоматического аннотирования представлены в разделах 4 и 5 соответственно. Оценка полученных результатов произведена в разделе 6. Все примеры взяты из новостного кластера, посвященного смене руководства алмазодобывающей компании «Алроса», содержащего 12 новостных документов.

2 Обзор существующих методов

Проблема определения вариативности именования в текстах является актуальной для различных задач автоматической обработки естественного языка. С формальной точки зрения данная проблема является задачей группирования набора языковых выражений входной текстовой коллекции на тематические группы, относящиеся к одинаковым сущностям. Существует ряд различных подходов со схожими постановками задач, наиболее близкими из которых являются:

- построение лексических цепочек;
- построение референциальных цепочек;

Труды 15-й Всероссийской научной конференции «Электронные библиотеки: перспективные методы и технологии, электронные коллекции» — RCDL-2013, Ярославль, Россия, 14-17 октября 2013 г.

- вероятностные тематические модели (в частности, LDA).

Лексическая цепочка представляет собой последовательность семантически связанных слов (повторы, синонимы, гипонимы, гиперонимы и др.) и является известным подходом к моделированию связности текста на естественном языке [10, 14, 21]. Алгоритмы построения лексических цепочек основаны на использовании информации о связях между словами и выражениями, описанных в некотором заранее определенном ресурсе, например, тезаурусе английского языка WordNet и тезаурусе русского языка РуТез.

С целью выделения наиболее значимых для содержания текста лексических цепочек рассматриваются различные параметры лексических цепочек, такие, как частотность ее элементов, текстовое покрытие и другие. В лексических цепочках выделяются наиболее частотные элементы цепочки в качестве наиболее важных тематических элементов текста.

Использование лексических цепочек является из существующих подходов идейно наиболее близким к предлагаемому в данной статье. Основные отличия предлагаемого подхода заключаются в расширении рассмотрения с одного документа на коллекцию документов, а также в использовании совокупности различных факторов для группирования семантически близких слов и выражений (а не только информации, описанной в предопределенном ресурсе).

Частично проблему разного именования именованных сущностей снимают посредством ***установления кореференции имен*** (***референциальных цепочек***), прежде всего, для людей и организаций (*Президент Российской Федерации Дмитрий Медведев, Президент Медведев, Дмитрий Медведев*) [19].

На конференциях TDT и ACE рассматривалась задача по извлечению и прослеживанию упоминаний сущностей по цепочкам кореференции (Entity Detection and Tracking) [6]. Специфика данной задачи заключается в обнаружении ограниченного набора типов сущностей, существующих в реальном мире.

Вероятностные тематические модели, такие, как Latent Dirichlet Allocation [2, 3, 7], основаны на предположении, что документы на естественном языке являются комбинацией различных тем (топиков), в то время как каждая тема (топик) является вероятностным распределением над словами. В подобных моделях обычно рассматриваются два вероятностных распределения:

- Темы (Топики) vs Документы (распределение тем (топиков) по документам коллекции);
- Слова vs Темы (Топики) (распределение слов по темам (топикам)).

Восстановление информации об исходном распределении тем (топиков) основано на итеративном применении статистических методов (например, Семплинга Гиббса [7]), использующих информацию о совместном появлении слов в документах исследуемой коллекции. При этом подобный статистический вывод обычно не учитывает информацию о существующих лексических отношениях между словами и внутреннем устройстве текстов на естественном языке. Результаты работы алгоритмов, основанных на подобных моделях, имеют вероятностный результат и трудно интерпретируемы.

3 Тематическое представление

Как известно, текст обладает такими свойствами, как глобальная и локальная связности. Глобальная связность текста проявляется в том, что его содержание может быть представлено в виде иерархической структуры пропозиций [5]. Самая верхняя пропозиция представляет собой основную тему документа, а пропозиции нижних уровней представляют собой локальные или побочные темы документа.

Локальная связность, т. е. связность между соседними предложениями текста, часто осуществляется такими средствами, как анафорические отсылки, например, с помощью местоимений, или посредством повторения одних и тех же или близких по смыслу слов (лексическая связность).

Пропозиция основной темы документа, т. е. взаимоотношения участников основной темы, должна находить свое отражение в конкретных предложениях текста, которые должны раскрывать и уточнять взаимоотношения между тематическими элементами. Если текст посвящен обсуждению взаимоотношений между тематическими элементами C_1, , C_n, то в предложениях текста должны обсуждаться детали этих отношений. Это проявляется в том, что сами тематические элементы C_1, , C_n или их лексические представители должны встречаться как разные актанты одних и тех же предикатов в конкретных предложениях текста.

Исходя из данных идей, для выявления участников ситуации, описываемой в исходном новостном кластере, мы сделали ряд следующих предположений:

1) взаимодействие участников описывается в предложениях текста, поэтому чем чаще слова (или выражения) встречаются в одних и тех же предложениях текста, тем больше вероятность того, что эти слова (или выражения) относятся к разным участникам ситуации;

2) каждому участнику в тексте соответствует группа слов и выражений; предполагается, что в тексте имеются наиболее частотное (главное название участника) и разные варианты, поэтому группа слов и выражений, относящихся к одному участнику, строится в форме узла, т. е. главное выражение и относящиеся к нему выражения – ***тематический узел***;

3) тематическое представление в предлагаемом подходе представляет собой совокупность выявленных тематических узлов и отношений между ними [9, 14].

Данные предположения основаны на внутреннем устройстве и тематической структуре текстов на естественном языке [5, 10]. Более подробная ин-

формация о сделанных предположениях, а также описание проведенных экспериментов по проверке сделанных гипотез, представлены в работе [1]. Новостной кластер не является связным текстом, но посвящен одной ситуации (или совокупности связанных ситуаций) и содержит большое количество документов, что влечет за собой усиление всех статистических особенностей.

4 Алгоритм построения тематического представления

4.1 Контексты употребления слов

Важным фактором для построения тематического представления являются контексты, в которых употребляются слова и выражения. Для получения контекстов слов предложения разбиваются на фрагменты между знаками препинания. Выделяются следующие типы контекстов в рамках таких фрагментов:

- соседнее прилагательное или существительное вправо или влево от исходного слова (Near);
- во фрагментах, в которых есть глаголы, фиксируются прилагательные и существительные, между которыми и исходным словом встречается глагол (AcrossVerb);
- прилагательные и существительные, встречающиеся во фрагментах предложений с данным словом, не разделенные глаголом и не являющиеся соседними к исходному слову (NotN).

Кроме того, для всех прилагательных и существительных запоминаются слова, встречающиеся в соседних предложениях (NS). Предложения для вычисления этого показателя берутся не полностью, учитываются фрагменты предложений с начала и до фрагмента, содержащего глагол (включительно), что позволяет извлекать из соседних предложений наиболее значимые слова.

4.2 Сборка многословных выражений

Важной основой извлечения многословного выражения из текста документа является частотность его встречаемости в тексте. Однако кластер представляет собой структуру, в которой многие цепочки слов повторяются многократно. Поэтому основным критерием для выделения многословных выражений является значительное превышение встречаемости слов непосредственно рядом друг с другом по сравнению с раздельной встречаемостью во фрагментах предложений [18]:

$$Near > 2 * (AcrossVerb + NotN). \quad (1)$$

Кроме того, используются ограничения по частотности встречаемости слов рядом друг с другом.

Просмотр подходящих пар слов (выражений) для склейки производится в порядке снижения коэффициента Near / (AcrossVerb + NotN). При нахождении подходящей пары слов они склеиваются в единый объект, и все контекстные отношения пересчитываются. Процедура просмотра начинается заново и повторяется до тех пор, пока произведена хотя бы одна склейка.

В результате данной процедуры собираются такие выражения, как *президент компании, международные экономические отношения, председатель совета директоров, контрольный пакет акций* и т. д.

4.3 Характеристики для определения семантических связей

Для определения семантически связанных выражений и последующего построения тематических узлов используется набор из шести основных характеристик схожести. Некоторые из данных характеристик являются контекстно-зависимыми и вычисляются непосредственно на основании рассматриваемого новостного кластера, в то время как другие определяются на основании формальной схожести выражений и информации из заранее определенных ресурсов. Каждая характеристика добавляет некоторый балл в общий вес схожести пары выражений, независимо от других характеристик схожести. В следующей секции будет дано подробное описание алгоритма расчета весов схожести пар выражений.

Контекстно-зависимые характеристики:

Количество вхождений в соседние предложения (**N**eighboring **S**entence **F**eature, **NSF**). Данная характеристика основана на гипотезе глобальной связности текстов на естественном языке [5] и её следствии о том, что элементы одного тематического узла чаще появляются в соседних предложениях исходных документов, чем в одних и тех же предложениях.

Характеристика NSF вычисляется на основе контекстных параметров AcrossVerb, Near, NotNear и NS и распределения их средних значений внутри исходного новостного кластера. Характеристика NSF дает численную оценку соотношения количества вхождений в соседние предложения (характеристика NS) по отношению к количеству вхождений в одни и те же предложения исходного корпуса (характеристики AcrossVerb, Near и NotNear) и основана на следующем соотношении:

$$C = NS - 2 * (\mathrm{AcrossVerb} + Near + NotNear). \quad (2)$$

Общая формула вклада характеристики NSF в вес схожести пары выражений имеют следующую форму:

$$NSF = Min\left[0.5, \frac{C}{Avg(C)}\right], \quad (3)$$

где AVG(C) является средним значением C среди всех положительных значений в рамках всего кластера.

NSF также является управляющей характеристикой. Это означает, что два выражения не могут быть включены в один и тот же тематический узел, если значение характеристики NSF имеет отрицательное значение. Подобная пара с отрицательным значением NSF не имеет общего веса и не рассматривается алгоритмом построения тематических узлов. Стоит

отметить, что подобная характеристика не использовалась раньше для подобных задач, таких, как определение вариантов именования основных участников ситуации, построение рядов квазисинонимов, а также лексических цепочек.

Строгие контексты (**S**trict **C**ontext, **SC**). Данная характеристика основана на сравнении строгих контекстов употреблений слов – текстовых шаблонов. В качестве шаблонов рассматриваются 4-граммы: по два слова справа и слева от рассматриваемого выражения. Чем больше одинаковых шаблонов разделяет пара-кандидат, тем больше схожесть по данной характеристике. Контексты с недостающей информацией (или неполные 4-граммы контекстов, в начале и конце предложений) получают меньший вес, чем целые шаблоны.

Вес шаблона строгого контекста рассчитывается следующим образом: каждое слово n-граммы шаблона контекста имеет вес, равный 0.25. Например, n-грамма (*, *, *состоит*, *из*) будет иметь вес 0.5, а n-грамма (*новостной*, *кластер*, *состоит*, *из*) будет иметь вес, равный 1.0, что является максимальным весом полного шаблона n-граммы.

Значение характеристики SC имеет вещественное значение, принадлежащее отрезку [0,1]. Вес характеристики вычисляется относительно веса пары с максимальным значением разделяемых строгих контекстов, пропорционально весу разделяемых строгих контекстов для текущей пары.

Схожесть контекстов употребления по внутренним характеристикам предложения (**S**calar **P**roduct **S**imilarity, **SPS**). Каждый из контекстных параметров, описанных в разделе 4.1, представляет собой вектор частот, сопоставленных с каждым словом или выражением. Размерности данного вектора отражают частоту совместной встречаемости рассматриваемого слова или выражения со всеми остальными словами и выражениями, упомянутыми в новостном кластере. После построения данных контекстных векторов они могут быть сопоставлены классическими метриками схожести, например, такой, как косинусная мера угла между векторами. Характеристика SPS может быть рассмотрена как более сглаженная и гибкая характеристика по отношению к характеристике SC, так как обе данных характеристики основаны на контекстах употребления слов и выражений.

Значение характеристики SPS имеет вещественное значение, лежащее в пределах от 0 до 0.5 (половинный вес характеристики), и вычисляется как косинусная мера схожести по всем контекстным характеристикам (AcrossVerb, Near и NotNear), ограниченная сверху значением 0.5.

Контекстно-независимые характеристики:

Формальное сходство (**B**eginning **S**imilarity, **BS**). Рассмотрение формального сходства выражений является естественным путем обнаружения семантически связанных объектов. На текущий момент используется простая метрика схожести – одинаковые начала слов. Данная характеристика позволяет находить сходство между такими выражениями, как *Руководитель – Руководство, Президент России – Российский президент* и т. д.

Общий вес характеристики BS имеет вещественное значение из отрезка [0,1] и вычисляется по следующей формуле (в случае, если есть слова с одинаковыми началами, иначе вес равен нулю):

$$BS = 1.0 - 0.1 * N_{diff},$$

где N_{diff} – число слов с различными началами.

Информация о схожести, описанная во внешнем ресурсе – тезаурусе РуТез (**T**hesaurus **S**imilarity, **TS**). На текущий момент существует большое количество разнообразных предопределенных ресурсов, которые содержат в себе дополнительную информацию о связях слов и выражений. Данная информация может быть использована для построения тематических узлов и сделать данное построение более стабильным и качественным. Более того, известно, что некоторые типы отношений между словами и выражениями широко используются для обеспечения связности реальных текстов (например, такие отношения, как синонимия). Вычисление характеристики TS основано на использовании информации из тезауруса русского языка РуТез [13]. При этом в рассмотрение попадали как непосредственные связи объектов, так и «длинные» связи по транзитивным типам отношений. Рассматриваются следующие типы связей: синонимия, часть – целое, род – вид.

Значение характеристики TS имеет вещественное значение от 0 до 1 и вычисляется обратно-пропорционально расстоянию между объектами в тезаурусе:

$$TS = 1.0 - 0.2 * N_{rel},$$

где N_{rel} – длина пути по отношениям тезауруса (количество связей).

Наличие одинаковых языковых выражений (**E**mbedded **O**bjects **S**imilarity, **EOS**). При анализе схожести комплексных тематических узлов, включающих несколько языковых выражений, важным фактором схожести является наличие общих языковых выражений у двух различных узлов. Данный фактор особенно важен на поздних итерациях работы алгоритма, когда имеется значительное количество сформированных тематических узлов и остальные характеристики схожести уже проработаны. Значение данной характеристики является булевым и может добавлять 1 балл в общий вес схожести пары в случае наличия одинаковых языковых выражений.

Общий вес схожести пары рассматриваемых объектов вычисляется как сумма весов по отдельным характеристикам схожести, описанным выше. Таким образом, каждая пара получает вес, лежащий в пределах от 0 (отсутствие схожести) до 5 (максимальная схожесть), получаемый на основе шести характеристик (3 контекстно-зависимых и 3 контекстно-независимых), лежащих в пределах от 0 до 1 (SC, BS, TS, EOS) и от 0 до 0.5 (NSF, SPS). Пример ранжирования пар в соответствии с описанным алгоритмом приведен в Таблица 1 (топ-5 пар по общему весу на первой итерации работы алгоритма, характеристика EOS равна нулю для всех пар на первой итерации).

Features / Pairs	Context-independent		Context-dependent			SCORE
	BS	TS	NSF	SC	SPS	
Президент России – Президент РФ	0.90	1.00	0.00	0.50	0.34	**2.74**
Инвестгруппа– Инвестиционная группа	0.90	1.00	0.20	0.00	0.32	**2.42**
ГМК Норильский никель – Норильский никель	1.00	1.00	0.20	0.00	0.11	**2.31**
Российская Федерация – Россия	0.90	1.00	0.00	0.00	0.25	**2.15**
Отставка – Отставка с должности	0.90	1.00	0.20	0.00	0.00	**2.10**

Таблица 1: Пример ранжирования пар-кандидатов

4.4 Алгоритм построение тематического представления на основе совокупности факторов

Алгоритм построения тематического представления конструирует тематические узлы из пар выражений в порядке убывания их схожести. Предлагаемая структура тематического узла обладает следующими свойствами:

- текстовое выражение может принадлежать к одному или двум тематическим узлам; разрешение множественной принадлежности обеспечивает возможность представления различных аспектов исходного текстового выражения, а также его лексической многозначности;
- каждый тематический узел имеет главный элемент – центр тематического узла, который может принадлежать только к одному тематическому узлу; центр тематического узла является наиболее частотным элементом среди всех элементов тематического узла.

Построение тематического представления состоит из следующих шагов:

- рассматривается пара текстовых выражений с наибольшим весом схожести среди всех пар-кандидатов;
- более частотный элемент пары поглощает менее частотный элемент вместе со всеми его текстовыми вхождениями и контекстными характеристиками и становится представителем данной пары текстовых выражений – центром нового тематического узла;
- менее частотный элемент рассматриваемой пары может в дальнейшем аналогичным образом присоединиться к другому тематическому узлу;
- объединение тематических узлов, состоящих из нескольких элементов, происходит аналогично объединению одиночных текстовых выражений; центр более частотного тематического узла становится центром нового, объединенного тематического узла.

В целом каждая итерация алгоритма состоит из трех основных шагов:

1. ранжирование пар-кандидатов;
2. выбор пары для объединения (наибольший вес + удовлетворение ограничений);
3. процедура объединения.

Итеративный процесс продолжается до тех пор, пока есть пары-кандидаты для объединения с весом схожести выше заданного порога. Например, тематический узел с центральным элементом *Пост* проходит следующие этапы в процессе построения (показаны пары с максимальным весом схожести на разных итерациях; более частотный элемент пары является первым элементом):

Итерация 7: (***Отставка)*** ← ***(Отставка с должности)***

Итерация 33: (***Отставка,*** *Отставка с должности)* ← ***(Уход в отставку)***

Итерация 44: (***Отставка,*** *Отставка с должности, Уход в отставку)* ← ***(Отставка президента)***

Итерация 61: (***Уход с поста)*** ← ***(Уход в отставку)***

Итерация 62: (***Отставка,*** *Отставка с должности, Уход в отставку, Отставка президента)* ← ***(Уход с поста,*** *Уход в отставку)*

Итерация 102: (***Отставка,*** *Отставка с должности, Уход в отставку, Отставка президента, Уход с поста)* ← ***(Пост)***

Итерация 103: (***Пост,*** *Отставка, Отставка с должности, Уход в отставку, Отставка президента, Уход с поста)* ← ***(Должность)***

Итерация 104: (***Пост,*** *Отставка, Отставка с должности, Уход в отставку, Отставка президента, Уход с поста, Должность)* ← ***(Уход)***

Следующие тематические узлы были получены в результате работы описанного алгоритма для кластера примера. Представлены 5 наиболее частотных тематических узлов в порядке убывания частоты. Данные узлы не подвергались какой-либо постобработке, центры тематических узлов выделены жирным шрифтом:

Пост: *уход с поста; должность; уход; отставка; отставка с должности; уход в отставку; отставка президента*

Алроса: *президент Алроса; АК Алроса*

Компания: *акция компании; владелец компании; объединение компаний; акция; акционер компании; владелец; пакет акций; состав владельцев; контрольный пакет акций; контрольный пакет; владение*

Ничипорук*: Александр Ничипорук*

Якутия*: президент Якутии; якутский; якутский президент*

5 Порождение аннотаций на основе тематического представления

Тематическое представление содержит в себе дополнительную информацию о внутреннем устройстве исходного новостного кластера, которая может быть использована для улучшения автоматических операций над текстовыми данными. Одной из таких задач является задача автоматического аннотирования, т. е. подготовки краткого изложения содержания исходного документа(ов). Решение данной задачи в значительной степени связано с наличием информации о различном именовании одних и тех же участников ситуации, описанной во входных документах, так как практически невозможно построить полную и не избыточную аннотацию без учета вариативности упоминаний наиболее значимых сущностей. В данном разделе описаны как существующие известные методы аннотирования (MMR, SumBasic), так и новые подходы, основанные на тематическом представлении. Все представленные подходы используются для оценки качества построенного тематического представления путем его интеграции в исходную структуру данных алгоритмов аннотирования.

5.1 Метод Maximal Marginal Relevance (MMR)

Метод Maximal Marginal Relevance для задачи многодокументного аннотирования является классическим не порождающим (выбирающим для аннотации целые предложения из исходных документов) методом аннотирования, который основан на концепции Maximal Marginal Relevance для информационного поиска [4]. В оригинале данный алгоритм является запрос-ориентированным, но существует также вариант и для общего аннотирования, когда в качестве запроса для аннотирования выступает исходных корпус документов.

Критерий MMR заключается в том, что лучшее предложение для аннотации должно быть максимально релевантным исходному набору документов и максимально отличным от всех предложений, уже отобранных в итоговую аннотацию.

Аннотация строится итеративно на основании списка ранжированных предложений. Предложение с максимальным значением MMR выбирается на каждой итерации алгоритма:

$$MMR = \arg\max_{s \in S} \left[\lambda \cdot Sim_1(s, Q) - (1 - \lambda) \cdot \max_{s_j \in E} Sim_2(s, s_j) \right]$$

где S – множество предложений-кандидатов в аннотацию, E – множество отобранных в аннотацию; λ – представляет собой интерполяционный коэффициент между релевантностью отбираемых предложений и их не избыточностью; Sim_1 – метрика схожести между предложением и запросом для аннотирования (например, косинусная мера угла между векторами, широко применяемая в информационном поиске); Sim_2 может быть такой же, как Sim_1, или другой метрикой схожести. В нашей работе в качестве метрик Sim_1 и Sim_2 использовалась косинусная мера угла между векторами.

5.2 Метод SumBasic

SumBasic – алгоритм для общего многодокументного аннотирования [14, 15]. В его основе лежит эмпирическое наблюдение о том, что более частотные слова рассматриваемого кластера документов с большей вероятностью попадают в экспертные аннотации, нежели слова с низкой частотностью.

Алгоритм SumBasic строится на базе частотного распределения слов в исходном документе и состоит из пяти шагов. На первом шаге происходит расчет вероятностей слов исходного кластера $p(w_i)$:

$$p(w_i) = n/N,$$

где n – число появлений слова w_i в исходной коллекции, N – общее число слов в данной коллекции. Каждому предложению S_j на втором шаге назначается вес, равный средней вероятности слов в данном предложении:

$$weight(S_j) = \sum w_i \in S_j \frac{p(w_i)}{|\{w_i \mid w_i \in S_j\}|}.$$

На третьем шаге предложение с наибольшим весом отбирается в итоговую аннотацию. После этого на шаге 4 происходит пересчет вероятностей всех слов, входящих в отобранное предложение, по следующей формуле:

$$p_{new}(w_i) = p_{old}(w_i) * p_{old}(w_i).$$

На пятом шаге проверяется общая длина получившейся аннотации, и если она не превосходит заданного порога, то происходит переход к шагу 2.

5.3 Аннотирование на основе тематического представления РуТез

В работе [20] предложен метод аннотирования на основе тематического представления, построенного на базе тезауруса русского языка РуТез. Одной из ключевых задач данного алгоритма является обеспечение одинаково высокого качества как полноты изложения информации, представленной в автоматической аннотации, так и её связности.

Алгоритм аннотирования на основе тематического представления на базе РуТез является итеративным, на каждой итерации отбирается по одному предложению. Поставленные цели по комбинации полноты и связности конечной аннотации решаются за счет наложения ограничений на этапе отбора предложений, а именно:

- для обеспечения полноты изложения информации предложение-кандидат должно содержать новый (ещё не упомянутый в отобранных предложениях) тематический узел;
- для обеспечения связности предложение-кандидат должно содержать уже упомянутый в отобранных предложениях тематический узел.

Из всех предложений, удовлетворяющих данным условиям, отбирается предложение с наибольшим весом тематических узлов – отражение основной темы исходного новостного кластера.

Алгоритм аннотирования на основе тематического представления на базе РуТез интересен нам в контексте его основы – тематического представления. Оно имеет схожую структуру с тематическим представлением, описанным в данной работе. При этом оно построено с использованием только одной характеристики – информации о наличии связей в тезаурусе РуТез. В данной работе добавлен ряд новых факторов, которые призваны обогатить полученное тематическое представление, а также повысить его качество.

5.4 Собственные методы аннотирования на основе тематического представления

Документы новостного кластера содержат в себе описание некоторого события (ситуации) или ряда связанных событий (ситуаций). Основная цель автоматического аннотирования заключается в наиболее полном отражении значимых фактов, относящихся к данному событию (ситуации) и описанных в исходной коллекции документов. Событие (ситуация), в свою очередь, характеризуется набором её участников. Под «фактом» в данном случае понимаются описание взаимоотношений между некоторыми участниками события (ситуации) или же детализация описания отдельного её участника.

Тематический узел предлагаемого тематического представления является воплощением некоторого участника события (ситуации) и в идеале должен содержать всевозможные варианты именования данного участника в рамках исходного новостного кластера. Таким образом, в основе предлагаемых методов аннотирования лежит учет наиболее значимых взаимоотношений тематических узлов построенного тематического представления – учет взаимоотношений основных участников события (ситуации). Предлагается два итеративных метода аннотирования, отличающихся стратегией учета значимости отношений между тематическими узлами. На каждой итерации в обоих подходах отбирается по одному предложению.

В качестве основы для расчета значимости отношений между тематическими узлами в первом алгоритме (OurSummary_Nodes) выступает значимость самих тематических узлов. Каждый тематический узел имеет вес, равный суммарной частоте его элементов. На каждой итерации в итоговую аннотацию отбирается предложение, содержащее три наиболее значимых и ещё не упомянутых тематических узла (ТУ_NEW):

$$s_i \Rightarrow \max\left(\sum_{TY\_NEW_j \in s_i,\, i=1..3}^{desc\,weight(TY\_NEW_j)} weight(TY\_NEW_j)\right).$$

В рамках второго предлагаемого алгоритма аннотирования (OurSummary_Relations) критерием для отбора предложения выступает наличие наиболее обсуждаемой и ещё не упомянутой пары тематических узлов. Для каждой пары тематических узлов предварительно рассчитывается количество вхождений в одни и те же предложения исходного новостного кластера – «обсуждаемость» пары. На каждой итерации отбирается предложение, содержащее наиболее обсуждаемую и неупомянутую в отобранных предложениях пару, а также обладающее наибольшим общим весом тематических узлов:

$$s_i \Rightarrow \max\left(\sum_{TY\_REL\_NEW_j \in s_i} weight(TY\_REL\_NEW_j)\right).$$

Итеративный процесс отбора предложений для аннотации в обоих алгоритмах продолжается до тех пор, пока не будет превышен заданный порог по количеству слов. Во всех генерируемых аннотациях данный порог равен 100 словам – стандартный размер аннотации для подобных задач на соревнованиях мирового уровня (DUC, TAC).

6 Оценка качества аннотаций

6.1 Методы оценки автоматических аннотаций ROUGE и Пирамид

Оценка качества порождаемых аннотаций является достаточно сложной процедурой. Несомненно, наиболее правдоподобные оценки можно получить при помощи ручной оценки путём привлечения большого количества экспертов. Но данный метод является очень дорогим и трудоёмким. Поэтому используются автоматические методы оценки качества аннотаций ROUGE [12] и формализованный метод Пирамид.

Метод ROUGE основан на автоматическом сравнении порожденной аннотации с эталонными аннотациями, созданными экспертами. Существуют различные модификации алгоритма, связанные с различными способами сравнения: сравнение n-грамм (ROUGE-N); сравнение максимальных общих последовательностей (ROUGE-L и ROUGE-W); сравнение пропусков монограмм и биграмм (ROUGE-S и ROUGE-SU). В статье [12] показано, что все основные ROUGE-метрики являются значимыми, так как в зависимости от специфики конкретной задачи каждая из метрик может иметь наилучшую корреляцию с ручными аннотациями.

В основе метода Пирамид также лежит сравнение автоматических аннотаций с эталонными аннотациями. Но в отличие от метода ROUGE данное сравнение происходит не в автоматическом режиме, а в ручном, на основании формализованного алгоритма сравнения. Эксперты выделяют из эталонных аннотаций все «информационные единицы» (**S**ummary **C**ontent **U**nits, **SCU**) – факты, описанные в аннотации. Каждая информационная единица получает вес пропорционально количеству упоминаний в экспертных аннотациях. Далее полученные информационные единицы вручную ищутся в автоматических аннотациях. Итоговая оценка аннотации равна общему весу упомянутых информационных единиц по отношению к суммарному весу информационных единиц, извлеченных для данного новостного кластера.

В данной работе оценка качества аннотаций построена на оценке методом ROUGE и дополнительной оценке методом Пирамид.

6.2 Автоматические аннотации и их оценка

Для оценки качества автоматических аннотаций были подготовлены 11 новостных кластеров по различным тематикам, собранные на основе пословной модели представления данных [17]. Независимые эксперты-лингвисты подготовили от 2 до 4 ручных аннотаций для каждого из данных кластеров. Все автоматические аннотации прошли единообразную обработку для автоматической оценки программным пакетом ROUGE [12]:

- ограничение аннотаций длиной свыше 100 слов (рассмотрение только первых 100 слов);
- приведение слов к соответствующим леммам;
- исключение стоп-слов и незначащих частей речи;
- транслитерация всех русскоязычных слов (пакет ROUGE работает только с латинскими символами);
- преобразование автоматических аннотаций в необходимый для пакета ROUGE входной формат (см. документацию пакета).

Всего в оценке участвовали 11 различных модификаций алгоритмов:

- классический пословный MMR в модификациях с учетом и без учета IDF (MMR_WithIDF и MMR_WithoutIDF соответственно);
- MMR с добавлением информации из построенного тематического представления (модификации с и без учета IDF, MMR_WithIDF+Groups и MMR+Groups соответственно);
- классический пословный SumBasic;
- SumBasic с добавлением информации из построенного тематического представления (SumBasic+Groups);
- аннотирование на основе тематического представления на базе тезауруса РуТез (ThematicLines);
- собственный алгоритм аннотирования на основе построенного тематического представления, по тематическим узлам (модификации с и без учета IDF, OurSummary_Nodes_WithIDF и OurSummary_Nodes соответственно);
- собственный алгоритм аннотирования на основе построенного тематического представления, по связям тематических узлов (модификации с и без учета IDF, OurSummary_Relations_WithIDF и OurSummary_Relations соответственно)

6.3 Результаты

В результате работы программного пакета ROUGE каждая автоматическая аннотация получает набор результатов по различным метрикам сопоставления автоматических аннотаций с аннотациями, составленными экспертами. По причине значимости различных ROUGE-метрик для задач с различной спецификой (см. раздел 6.1) в качестве основного параметра для сравнения автоматических аннотаций была взята средняя позиция в результатах по всем основным ROUGE-метрикам.

Метод	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-S	ROUGE-SU	Avg
MMR + Groups	**0,62499 (1)**	**0,41633 (1)**	**0,6021 (1)**	**0,35529 (1)**	**0,36649 (1)**	**1,0**
OurSummary_Nodes	0,58652 (2)	0,36154 (3)	0,5645 (2)	0,32113 (2)	0,33203 (2)	2,2
OurSummary_Nodes_WithIDF	0,58497 (3)	0,33918 (5)	0,55745 (3)	0,30124 (3)	0,31283 (3)	3,4
MMR_WithIDF	0,57623 (4)	0,38116 (2)	0,55503 (4)	0,29792 (4)	0,30971 (4)	3,6
MMR_WithoutIDF	0,56784 (5)	0,34595 (4)	0,55124 (5)	0,26092 (6)	0,27349 (6)	5,2
ThematicLines	0,53416 (6)	0,33364 (6)	0,51238 (6)	0,2713 (5)	0,28243 (5)	5,6
OurSummary_Relations	0,53141 (7)	0,2892 (7)	0,50422 (7)	0,25382 (7)	0,26509 (7)	7,0
SumBasic + Groups	0,52255 (8)	0,22881 (10)	0,493 (9)	0,24356 (8)	0,25525 (8)	8,6
SumBasic	0,51847 (9)	0,24735 (9)	0,49786 (8)	0,23064 (9)	0,24257 (9)	8,8
OurSummary_Relations_WithIDF	0,45494 (10)	0,24856 (8)	0,43768 (10)	0,19419 (11)	0,20492 (11)	10,0
MMR_WithIDF + Groups	0,44475 (11)	0,22238 (11)	0,42318 (11)	0,20627 (10)	0,21648 (10)	10,6

Таблица 2: Результаты оценки автоматических аннотаций методом ROUGE

В Таблица 2 приведены итоговые результаты оценки всех исследуемых модификаций алгоритмов по всем основным ROUGE-метрикам, а также агрегирующая оценка, по которой выполнена сортировка.

Для проведения дополнительной оценки качества автоматических аннотаций лучших и наиболее значимых для нас методов (с и без интеграции построенного тематического представления) была проведена альтернативная оценка полученных аннотаций методом Пирамид [8]. Результаты данной оценки представлены в Таблице 3.

Метод	Оценка по Пирамидам
MMR + Groups	**0,645 (1)**
MMR_WithIDF	0,617 (2)
OurSummary_Nodes	0,602 (3)
SumBasic + Groups	0,575 (4)
SumBasic	0,567 (5)

Таблица 3: Результаты оценки методом Пирамид

На основе результатов оценки полученных аннотаций методами ROUGE и Пирамид необходимо отметить, что:

- наилучший результат показал метод аннотирования, основанный на построенном тематическом представлении;
- добавление тематических узлов к обоим базовым методам улучшило результаты исходных методов;
- предлагаемое тематическое представление показало более высокий результат в аннотировании, чем тематическое представление только на основе тезауруса.

7 Заключение

В статье предложен алгоритм выявления семантически связанных слов и выражений, описывающих различных участников ситуации новостного кластера – тематических узлов. Предложенный алгоритм основан на совместном использовании характеристик схожести различной природы. В дополнение к известным контекстным характеристикам схожести, таким, как анализ жестких контекстов (шаблонов) употребления слов и выражений, используется характеристика, основанная на внутреннем устройстве текстов на естественном языке – анализ встречаемости в соседних предложениях кластера по отношению к встречаемости в одних и тех же предложениях. В едином алгоритме объединены характеристики следующих различных типов:

- формальное сходство слов и выражений;
- информация из предопределенных ресурсов (тезаурус русского языка РуТез [13]);
- контекстные характеристики схожести.

Оценка предложенного алгоритма производилась в контексте применения полученного тематического представления к задаче автоматического аннотирования. Полученные результаты подтверждают, что информация, заложенная в построенных тематических узлах, позволяет улучшать качество алгоритмов много документного аннотирования.

Литература

[1] Alekseev A., Loukachevitch N. *Use of Multiple Features for Extracting Topics from News Clusters* // Труды конференции SYRCODIS'2012, 2012. pp. 3-11.

[2] Allan J. *Introduction to Topic Detection and Tracking* // Topic detection and tracking, Kluwer Academic Publishers Norwell, MA, USA, 2002. pp. 1-16.

[3] Blei D., Ng A., Jordan M. *Latent Dirichlet Allocation* // Journal of Machine Learning Research, 3, 2003. pp. 993-1022.

[4] Carbonell J., Goldstein J. *The use of MMR, diversity-based reranking for reordering documents and producing summaries* // Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Melbourne, Australia, 1998. pp. 335-336.

[5] Dijk van T. *Semantic Discourse Analysis* // Handbook of Discourse Analysis / Teun A. van Dijk, (Ed.), vol. 2. London: Academic Press, 1985. pp. 103-136.

[6] Doddington G., Mitchell A., Przybocki M., Ramshaw L., Strassel S., Weishedel R. *The Automatic Content Extraction (ACE): Task, Data, Evaluation* // Proceedings of Fourth International Conference on Language Resources and Evaluation, 2004.

[7] Griffiths T., Steyvers M. *Finding scientific topics* // Proceedings of the National Academy of Sciences of the United States of America, Vol. 101, No. Suppl. 1, 2004. pp. 5228-5235.

[8] Harnly A., Nenkova A., Passonneau R., Ram-bow O. *Automation of summary evaluation by the pyramid method* // Proceedings of the International Conference on Recent Advances in Natural Language Processing, 2005.

[9] Hasan R. *Coherence and Cohesive harmony* // Understanding reading comprehension / J. Flood, editor, Newark, 1984. pp. 181-219.

[10] Hirst G., St-Onge D. *Lexical Chains as representation of context for the detection and correction malapropisms* // WordNet: An electronic lexical database and some of its applications / C. Fellbaum, editor. Cambrige, MA: The MIT Press, 1998.

[11] Li J., Sun L., Kit C., Webster J. *A Query-Focused Multi-Document Summarizer Based on Lexical Chains* // Proceedings of the Document Understanding Conference, 2007.

[12] Lin C.-Y. *ROUGE: a package for automatic evaluation of summaries* // Proceedings of the Workshop on Text Summarization Branches Out (ACL'2004), Barcelona, Spain, 2004. pp. 74-81.

[13] Loukachevitch N., Dobrov B. *Evaluation of Thesaurus on Sociopolitical Life as Information Retrieval Tool* // Proceedings of Third International Conference on Language Resources and Evaluation, Vol.1, 2002. pp. 115-121.

[14] Loukachevitch N. *Multigraph representation for lexical chaining* // Proceedings of SENSE workshop, 2009. pp. 67-76.

[15] Nenkova A., Vanderwende L. *The impact of frequency on summarization* // Microsoft Research Technical Report, MSR-TR-2005-101, 2005.

[16] Vanderwende L., Suzuki H., Brockett C., Nenkova A. *Beyond SumBasic: Task-focused summarization with sentence simplification and lexical expansion* // Information Processing and Management Journal, Volume 43 Issue 6, November, 2007. pp. 1606-1618.

[17] Добров Б.В., Павлов А.М. *Исследование качества базовых методов кластеризации новостного потока в суточном временном окне* // Труды конференции RCDL'2010, 2010.

[18] Добров Б.В., Лукашевич Н.В., Сыромятников С.В. *Формирование базы терминологических словосочетаний по*

текстам предметной области // Труды пятой всероссийской научной конференции «Электронные библиотеки: Перспективные методы и технологии, электронные коллекции», 2003. с. 201-210.

[19] Ермаков А.Е. *Референция обозначений персон и организаций в русскоязычных текстах СМИ: эмпирические закономерности для компьютерного анализа* // Компьютерная лингвистика и интеллектуальные технологии: труды Международной конференции Диалог'2005, 2005.

[20] Лукашевич Н.В., Добров Б.В. *Автоматическое аннотирование новостного кластера на основе тематического представления* // Компьютерная лингвистика и интеллектуальные технологии: труды Международной конференции Диалог'2009, Вып. 8 (15), 2009. с. 299-305.

[21] Лукашевич Н.В., Добров Б.В. *Исследование тематической структуры текста на основе большого лингвистического ресурса* // Компьютерная лингвистика и интеллектуальные технологии: труды Международной конференции Диалог'2000, 2000. с. 252-258.

Thematic representation of a news cluster as a basis for summarization

Aleksey A. Alekseev

In this paper we consider a method for extraction of various references of a concept or a named entity mentioned in a news cluster. The method is based on the structural organization of news clusters and exploits comparison of various word contexts. The word contexts are used as a basis for multiword expression extraction and main entity detection. At the end of cluster processing we obtain groups of thematically-related elements, in which the main element of a group is determined. Evaluation of the proposed algorithm is performed in news cluster summarization task.

Задачи и методы определения атрибутов пользователей социальных сетей[1]

© Антон Коршунов
Институт системного программирования РАН, Москва
korshunov@ispras.ru

Аннотация

Рост популярности онлайновых сервисов социальных сетей — основных источников персональных данных о пользователях Интернета — открывает беспрецедентные возможности для решения исследовательских и бизнес-задач, а также создания вспомогательных сервисов и приложений для пользователей социальных сетей. Определение неизвестных атрибутов пользователей является одной из фундаментальных проблем анализа социальных данных. В представленной работе рассмотрены методы решения некоторых актуальных задач, связанных с определением скрытых пользовательских атрибутов: поиск сообществ пользователей, определение демографических атрибутов пользователей, а также идентификация пользователей в различных социальных сетях.

1 Введение

Онлайновые социальные сети (Facebook, Twitter, YouTube и другие) к настоящему моменту стали неотъемлемой частью Веба и продолжают набирать популярность [1, 2]. За последнее десятилетие социальные сервисы существенно изменились в плане архитектуры, функционала и пользовательского интерфейса. С одной стороны, это обусловлено стремлением сделать их использование более удобным, а с другой - активной коммерциализацией и необходимостью увеличить время, проводимое пользователями на страницах сервисов.

С точки зрения анализа данных, социальная сеть в её современном понимании представляет собой граф с произвольным числом типов вершин и рёбер, весами и атрибутами, допускающий наличие множественных связей между узлами [3]. Возможность создания текстовых и мультимедийных объектов внутри сети делают её уникальным источником данных о личной жизни и интересах реальных пользователей (переписка, дневники, фотоальбомы, видеозаписи, музыкальные композиции и т.д.). Всё это обуславливает повышенный интерес к сбору и анализу социальных данных со стороны компаний (конкурентное преимущество) и исследовательских институтов (новые задачи и точки приложения известных подходов) [4].

Обработка социальных данных требует также разработки соответствующих алгоритмических и инфраструктурных решений, позволяющих учитывать их размерность. К примеру, граф социальной сети Facebook на сегодняшний день содержит более 1 миллиарда пользовательских аккаунтов и более 100 миллиардов связей между ними. Каждый день пользователи добавляют более 200 миллионов фотографий и оставляют более 2 миллиардов комментариев к различным объектам сети. На сегодняшний день большинство существующих алгоритмов, позволяющих эффективно решать актуальные задачи, не способны обрабатывать данные подобной размерности за приемлемое время. В связи с этим, возникает потребность в новых решениях, позволяющих осуществлять распределённую обработку и хранение данных без существенной потери качества результатов.

Помимо большого объёма данных и высокой динамичности социальной сети, нужно принимать во внимание такие факторы, как нестабильность качества пользовательского контента (спам и ложные аккаунты), проблемы с обеспечением приватности личных данных пользователей при хранении и обработке, а также частые обновления пользовательской модели и функционала. В дополнение к перечисленным проблемам, это требует постоянного совершенствования алгоритмов решения различных аналитических и бизнес-задач.

Одной из фундаментальных задач анализа социальных данных является *определение неиз-*

[1]Работа выполнена при поддержке гранта РФФИ №13-07-12134 офи_м “Исследование и разработка методов распределенной обработки больших баз графовых данных”.

Материалы 15-й всероссийской конференции "Электронные библиотеки: перспективные методы и технологии, электронные коллекции— RCDL-2013, Ярославль, Россия, 14-17 октября 2013 г.

вестных значений атрибутов пользователей. С этой целью специализированные методы анализа сетевых, текстовых и других данных применяются к социальному графу на разных уровнях его организации:

- **на уровне сети** скрытыми атрибутами могут быть группы пользователей (сообщества), в которых состоит пользователь;
- **на уровне пользователя** скрытыми могут быть биографические и демографические атрибуты пользователя, а также его интересы и предпочтения;
- **на межсетевом уровне** скрытым атрибутом пользователя в одной сети может быть идентификатор этого пользователя в другой сети.

В представленной работе рассмотрены некоторые актуальные задачи, связанные с определением атрибутов пользователей, а также разработанные нами методы их решения. Раздел 2 посвящён задаче поиска сообществ пользователей, целью которой является определение набора сообществ, в котором состоит каждый пользователь сети. В разделе 3 рассмотрена задача определения демографических атрибутов пользователей по текстам их сообщений и атрибутам профиля. Раздел 4 содержит описание задачи идентификации пользователей в различных социальных сетях.

2 Поиск сообществ пользователей

Поиск сообществ пользователей является важным инструментом изучения и анализа социальных сетей, позволяющим исследовать мезоскопическую (модульную) организацию сети и использовать полученную информацию для решения различных задач. К примеру, знания о структуре сообществ незаменимы для предсказания связей и атрибутов пользователей, расчёта близости пользователей в социальном графе, оптимизации потоков данных в социальной сети, некоторых аналитических приложений и т.д.

2.1 Задача

С функциональной точки зрения *сообщество* - это группа пользователей, выполняющая общую роль или функцию и обладающая общими свойствами, ценностями и целями. Немаловажным свойством также является тенденция к взаимному влиянию участников сообщества друг на друга.

Поскольку формализация приведённого определения и построение соответствующей модели представляет определённые практические трудности, важно выделить некоторые особенности сообществ пользователей, характерные для социального графа на уровне связей между пользователями.

Благодаря проведённым исследованиям эмпирических данных социальных сетей стало возможным составить следующий список *фундаментальных свойств* сообществ пользователей на уровне связей между пользователями в социальном графе [11, 12, 21]:

- вершины в сообществе более тесно связаны друг с другом, чем с вершинами за пределами сообщества;
- количество рёбер в сообществе растёт суперлинейно в зависимости от его размера;
- сообщества могут пересекаться, т.е. один пользователь может относиться к нескольким сообществам, что хорошо согласуется с тем фактом, что человек одновременно может играть несколько социальных ролей в обществе;
- сообщества имеют иерархическую структуру, что можно объяснить тенденцией человеческого общества к формированию иерархии социальных групп;
- размер сообществ распределён по степенному закону;
- количество сообществ, к которым принадлежит вершина, распределено по степенному закону;
- вершины с небольшой степенью чаще входят в небольшое число сообществ, тогда как вершины с большой степенью входят во множество сообществ.

Использование перечисленных свойств позволяет выполнять поиск сообществ пользователей как множеств вершин социального графа. Результатом работы метода поиска сообществ является покрытие — множество сообществ, в котором каждая вершина принадлежит как минимум одному сообществу.

2.2 Метод

В связи с перечисленными особенностями сообществ пользователей многие алгоритмы определения модульной структуры сетей неспособны корректно идентифицировать сообщества в социальном графе. Потенциально применимые алгоритмы можно разделить на классы, основанные на статистической значимости сообществ, случайных блужданиях, локальной оптимизации подграфов, вероятностных моделях и агентских моделях [5, 6]. Из рассмотренных классов только методы, основанные на агентских моделях, позволяют достичь оптимального сочетания качества

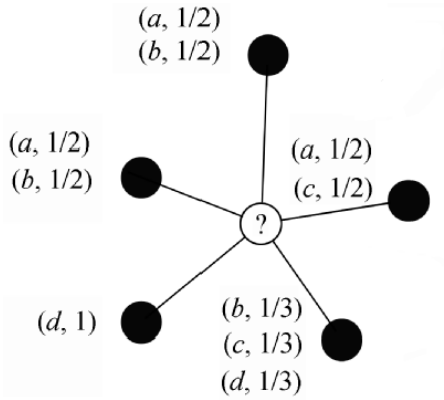


Рис. 1: Алгоритм SLPA: на каждой итерации "говорящие" узлы (окрашены чёрным) посылают "слушающему" узлу (окрашен белым) метки сообществ (a,b,c,d), выбирая их из текущего набора меток для каждого узла. "Слушающий" узел добавляет в свою память самую популярную из полученных меток.

результатов и производительности, необходимого для получения качественных результатов на графах из миллиардов вершин.

Разработанный нами метод представляет собой модификацию алгоритма *SLPA* [10], основанного на агентской модели. Данный алгоритм локально имитирует человеческое общение между парами индивидуумов, а глобально моделирует инфекционный процесс. Основой алгоритма является процесс обмена метками сообществ между вершинами в соответствии с динамическими правилами взаимодействия (рисунок 1):

1. Память каждого узла инициализируется уникальной меткой сообщества;
2. Затем итеративно повторяется последовательность шагов:
   (a) Выбирается "слушающий" узел;
   (b) Каждая из вершин-соседей выбранного узла ("говорящие" узлы) случайным образом выбирает метку с вероятностью, пропорциональной количеству меток данного типа в своей памяти, и посылает выбранную метку "слушающему" узлу;
   (c) "Слушающий" узел выбирает самую популярную из присланных ему меток и добавляет её в свою память.
3. В ходе пост-обработки для каждой вершины выбираются самые популярные метки с помощью заданного порога t;
4. Выбранные метки определяют принадлежность вершин к сообществам.

Вычислительная сложность алгоритма $O(T \cdot |E|)$ для произвольного графа и $O(T \cdot |V|)$ для разреженного социального графа (T - количество итераций).

Кроме того, алгоритм естественным образом формулируется в терминах *Pregel* [10] - вычислительной парадигмы для параллельных вычислений над графовыми данными. Разработанный метод был реализован в рамках *Spark.Bagel* [2] - фреймворка для параллельной обработки данных на кластере из потребительских компьютеров.

Вместе с тем, проведённое нами исследование результирующих покрытий выявило недостаточное качество результатов алгоритма SLPA для случая значительно пересекающихся сообществ. Более детальное исследование получающихся покрытий выявило неспособность алгоритма разделять значительно пересекающиеся сообщества.

Для решения этой проблемы была предложена следующая модификация оригинального алгоритма:

- применить алгоритм поиска *максимальных клик* размером не более 5 вершин к исходному социальному графу;
- всем вершинам, принадлежащим одной клике, назначить одну и ту же метку сообщества;
- вершинам, не принадлежащим найденным кликам, назначить уникальные метки сообществ;
- для каждой вершины найти *локальные сообщества* среди вершин, непосредственно связанных с ней;
- на каждой итерации "говорящий" узел посылает не одну, а несколько случайно выбранных меток каждому из "слушающих" узлов;
- на каждой итерации "слушающий" узел принимает только по одной метке от каждого из своих локальных сообществ (выбирается самая популярная из меток, отправленных вершинами одного сообщества);
- выполнить итерации и пост-обработку по аналогии с оригинальным алгоритмом.

Таким образом, на этапе инициализации намечаются центры будущих сообществ с помощью найденных клик. Затем с помощью модифицированных правил взаимодействия вершин между собой поощряется объединение локальных сообществ в глобальные.

[2] http://spark-project.org/

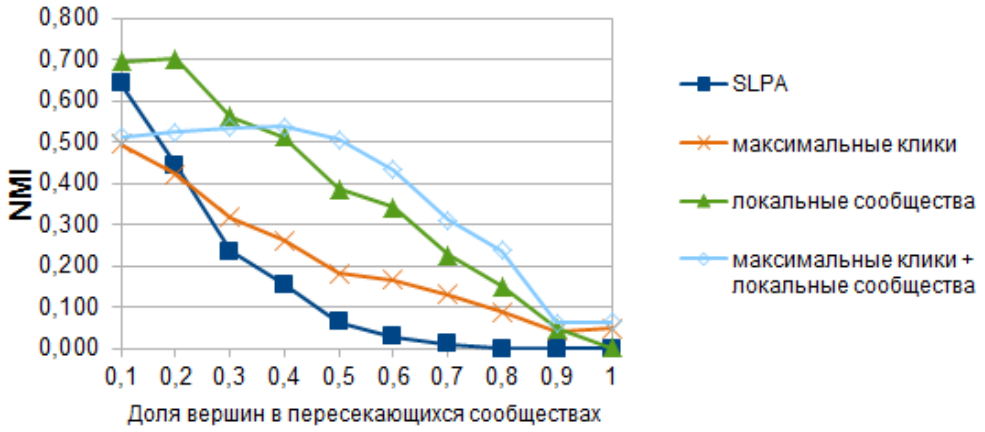


Рис. 2: Результаты оценки качества с помощью неориентированных LFR-графов. Каждый граф состоит из 2000 вершин, часть вершин состоит в пересекающихся сообществах (в данном случае каждая вершина состоит в 6 различных сообществах), остальные вершины состоят в непересекающихся сообществах.

2.3 Результаты

Наиболее распространённым способом оценки качества результатов методов поиска сообществ пользователей является сравнение двух покрытий для некоторого графа: найденного алгоритмом и *референсного*, то есть заранее заданного или известного.

Для оценки качества результатов разработанного метода использовался генератор LFR [11], способный генерировать случайные графы с заданной структурой сообществ.

В качестве количественной меры для сравнения двух покрытий применялась *нормализованная взаимная информация (NMI)* [22], значение которой показывает, в какой степени наличие информации о структуре одного из покрытий уменьшает неопределённость по поводу другого покрытия:

$$N(X:Y) = 1 - \frac{1}{2}[H(X|Y)_{norm} + H(Y|X)_{norm}],$$

где $H(X|Y)_{norm}$ - нормализованная условная энтропия X при условии Y (наоборот для $H(Y|X)_{norm}$), а X и Y - случайные величины, ассоциированные со сравниваемыми покрытиями.

С помощью выбранного метода оценки качества было проведено сравнение разработанного метода поиска сообществ пользователей с современными методами поиска пересекающихся сообществ (алгоритмы SLPA [10], MOSES [20], OSLOM [21] и другие). По результатам тестирования значение NMI для предложенного метода в большинстве случаев превосходит аналогичные показатели выбранных для сравнения методов. В остальных случаях лучшее качество показывают методы, обладающие большей вычислительной сложностью и неприменимые к графам большой размерности (миллиарды вершин).

На рисунке 2 приведено сравнение оригинального алгоритма SLPA с предложенными модификациями. Из графика следует, что лучших результатов позволяет добиться сочетание инициализации кликами с модификацией правил взаимодействия вершин с помощью локальных сообществ. Вместе с тем, использование локальных сообществ без клик позволяет исключить значительный объём вычислений, необходимый для поиска максимальных клик. При этом качество результатов ухудшается незначительно и по-прежнему существенно лучше оригинального алгоритма.

Для оценки производительности разработанного метода было проведено тестирование метода в параллельном режиме с помощью сервиса облачных вычислений *Amazon EC2*. По результатам тестирования метод показал линейную масштабируемость от числа вершин в исходном графе, а также от количества параллельно функционирующих вычислительных элементов.

3 Определение демографических атрибутов пользователей

При заполнении своего профиля в социальной сети пользователи зачастую по ошибке или преднамеренно не заполняют некоторые поля либо дают ложную информацию о фактах своей биографии, интересах и предпочтениях. Кроме того, в контентных сетях (Twitter, YouTube) пользовательский профиль часто ограничен набором базовых атрибутов, недостаточным для решения многих задач, предполагающих персонализацию результатов.

3.1 Задача

В системах интернет-маркетинга и рекомендаций особую важность представляет определение *демографических атрибутов* пользователя для таргетированного продвижения товаров и услуг в группах пользователей с одинаковыми значениями атрибутов. К таким атрибутам относятся пол, возраст, семейное положение, уровень образования, профессия, трудоустроенность, религиозные и политические взгляды, место жительства и т.д. Помимо интернет-сервисов, такие социодемографические характеристики находят применение в различных дисциплинах: социология, психология, криминология, экономика, управление персоналом и др.

Демографические атрибуты можно условно разделить на *категориальные* (пол, национальность, раса, семейное положение, уровень образования, профессия, трудоустроенность, религиозные и политические взгляды) и *численные* (возраст, уровень доходов). Условность разделения связана с тем, что значения численного атрибута можно отобразить в набор категорий и в дальнейшем рассматривать этот атрибут как категориальный. В частности, значения возраста можно разделить на несколько возрастных категорий, что часто применяется на практике.

Важным вопросом в решении задачи определения скрытых демографических атрибутов является выбор признаков. В представленной работе целевым источником данных была выбрана сеть *Twitter* - контентная сеть с преобладанием текстового содержимого (сообщений пользователей). Таким образом, задача состоит в определении скрытых демографических атрибутов пользователей социальной сети по текстам их сообщений. По смыслу задача эквивалентна классической задаче *социолингвистики*: определению характерных особенностей языка представителей различных социальных групп, позволяющих производить частичную идентификацию человека по принадлежности к этим группам. Такая постановка задачи позволяет использовать предложенный метод для обработки данных многих популярных сетей, поскольку текстовые данные являются наиболее распространённым средством коммуникации.

3.2 Метод

Абсолютное большинство современных методов определения демографических атрибутов пользователей основаны на применении методов машинного обучения с учителем с целью классификации пользователей по лингвистическим и другим признакам в предопределённые классы, соответствующие различным значениям изучаемых атрибутов. Сообщения пользователя рассматриваются как набор символьных строк, из которых извлекаются признаки, а для разметки применяются дополнительные источники данных о пользователе, причём в большинстве случаев разметка производится вручную [16–18].

Разработанный нами метод обладает следующими преимуществами:

- автоматическое построение исходного набора данных;
- извлечение большого количества признаков различных типов как из текстов сообщений, так и из полей профиля пользователя, с учётом особенностей микросинтаксиса *Twitter*;
- использование быстрого и эффективного метода отбора информативных признаков;
- расширяемый набор поддерживаемых атрибутов: все поля *Facebook*-профиля, а также любая информация о предпочтениях и интересах пользователя могут быть использованы в качестве атрибутов [3];
- расширяемый набор поддерживаемых языков благодаря использованию автоматической идентификации языка текста сообщений и применению метода построения исходного набора данных, не зависимого от языка.

Метод состоит из следующих этапов:

- построение исходного набора данных;
- предварительная обработка текста;
- построение признакового описания;
- отбор информативных признаков;
- обучение;
- классификация.

Все этапы, за исключением первого, выполняются отдельно для каждого атрибута.

На этапе **построения исходного набора данных** производится сбор данных пользователей из сети *Twitter*. Для каждого пользователя сначала запрашивается только его профиль в сети *Twitter*. При наличии в нём ссылки на профиль того же пользователя в сети *Facebook* (в которой набор пользовательских атрибутов существенно больше, чем в Twitter) запрашиваются и сохраняются все доступные сообщения пользователя из сети *Twitter*. После чего для текущего пользователя запрашивается и сохраняется его профиль в сети *Facebook*, из которого извлекаются указанные пользователем значения его атрибутов.

[3] https://developers.facebook.com/docs/reference/api/user/

Таким образом, элементом набора данных для каждого атрибута и языка является набор символьных строк, полученных из текстов сообщений и профиля одного пользователя в *Twitter*, а также значение атрибута у данного пользователя в *Facebook*.

На этапе **предварительной обработки текста** к текстам полученного на предыдущем этапе набора данных применяется метод определения языковой принадлежности текста (библиотека *language-detection* [4]). После этого данные пользователей распределяются в различные наборы данных в зависимости от языка пользователя.

Предварительно осуществляется фильтрация сообщений, авторство которых не принадлежит пользователю (*ретвиты*). Поскольку цитирование сообщений других пользователей является весьма популярным способом распространения информации в сети *Twitter*, этот шаг предварительной обработки особенно важен для повышения точности метода.

На этапе **построения признакового описания** из сообщений и полей *Twitter*-профиля пользователей извлекаются лингвистические признаки.

Сначала к исходным текстам применяется токенизация. Для элементов специфического синтаксиса сообщений (*хэштегов*, *@-ссылок*), а также слов из полей профиля создаются токены специальных типов, а для обычных слов из сообщений - токены стандартного текстового типа.

Из полученных токенов сообщений и полей профиля строится набор признаков в виде N-грамм размером от 1 до 7 с учётом порядка токенов. Аналогичный набор признаков строится для всех символов в текстах пользователя. Каждый тип признаков представлен двумя подтипами: с учётом и без учёта регистра символов.

Итоговый вектор признаков для пользователя является бинарным, то есть содержит только информацию о наличии или отсутствии признака в его текстовых данных. Количество экземпляров одного признака игнорируется.

На этапе **отбора информативных признаков** применяется метод, основанный на расчёте *условной взаимной информации* [19]. Производится итеративный отбор тех признаков, которые содержат наибольшее количество информации о значении атрибута и при этом существенно отличаются от признаков, выбранных на предыдущих итерациях. Таким образом, каждый признак результирующего набора высоко информативен и слабо зависит от остальных признаков.

На этапе **обучения** производится построение модели классификации с использованием *онлайнового пассивно-агрессивного* алгоритма [13].

На этапе **классификации** в качестве входных данных используются тексты сообщений и поля профиля произвольного пользователя. Выполняется алгоритм классификация для заданного языка и атрибута. Результатом является значение атрибута выбранного пользователя.

3.3 Результаты

Для тестирования использовались наборы данных англоязычных пользователей *Twitter*, размеченные по полу (мужской/женский) и возрасту (моложе 20 лет/от 20 до 40 лет/старше 40 лет). Набор данных, размеченный по полу, включает 3 755 пользователей и 180 240 сообщений. Набор данных, размеченный по возрасту, включает 17 050 пользователей и 818 400 сообщений. Все наборы данных сбалансированы по значениям атрибутов.

Для оценки качества результатов используется точность классификации (*accuracy*). Исходный набор данных разделяется на обучающую и тестовую подвыборки. В качестве входных данных используются тексты пользователей сети *Twitter* из тестовой подвыборки исходного набора данных.

Точность классификации по полу составляет 83,3%. Использование для разметки словарей мужских и женских имён английского языка позволяет увеличить исходный набор данных более чем в 4 раза (70734 пользователя) и повысить точность классификации до 89,2%. Rao et al [16] сообщают о точности 72,33%, Al Zamal et al [18] - о точности 80,2% для идентичной задачи.

Точность классификации по возрасту составляет 71,4%.

4 Идентификация пользователей в различных социальных сетях

Одной из фундаментальных проблем проблем при использовании социальной информации о пользователе является её фрагментированность среди множества различных онлайновых социальных сетей. Каждый год появляется множество как общенаправленных, так и нишевых социальных сервисов, и для активных пользователей Интернет типично иметь несколько профилей в различных социальных сетях. Несмотря на то, что существуют попытки по обеспечению единого способа взаимодействия между различными социальными платформами (например, OpenSocial [5]), они не получили широкого применения, а новые социальные сервисы продолжают появляться. Идентификация пользователя в различных социальных сетях позволяет получить более полную картину о социальном поведении данного пользователя в

[4] https://code.google.com/p/language-detection/

[5] http://opensocial.org/

сети Интернет. Обнаружение аккаунтов, принадлежащих одному человеку, в нескольких социальных сетях, позволяет получить более полный социальный граф, что может быть полезно во многих задачах, таких как информационный поиск, интернет-реклама, рекомендательные системы и т.д.

Поскольку поиск аккаунтов пользователя в различных сетях в общем случае требует наличия актуальных данных обо всех пользователях данных сетей, целесообразно ограничить пространство поиска ближайшими соседями какого-либо пользователя, аккаунты которого в исследуемых сетях известны. Таким образом, задача идентификации пользователей в различных социальных сетях в *локальной перспективе* подразумевает сопоставление аккаунтов пользователей в рамках списков контактов некоторого центрального пользователя в различных социальных сетях. Такая задача часто возникает при работе с контактами пользователей в социальных мета-сервисах, которые, в частности, могут служить для объединения новостных потоков в поддерживаемых социальных сервисах или предоставления единой системы обмена сообщениями. Другая область, в которой возникает подобная задача, это функция автоматического объединения контактов из различных источников (телефонная книга, социальные сети, мессенджеры), распространённая в современных мобильных устройствах.

4.1 Задача

Задача идентификации пользователей заключается в поиске как можно большего числа правильно определенных пар аккаунтов (v, u) таких, что $v \in A, u \in B$, принадлежащих одному и тому же пользователю ($< A, B >$ - социальные графы). Сопоставленный аккаунт для аккаунта $v \in A$ обозначается как $\mathrm{pr}(v) \in B$ и называется *проекцией* аккаунта $v \in A$ в B, а множество всех проекций $\{\mathrm{pr}(v)\}_{v \in A}$ аккаунтов из A в B как $\mathrm{PR}(A)$. Если же для аккаунта $v \in A$ не найдено подходящей проекции, то проекция для v называется *нейтральной* и обозначается как $\mathrm{pr}(v) = \mathbf{N}$. Пример двух таких социальных графов $< A, B >$ и сопоставленных пар аккаунтов изображен на рисунке 3.

Поскольку задача идентификации рассматривается в локальной перспективе, то подразумевается, что графы A и B имеют структуру эго-сетей некоторого центрального пользователя. *Эго-сеть* вершины e представляет из себя граф, состоящий из вершины e, ближайших соседей e, а также данных о связях соседей e между собой и с другими вершинами. Такая формулировка отражает реальные ограничения использования социальных сетей, в которых для предоставлении какой-либо

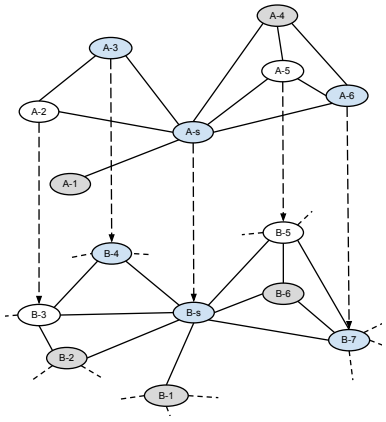


Рис. 3: Результат идентификации пользователей. Пунктирные стрелки обозначают проекции между аккаунтами. Для вершин, закрашенных синим, проекции были известны заранее, проекции для незакрашенных вершин были установлены алгоритмом, для вершин, закрашенных серым, проекции не были найдены

информации о социальных связях пользователя требуется его непосредственное разрешение.

4.2 Метод

Большинство современных методов идентификации пользователей в различных социальных сетях ограничивается лишь анализом атрибутов профилей пользователей, поскольку они зачастую содержат информацию, помогающую идентифицировать пользователя. Общая схема работы таких методов выглядит следующим образом [14]:

1. приведение данных из полей профилей из двух социальных сетей к некоторому общему виду (например, вектору, элементами которого являются атрибуты профилей);

2. попарное применение различных способов сравнения к атрибутам профилей из анализируемых сетей;

3. подсчет результирующего показателя *похожести* между профилями и отсечение всех парных результатов, для которых этот показатель ниже некоторого порогового значения.

После этого все оставшиеся пары считаются сопоставленными между собой и принадлежащими одному пользователю.

Очевидно, что информация, содержащаяся в профилях, достаточно ненадежна, так как данные, указанные пользователем в разных социальных сетях, могут существенно отличаться, быть скрытыми из-за настроек приватности или не поддерживаться в актуальном состоянии.

Одним из способов улучшения результатов описанного подхода является привлечение дополнительных источников данных, в частности информации о социальных связях между пользователями.

Разработанный нами метод [7,8] использует социальные связи обеих рассматриваемых социальных сетей путем сравнения оригинальных списков контактов, естественным образом комбинируя их с информацией атрибутов профилей, благодаря чему лишен многих недостатков существующих методов идентификации пользователей.

Метод основывается на двух основных принципах:

1. задачи выбора проекций для связанных вершин в графе A взаимосвязаны, иначе говоря, выбор проекции для некоторой вершины зависит от значений проекций связанных с ней вершин;

2. если две вершины в графе A связаны, их проекции должны иметь наиболее высокое значение графовой близости.

В качестве функции графовой близости $0 \leq \mathsf{network\text{-}similarity}(\mathrm{pr}(v), \mathrm{pr}(u)) \leq 1$ используется модифицированный коэффициент Дайса:

$$\mathsf{network\text{-}similarity}(v,u) = \frac{2 \cdot w(L_v \cap L_u)}{w(L_v) + w(L_u)}, v, u \in B,$$

где L_v и L_u - множества вершин, связанных с v и u соответственно, а $w(L) = |L|$ - вес этих множеств.

Также предполагается, что один из графов $< A, B >$ является ненаправленным. В дальнейшем без ограничений общности таким графом считается A.

На основе графа A строится модель *условных случайных полей* [15], в которой множество наблюдаемых переменных представлено вершинами графа A: $\mathbf{X} = V(A) = \{\mathbf{x}_v, v \in A\}$, с каждой из которых ассоциирована одна скрытая переменная $\mathbf{Y} = \{\mathbf{y}_v,\ v \in A\}$, определяющая проекцию данной вершины $\mathbf{y}_v = \mathrm{pr}(v) \in B$. Скрытые переменные могут принимать в качестве значения одну из вершин графа B. Связи же наследуются из графа A: $E = E(A)$. Данная модель порождает следующее вероятностное распределение:

$$p(\mathbf{Y}|\mathbf{X}) = \exp(-E(\mathbf{Y}|\mathbf{X})),$$
$$E(\mathbf{Y}|\mathbf{X}) = \sum_{v \in V} \Phi(\mathbf{y}_v|\mathbf{x}_v) + \sum_{(v,u) \in E} \Psi(\mathbf{y}_v, \mathbf{y}_u|\mathbf{x}_v, \mathbf{x}_u),$$

где E - *функционал энергии*, моделируемый функцией *унарной энергии* Φ и функцией *бинарной энергии* Ψ. Обе энергетические функции вещественны и неотрицательны.

Унарная энергия характеризует похожесть вершины в A и его проекции в B на основании полей профилей в этих двух социальных сетях:

$$\Phi(\mathbf{y}_v|\mathbf{x}_v) = \alpha(v) \cdot (1 - \mathsf{profile\text{-}similarity}(v, \mathrm{pr}(v)))$$

Бинарная энергия отвечает за близость между проекциями вершин v и u в графе B:

$$\Psi(\mathbf{y}_v, \mathbf{y}_u|\mathbf{x}_v, \mathbf{x}_u) = 1 - \mathsf{network\text{-}similarity}(\mathrm{pr}(v), \mathrm{pr}(u))$$

Здесь $0 \leq \mathsf{profile\text{-}similarity} \leq 1$, и $\alpha(v) = \log(degree(v)) \geq 0$ - коэффициент баланса между унарной и бинарной энергией.

В качестве функции близости profile-similarity между полями профилей используется вероятность, с которой бинарный классификатор (*C4.5* с *MultiBoosting*) считает их принадлежащими одному пользователю на основании сравнения между строковыми значениями атрибутов профилей. Используемые для сравнения поля профилей *Facebook* и *Twitter* приведены в таблице 1.

Таким образом, для графов $< A, B >$ существует оптимальная конфигурация проекций:

$$\mathbf{Y}^* = \underset{Y}{\mathrm{argmin}} E(\mathbf{Y}|\mathbf{X}),$$

которая минимизирует функционал энергии, максимизируя сумму функций близости и правдоподобие модели.

Разумно допустить, что не для всех вершин необходимо выбирать проекции. В разработанном методы проекция вершины v считаются *заранее известной*, если $\mathsf{profile\text{-}similarity}(v, \mathrm{pr}(v)) \geq \Delta$. Кроме того, проекции некоторых вершин могут быть указаны явно. Использование известных проекций позволяет уменьшить объём вычислений и повысить качество результатов за счёт добавления априорной информации о модели.

Поскольку выбрать разумные фиксированные значения функций близости для *нейтральных* проекций не представляется возможным, для фильтрации неправильно выбранных проекций используется схема обучения бинарного классификатора (*C4.5* с *MultiBoosting*). Используя информацию о контексте каждой вершины в A, классификатор решает, правильно ли для неё выбрана проекция. Для этого используются следующие признаки:

1. $\mathsf{profile\text{-}similarity}(v, \mathrm{pr}(v))$;

2. средняя графовая близость к проекциям смежных вершин;

3. доля заранее известных проекций среди смежных вершин;

4. взаимная согласованность смежных вершин с заранее известными проекциями:

$$\frac{1}{n} \cdot \sum_v \frac{1}{n-1} \sum_{u \neq v} \mathsf{network\text{-}similarity}(\mathrm{pr}(v), \mathrm{pr}(u)|v, u)$$

Таблица 1: Поля профилей Facebook и Twitter, участвующие в сравнении

Поле в Facebook	**Поле в Twitter**
Имя (name)	Имя (name)
	Псевдоним (screen_name)
Веб-сайт (website)	URL

Таблица 2: Результаты экспериментов

метод	полнота	точность	F_1
взвешенная сумма	0.45	0.94	0.61
profile-similarity	0.51	**1.0**	0.69
предложенный метод	**0.80**	**1.0**	**0.89**

4.3 Результаты

Разработанный метод был протестирован на данных из социальных сетей *Facebook* и *Twitter*. 16 *центральных* пользователей, имеющих профиль в обеих сетях, предоставили доступ к своим эго-сетям, а также указали пары аккаунтов, принадлежащих одному и тому же пользователю. Для всех участников эксперимента были загружены профили их друзей (вместе со связями между ними), а также друзей их друзей. В *Twitter* профиль загружался только при наличии между пользователями взаимных связей *следования* для поддержания семантики связей *дружбы*, характерных для *Facebook*. Суммарное число профилей в *Twitter* и *Facebook* 398 и 977, а число связей 108 и 641 соответственно. Общее число сопоставленных пар пользователей - 102.

Для оценки качества результатов используется точность, полнота и F_1-мера. Исходный набор данных разделяется на обучающую и тестовую выборки. Для расчёта показателей качества применяется кросс-валидация с разбиением исходных данных на 3 непересекающихся блока. В качестве входных данных используется пара эго-сетей в *Facebook* и *Twitter* какого-либо центрального пользователя.

Для сравнения были выбраны два базовых алгоритма, основанных на расчёте похожести профилей пользователей. Первый алгоритм использует взвешенную сумму значений функций строковых близостей между полями профилей, коэффициенты для которых подбирались при помощи линейной регрессии из предположения, что между правильно сопоставленными профилями сумма близостей должна быть равна 1. Второй алгоритм использует функцию profile-similarity. Результатом работы базового алгоритма считается *максимальное паросочетание* между графом A и B с некоторым порогом близости профилей, ниже которого проекция не включалась в результаты и считалась нейтральной.

Результаты тестирования приведены в таблице 2.

5 Заключение

В работе были рассмотрены основные особенности социальных сетей как источников данных, а также некоторые задачи и методы анализа разнородных пользовательских данных из социальных сетей, связанные с определением неизвестных значений пользовательских атрибутов.

Одной из доминирующих тенденций развития социальных сетей как социокультурного феномена является более глубокое понимание особенностей социального поведения человека и, как следствие, создание новых средств для самовыражения, а также обмена информацией и опытом. Разумно ожидать дальнейшего расширения пользовательской модели и функционала социальных сетей, что приведёт к появлению новых типов данных в виде объектов и связей социального графа и, как следствие, возможности решать новые задачи, связанные с обработкой персональной информации.

Благодарности

Автор благодарит научного руководителя д.т.н. Кузнецова С.Д., а также коллег из отдела информационных систем ИСП РАН за помощь в разработке и реализации представленных в работе методов: Аванесова В.С., Андрианова И.А., Бартунова С.О., Белобородова И.Б., Бузуна Н.О., Гомзина А.Г., Ипатова С.А., Турдакова Д.Ю., Филоненко И.И.

Список литературы

[1] D. M. Boyd, N.B. Ellison. Social network sites: Definition, history, and scholarship // Journal of Computer-Mediated Communication, 2007, 13(1), article 11

[2] George Pallis, Demetrios Zeinalipour-Yazti, Marios D. Dikaiakos. Online Social Networks: Status and Trends // New Directions in Web Data Management 1, Studies in Computational Intelligence Volume 331, 2011, pp 213-234

[3] Facebook Open Graph. `https://developers.facebook.com/docs/opengraph/`

[4] Social Network Data Analytics. Editors: Charu C. Aggarwal // Springer, 2011

[5] Nazar Buzun, Anton Korshunov. Innovative Methods and Measures in Overlapping Community Detection // Proceedings of the International Workshop on Experimental Economics and Machine Learning (EEML 2012). — Leuven, Belgium, 6 May 2012

[6] Назар Бузун, Антон Коршунов. Выявление пересекающихся сообществ в социальных сетях // Доклады Всероссийской научной конференции «Анализ изображений, сетей и текстов» (АИСТ'2012). — Екатеринбург, 16-18 марта 2012 г.

[7] Sergey Bartunov, Anton Korshunov, Seung-Taek Park, Wonho Ryu, Hyungdong Lee. Joint Link-Attribute User Identity Resolution in Online Social Networks // Proceedings of The Sixth SIGKDD Workshop on Social Network Mining and Analysis (SNA-KDD'12). — Beijing, China, 12 August 2012

[8] Сергей Бартунов, Антон Коршунов. Идентификация пользователей социальных сетей в Интернет на основе социальных связей // Доклады Всероссийской научной конференции «Анализ изображений, сетей и текстов» (АИСТ'2012). — Екатеринбург, 16-18 марта 2012 г.

[9] J. Xie, B. Szymanski. Towards Linear Time Overlapping Community Detection in Social Networks // PAKDD 2012

[10] Grzegorz Malewicz, Matthew Austern, Aart Bik, James Dehnert, Ilan Horn, Naty Leiser, Grzegorz Czajkowski. Pregel: a system for large-scale graph processing // Proceedings of the 2010 ACM SIGMOD International Conference on Management of data

[11] Andrea Lancichinetti, Santo Fortunato. Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities // Physical Review E80, 016118 (2009)

[12] Jaewon Yang, Jure Leskovec. Defining and Evaluating Network Communities based on Ground-truth // Proceedings of 2012 IEEE International Conference on Data Mining (ICDM), 2012

[13] Koby Crammer, Ofer Dekel, Joseph Keshet, Shai Shalev-Shwartz, Yoram Singer. Online Passive-Aggressive Algorithms // JMLR, 7(Mar):551–585, 2006

[14] I. Veldman. Matching profiles from social network sites: similarity calculations with social network support // Master's thesis, University of Twente, 2009

[15] J. Lafferty, A. McCallum, F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data // Proc. 18th International Conf. on Machine Learning, 2001. Morgan Kaufmann. pp. 282–289

[16] Delip Rao, David Yarowsky, Abhishek Shreevats, Manaswi Gupta. Classifying Latent User Attributes in Twitter // Proceedings of the 2nd International Workshop on Search and Mining User-generated Contents, 2010

[17] John D. Burger, John Henderson, George Kim, Guido Zarrella. Discriminating Gender on Twitter // Proceedings of the Conference on Empirical Methods in Natural Language Processing, 2011

[18] Faiyaz Al Zamal, Wendy Liu, Derek Ruths. Homophily and Latent Attribute Inference: Inferring Latent Attributes of Twitter Users from Neighbors // Proceedings of the Sixth International AAAI Conference on Weblogs and Social Media, 2012

[19] François Fleuret. Fast Binary Feature Selection with Conditional Mutual Information // JMLR, 5:1531–1555, 2004

[20] Aaron McDaid, Neil Hurley. Detecting Highly Overlapping Communities with Model-Based Overlapping Seed Expansion // Proceedings of the 2010 International Conference on Advances in Social Networks Analysis and Mining (ASONAM '10)

[21] Andrea Lancichinetti, Filippo Radicchi, José J. Ramasco, Santo Fortunato. Finding statistically significant communities in networks // PLoS ONE 6, e18961 (2011)

[22] Andrea Lancichinetti, Santo Fortunato1, János Kertész. Detecting the overlapping and hierarchical community structure in complex networks // New J. Phys. 11 033015, 2009

Problems and methods for attribute detection of social network users

Anton Korshunov
Institute for system programming of RAS

The increasing popularity of online social network services — the main sources of personal data of Internet users — brings unprecedented opportunities for solving research and business problems, and also for building auxiliary services and applications for social network users. Detection of latent user attributes constitutes a fundamental problem of social data analysis. In the paper, methods for solving several actual problems related to latent user attribute detection are considered: user community detection, detection of demographic user attributes, and user identity resolution in different social networks.

Разработка методов и средств контроля достоверности и актуальности фактографического наполнения информационных систем

© А.С. Серый
Институт Систем Информатики им. А.П. Ершова СО РАН
Новосибирск
Alexey.Seryj@iis.nsk.su

Аннотация

В данном исследовании представлены методы и подходы к автоматизации обработки входящего потока данных в информационной системе, где информация — это множество информационных объектов, соответствующих понятиям и отношениям онтологии предметной области системы. Решаются задачи поиска референциальных связей и идентификация объектов. Кроме того, предлагаются методы получения из трастовых метрик (trust metrics) информационных ресурсов соответствующих метрик для извлекаемых документов и той информации, которая заключена в извлекаемых из документов объектах. Предполагается, что такой подход позволит обеспечить удаление утративших доверие данных, тем самым, снизив долю участия эксперта в процессе проверки информации и уменьшив количество ошибок и противоречий в системе.

1 Введение

В современном мире информационные системы занимают довольно обширную нишу. Глобальная Сеть непрерывно пополняется новой информацией: как текстовой, так и мультимедийной. Пользователю все труднее становится найти то, что было бы для него полезно. Отсюда и появление многочисленных поисковых сервисов, информационных порталов, а также систем, аккумулирующих информацию, относящуюся к некоторой ограниченной области знаний или предметной области.

Результаты исследований двух последних десятилетий привели к активному использованию онтологий в качестве концептуальных схем реляционных баз данных, лежащих в основе информационных систем [2]. В контексте данной работы понятия онтологии и концептуальной схемы используются как равнозначные, онтология предметной области задается в виде базовых понятий, организованных в таксономию, и совокупности связей между ними. Данные, при этом, представляются в виде множества разнотипных информационных объектов — экземпляров понятий и отношений онтологии. В совокупности объекты образуют контент или информационное наполнение системы. Каждый объект определяется понятием или отношением онтологии и, являясь экземпляром класса, имеет заданную им структуру.

Информационная система должна отображать изменения, происходящие в ее предметной области. Очевидно, что накапливаемые в системе факты (свойства или утверждения об объектах) могут оказаться неверными, противоречивыми или некорректными. Поддержание контента в актуальном состоянии повышает эффективность исполнения системой своих функций, позволяет менее расточительно использовать компьютерные ресурсы и снижает вероятность возникновения ошибок.

В данной работе предложены метод автоматической обработки входящего потока информационных объектов и метод оценки достоверности данных в информационной системе.

Входящим потоком данных в нашем случае считается множество информационных объектов, источником которых являются текстовые документы. Основными задачами данного этапа являются поиск референциальных связей объектов и разрешение контекстной омонимии (или идентификация) объектов. Контекстная омонимия зачастую сопровождает процесс автоматической обработки текстов на естественном языке и проявляется в наличии двух и более вариантов отождествления полученных из текста объектов с объектами базы данных информационной системы. Разрешение референции предполагает поиск кореферентных информационных объектов, т.е.

Труды 15-й Всероссийской научной конференции «Электронные библиотеки: перспективные методы и технологии, электронные коллекции» — RCDL-2013, Ярославль, Россия, 14-17 октября 2013 г.

описывающих одну и ту же внеязыковую сущность предметной области, называемую референтом. Разработано множество методов поиска референциальных связей языковых выражений в текстах, но, в силу сложности подобных задач вообще и для русскоязычных текстов в частности, они не всегда решаются целиком. Не охваченные в процессе обработки текста случаи могут послужить причиной появления информационных объектов, собранных на основе кореферентных выражений. Наличие подобных объектов во входящем потоке данных нежелательно, т.к. снижает точность идентификации. Разработанный для решения этой задачи подход позволяет абстрагироваться от технологии обработки текста, лишь налагая на формат объектов некоторые требования, определяемые способом описания онтологии предметной области [3].

Задача поддержания актуальности данных ставится разработчиками информационных систем повсеместно, однако используемые методы могут сильно различаться. Универсальным методом можно назвать периодическую сверку с источником. В большей степени это относится к информационно-справочным системам. Перепись населения страны также можно назвать сверкой с источником и на примере такой переписи можно увидеть, что процедура перепроверки данных может быть весьма продолжительной и дорогостоящей; более того, перепроверка данных возможна не всегда. Предлагаемый в данной работе метод позволяет оценить достоверность данных в информационной системе, спроектированной на основе онтологии, отслеживать его изменения и удалять информацию, которой более нельзя доверять.

2 Поиск кореферентных объектов

Поиск кореферентных объектов рассматривается как подготовительный этап процедуры идентификации и включает в себя установление степени сходства объектов, построение множества гипотетических эквивалентов для каждого объекта и объединение кореферентных объектов. Подход, разработанный для решения этой задачи, опирается на результаты работы группы исследователей из университета Стэнфорда [4] по разрешению референции между языковыми выражениями в текстах на английском языке, а также на исследования компании RCO [5] закономерностей использования референции при построении связных предложений на русском языке. Кроме того, концепты предметной области и их экземпляры, представленные в системе, должны удовлетворять ограничениям, описанным в [6].

Разрешение кореферентности объектов представляет собой итерационный процесс, где одной итерации соответствует единичный проход по множеству входных объектов и проверка каждого из них на наличие эквивалента — ближайшего кореферентного объекта. В случае обнаружения для объекта q эквивалентного объекта q' они объединяются в кластер, который в дальнейшем интерпретируется как единый объект q''. В рамках одной итерации для каждого объекта q выполняются действия, описанные в п. 2.1–2.3.

2.1 Вычисление степени сходства q со всеми объектами из его окрестности

Для сравнения объектов вводится коэффициент сходства $SI(q^1, q^2)$ (similarity index), где q^1 и q^2— сравниваемые объекты.

$$SI(q^1, q^2) = \begin{cases} k \cdot SI_C + (1-k) \cdot SI_L;\ SI_c \neq 0 \\ 0;\ SI_c = 0 \end{cases} \quad (1)$$

$$0 \leq k \leq 1$$

Операция вычисления SI не является коммутативной, поэтому будем говорить, что вычисляется степень сходства объекта q^2 с объектом q^1. Объект q^1 при этом называется эталоном, а q^2 — кандидатом. $SI_C = SI_C(q^1, q^2)$ называется *таксономической близостью* объектов q^1 и q^2 и зависит от взаимного расположения соответствующих им классов онтологии в ее иерархическом древе, $SI_L = SI_L(q^1, q^2)$ характеризует близость наборов свойств: атрибутов и связей. Коэффициент k регулирует уровень влияния онтологических и атрибутивно-реляционных факторов на итоговую величину. Его значение определяется экспериментальным путем и может изменяться в зависимости от задачи. Формулы для вычисления выражения (1) и его подвыражений подробно описаны в [6].

2.2 Построение множества потенциальных эквивалентов объекта q

Множество потенциальных эквивалентов объекта q состоит из всех объектов q', удовлетворяющих условиям (2).

$$Pr(q) = \{q' \in Ctx(q) | SI(q', q) > \alpha > 0\} \quad (2)$$

Здесь $Ctx(q)$— это некоторая окрестность объекта q в списке объектов (изначально объекты упорядочены по встречаемости в источнике). Размер окрестности определяется исходя из правил и экспериментальных наблюдений, в частности описанных в компании RCO [5, 6].

2.3 Выбор эквивалента для q из множества его потенциальных эквивалентов

Эквивалентом объекта q считается ближайший к нему объект q' из множества $Pr(q)$ с максимальным либо близким к максимальному значением $SI(q', q)$. Если таковой отсутствует или не является предшествующим объекту q, то говорим, что q упомянут в тексте впервые. Объект, состоящий в кластере и не имеющий эквивалента, т.е. соответствующий самому первому упоминанию, будем называть его глобальным эквивалентом или G-эквивалентом. В случае невозможности выделить единственный эквивалент, говорим, что объект q не имеет эквивалента.

2.4 Условия остановки и результат

Итерации следует повторять до тех пор, пока существует возможность строить новые кластеры

или пополнять уже существующие. Первая итерация, не принесшая новых данных, считается завершающей.

Интерпретация кластеров как обычных объектов позволяет на каждом шаге процесса в полной мере использовать информацию о референциальных связях объектов, добытую на предыдущих шагах. За счет интеграции внутри кластера информации обо всех содержащихся в нем объектах такой подход повышает эффективность всего процесса в целом.

Отношение кореферентности объектов, обозначим его $\Re$, очевидно, является отношением эквивалентности. Множество объектов Q разбивается, таким образом, на непересекающиеся кластеры, представляющие собой классы эквивалентности по отношению $\Re$, а после объединения кореферентных объектов мощность Q совпадет с мощностью соответствующего фактор-множества $Q/\Re$. Ясно, что $|Q/\Re| \le |Q|$. Снижение количества кореферентных объектов призвано повысить эффективность следующего этапа — идентификации.

3 Идентификация информационных объектов

Идентификация заключается в разрешении контекстной омонимии входных объектов, когда входному объекту по его набору атрибутов можно сопоставить несколько объектов из базы данных.

Предлагаемый подход предполагает наличие «стартового» списка идентифицированных объектов, который, может быть получен с помощью процедуры поиска по точному совпадению минимального набора атрибутов, определяющих объект. Если был найден лишь один объект, то входной объект считается идентифицированным, и дальнейший его анализ уже не требуется. В итоге множество информационных объектов разделяется на множество *A* идентифицированных объектов и множество *B*, куда входят те объекты, которые не удалось идентифицировать «сходу». Каждому объекту $q \in B$ сопоставляется множество Q объектов из базы данных — множество похожих объектов. Строится оно путем сравнения *q* с объектами базы данных по различным подмножествам атрибутов. Алгоритм построения множества похожих объектов представлен в листинге 1. Здесь `Intersect(q, i)` возвращает объекты, совпадающие с *q* не менее чем по *i* атрибутам.

```
algorithm ПОИСК ПОХОЖИХ ОБЪЕКТОВ
var q = {a_k|k = 1,…, n} %объект, включающий n атрибутов
Q результирующее множество объектов
T, i вспомогательные переменные
begin
Q ←∅
i ← 1
while i ≤ n
T ← Intersect(q, i)
if T = ∅ then
return Q
Q ← T
i ←i + 1
end while
return Q
end algorithm
```

Листинг 1. Поиск похожих объектов

Результатом работы алгоритма будет множество Q объектов, совпадающих с заданным объектом *q* по максимальному количеству атрибутов.

Для того чтобы идентифицировать объект, необходимо сузить множество похожих объектов до одного, т.е. снять неопределенность. Алгоритм идентификации описан в листинге 2 (подробнее см. [7]). В описании алгоритма присутствуют следующие вспомогательные функции:

- `Active(q)` возвращает **`true`**, если `q` активен, **`false`** – иначе
- `Activate(q)` присваивает объекту активный статус
- `Deactivate(q)` присваивает объекту неактивный статус
- `Move_Object(q, Q)` переносит объект `q` во м-во `Q`
- `Move_Rel(r, R)` переносит отношение `r` во м-во `R`
- `Filter(Q, R, i)` удаляет из м-ва `Q` объекты, имеющие не более `i` отношений, аналогичных отношениям из `R`.

```
algorithm ИДЕНТИФИКАЦИЯ ОБЪЕКТОВ
var A множество идентифицированных объектов
    B множество неидентифицированных объектов
    F^A, F^B, D^A, D^B, S^b, i вспомогательные переменные
begin
 A ← стартовое м-во идентифицированных объектов
 B ← м-во неидентифицированных объектов
 while B ≠ ∅
  Choose b ∈ B: Active(b) = true
  if ∄b then
   return A
  F^A← связи b с объектами из A
  F^B← связи b с объектами из B
  S^b← ПОИСК_ПОХОЖИХ_ОБЪЕКТОВ(b)
  i ← 1
  while i ≤ |F^I|& S^b ≠ ∅
   Filter(S^b, F^I, i)
   if ∃!q∈S^b then
    Move_Object(b, A) %q – объект БД, эквивалентный b
    ∀d∈B: ∃r(b,d) %r – связь объектов
```

```
      d и b
       D^A← связи d с объектами из A
       D^B← связи d с объектами из B
       ∀r ∈ D^B: r(b, d)
        Move_Rel(r, D^A)
       if Active(d) = false then
        Activate(d)
      ВЫХОД ИЗ ЦИКЛА
     i ← i + 1
   end while
   if b ∉ A then
    Deactivate(b)
  end while
  return A
end algorithm
```

Листинг 2. Идентификация объектов

Фактически, для b∈B мы находим непустое подмножество множества S^b, такое, что его элементы имеют наибольшее число связей, аналогичных связям объекта b. Контекстная омонимия для объекта b снимается, если это подмножество содержит единственный элемент.

4 Достоверность как показатель доверия к информации

Для того чтобы оценить полезность факта для информационной системы, необходимо определить его трастовую метрику или *достоверность*. Фактом, в нашем случае, называется минимальное знание об объекте, другими словами это либо значение атрибута объекта, либо его связь с другим объектом. Достоверность (trustworthiness) определяет, до какой степени может доверять данному факту рядовой пользователь информационной системы. Для оценки используются характеристики источников факта, и учитывается время его существования в информационной системе. Данные характеристики описаны ниже.

Пусть F — некоторый факт, d^i — i-й документ, упоминающий F. Обозначим экспертную оценку документа d^i как $x^i \in [-1;1]$. Экспертная оценка характеризует уровень доверия эксперта к информации из документа d^i на основании знаний об источнике этого документа и, возможно, какой-то дополнительной информации, которой располагает эксперт. Границы интервала, в которых заключено значение экспертной оценки, соответствуют предельным случаям: полному доверию при $x^i = 1$ и, соответственно, полному недоверию при $x^i = -1$. Значение $x^i = 0$ соответствует отсутствию информации об источнике у эксперта. Значения по умолчанию в случае отсутствия экспертной оценки x^i вычисляются по формуле $x^i = \frac{N-1}{N}$, где N— количество различных источников, содержащих документ d^i.

Введем характеристику источника, выражающую вероятность получения из него достоверного знания. Она должна быть непрерывным образом связана с экспертной оценкой. Семейство функций вида **(3)** очевидно обладает требуемыми свойствами.

$$f_{\mathfrak{M}}(x) = \left(\frac{x+1}{2}\right)^{\mathfrak{M}} \mathfrak{M} = 1, 2, \ldots \quad \textbf{(3)}$$

Допустим, что нам известно среднее значение допущенных при извлечении фактов ошибок. Пусть φ — это среднее отношение допущенных ошибок к общему числу извлеченных фактов. В простейшем случае мы считаем φ константной величиной, но в общем виде ничто не мешает обозначить ее $\varphi(t)$ и вычислять как функцию от некоторого аргумента. В дальнейшем для определенности будем считать $\varphi = const$.

С помощью функций $f_{\mathfrak{M}}(x)$ и параметра φ породим семейство вероятностных характеристик документа d^i.

$$\boldsymbol{\delta}^{\boldsymbol{i}}_{\mathfrak{M}} = \varphi\left(\frac{x^i+1}{2}\right)^{\mathfrak{M}} \quad \textbf{(4)}$$

Далее, если не указано обратное, значение $\mathfrak{M}$ будет считаться равным единице, поэтому нижний индекс будет опускаться. Если значения x^i вычислялись по умолчанию, то $\delta^i = \varphi\left(1 - \frac{1}{2N}\right)$. При $N = 1$ в системе представлен только один источник документа d^i, и отсутствует какая-либо информация о его свойствах. Занижение значений δ^i по причине неполноты знаний об источниках документов, очевидно, повлияет и на достоверность фактов, в частности, ускоряя потерю актуальности. В предельно неблагоприятном случае (при $N = 1$), получим $\delta^i = \frac{\varphi}{2}$, вместо $\delta^i \approx \varphi$ (при $N \gg 1$).

Информация может со временем стать менее актуальной и, соответственно, менее заслуживающей доверия пользователя. Косвенным признаком утери актуальности факта является длительное отсутствие упоминаний факта в новых документах. Введем следующую функцию $h(t)$, зависящую от времени.

$$h(t) = \frac{1}{1+\ln(\frac{t}{\mathcal{M}}+1)}, \mathcal{M} = const \quad \textbf{(5)}$$

Будем называть $h(t)$ *темпоральным множителем*. Здесь $\mathcal{M}$ — это время, за которое значение достоверности понизится в l раз, из формулы (5) следует, что $l = 1 + ln2 \approx 1.69$. Таким образом, величина $\tau = \frac{t}{\mathcal{M}}$ — это безразмерное время, равное отношению времени существования факта в системе на время, необходимое для понижения его достоверности в l раз. Значение $\mathcal{M}$ подбирается исходя из оценки экспертом скорости устаревания фактов в данной предметной области.

За основу модели достоверности факта был взят неоднородный дискретный марковский случайный процесс, имеющий три состояния $\{E_1, E_2, E_3\}$, определяющих текущий уровень доверия к факту: «недоверие», «неопределенность» и «доверие» соответственно. Вероятность пребывания в третьем состоянии — это вероятность того, что факт заслуживает доверия. Остальные две вероятности

имеют вспомогательный характер. Моментами времени процесса считаем поступление очередного подтверждения факта, т.е. нового документа, упоминающего факт, X_n — случайная величина, выражающая состояние факта в момент времени *n*.

Обозначим через $\bar{\pi}^n$ вектор-строку распределения случайной величины X_n. Начальное распределение процесса задается предварительно: $\bar{\pi}^0 = (\pi_1^0, \pi_2^0, \pi_3^0), \pi_i^0 = P(X_0 = E_i)$. После n шагов вектор $\bar{\pi}^0$ переходит в $\bar{\pi}^n = \bar{\pi}^0 \cdot \mathbb{P}^{(n)}$, где $\mathbb{P}^{(n)} = (p_{jk}(0,n))$ — матрица перехода за n шагов, элементы которой вычисляются при помощи тождества Колмогорова-Чепмена, выполняющегося для любого $m < r < n$, в том числе для $r = n - 1$.

$$p_{jk}(m,n) = \sum_v p_{jv}(m,r) p_{vk}(r,n) \qquad \textbf{(6)}$$

Вероятности вида $p_{vk}(n, n+1)$ определяются матрицей перехода за один шаг или *переходной матрицей*, обозначаемой $\mathbb{P}(n, n+1)$.

$$\mathbb{P}(n,n+1) = \begin{bmatrix} 1-\delta^{n+1} & 0 & \delta^{n+1} \\ 1-\delta^{n+1} & 0 & \delta^{n+1} \\ \frac{1-p_{33}}{2} & \frac{1-p_{33}}{2} & p_{33} \end{bmatrix} \qquad \textbf{(7)}$$

$$p_{33} = \frac{1}{2}(p_{23}(0,n) + \delta^{n+1})$$

Для учета влияния времени существования факта в системе на вектор распределения было построено семейство линейных операторов, обозначаемое $\mathbf{T_t}$. Один из операторов семейства $\mathbf{T_t}$ применяется к вектору распределения, являющемуся результатом последнего, на тот момент, шага случайного процесса. Кроме того, этот же вектор, не претерпевший никаких темпоральных изменений, вовлекается в следующий шаг процесса, при условии, что факт не устарел и не был исключен из системы за прошедшее время. Как любое линейное преобразование операторы $\mathbf{T_t}$ можно записать в виде матрицы:

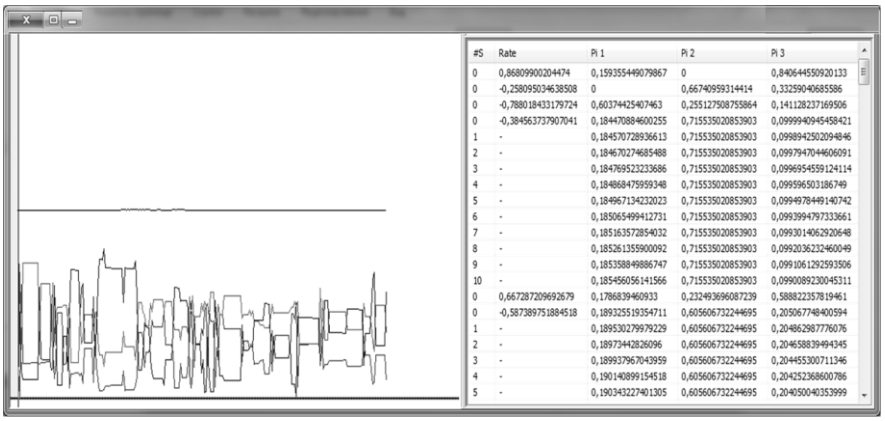
$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1-h(t) & 0 & h(t) \end{bmatrix} \qquad \textbf{(8)}$$

4.1 Калибровка параметров

Для проверки и калибровки параметров была проведена серия экспериментов. Напомним, что значения координат вектора распределения $\bar{\pi}$ зависят от двух параметров. Это показатель степени $\mathfrak{M}$ семейства функций (3) величина $\mathcal{M}$ из формулы вычисления темпорального множителя (5).

В общем виде зависимость распределения по времени представляет собой множество точек в четырехмерном евклидовом пространстве. Для удобства визуализации разобьем его на три двумерных зависимости — зависимость каждой из координат вектора распределения по времени.

Эксперимент показал, что при увеличении параметра $\mathfrak{M}$ вектор $\bar{\pi}$ сильнее реагирует на появление ненадежных источников с низкой экспертной оценкой. Однако, нежелательно, чтобы один отдельно взятый источник оказывал сильное влияние на $\bar{\pi}$, т.к. в этом случае ошибка при оценке источника может значительно исказить течение процесса. С учетом этих соображений оптимальным для $\mathfrak{M}$ очевидно является значение $\mathfrak{M} = 1$.



#S	Rate	Pi 1	Pi 2	Pi 3
0	0,868099002044474	0,159355449079867	0	0,840644550920133
0	-0,258095034638508	0	0,667409593144 14	0,332590406855 86
0	-0,788018433179724	0,603744254074 63	0,255127508755864	0,141128237169506
0	-0,384563737907041	0,184470884600255	0,715535020853903	0,099994094554821
1	-	0,184570728936613	0,715535020853903	0,099894250294846
2	-	0,184670274685488	0,715535020853903	0,099794704406091
3	-	0,184769523233686	0,715535020853903	0,099695455912411 4
4	-	0,184868475959348	0,715535020853903	0,099596503186749
5	-	0,184967134232023	0,715535020853903	0,099497844914074 2
6	-	0,185065499412731	0,715535020853903	0,099399479733366 1
7	-	0,185163572854032	0,715535020853903	0,099301406292064 8
8	-	0,185261355900092	0,715535020853903	0,099203623246004 9
9	-	0,185358849886747	0,715535020853903	0,099106129259350 6
10	-	0,185456056141566	0,715535020853903	0,099008923004531 1
0	0,667287209692679	0,178683946093 3	0,232493696087239	0,588822357819461
0	-0,587389751884518	0,189325519354711	0,605606732244695	0,205067748400594
1	-	0,189530279979229	0,605606732244695	0,204862987776076
2	-	0,189734428260 96	0,605606732244695	0,204658839494345
3	-	0,189939670439 59	0,605606732244695	0,204455300711346
4	-	0,190140899154518	0,605606732244695	0,204252368600786
5	-	0,190343227401305	0,605606732244695	0,204050040353999

Рис. 1 Экспериментальная калибровка параметров

По результатам экспериментов установлены следующие значения параметров: $\mathfrak{M} = 1$, $\mathcal{M} = 1000$ (предполагается, что время измеряется в сутках). На Рис. 1 показаны графики координат вектора $\bar{\pi}$ для случайного процесса длительностью 200 и вышеуказанными значениями параметров. В силу дискретности процесса каждый график представляет собой множество точек, поэтому для наглядности точки соответствующих графиков были соединены между собой отрезками, сформировав, таким образом, ломаные линии.

Параметр $\mathcal{M}$ не оказывает влияния на случайный процесс, его значение выбирается исходя из предпочитаемой скорости убывания актуальности факта во времени. Для различных видов фактов зависимость от времени, а, следовательно, и значение $\mathcal{M}$, может быть задано индивидуально.

4.2 Критерии удаления ненадежных данных

Изменение достоверности факта *F* в информационной системе описывается цепочкой пар $< j, \pi_j^F >$, где j — момент времени, π_j^F — достоверность в момент времени *j*, т.е. дискретным множеством. Принятие решения относительно факта на основе только текущего значения достоверности

неэффективно вследствие того, что достоверность может опуститься ниже минимально допустимого значения в случае погрешности при оценке, низкого авторитета выбранного источника и других возмущающих факторов, для уменьшения степени влияния подобных возмущений необходимо анализировать окрестность текущей точки. Анализ дискретных окрестностей также оказался неэффективен, т.к. не позволяет принять решение в случае колебаний достоверности вокруг среднего значения. В этом случае мы можем усреднить и оценить значения в промежуточных точках, аппроксимируя или интерполируя имеющееся множество точек, либо его подмножество, гладкой кривой. Отдельные сегменты кривой позволяют оценить уровень доверия в заданной окрестности текущего момента времени без учета влияния более ранней истории. Для решения этой задачи был проведен анализ различных методов аппроксимации и интерполяции кривыми и по его итогам выбран метод аппроксимации B-сплайном [8, 9].

Чтобы принять решение о дальнейшей судьбе факта, доверие к которому на текущий момент времени опустилось ниже минимально допустимого, предлагается выделять общую тенденцию поведения, основываясь при этом на анализе кривой аппроксимации хвоста из нескольких значений. С учетом соображений о вычислительной сложности было решено строить сплайн четвертого порядка без кратных вершин, при этом число опорных точек должно быть не меньше четырех, что и определило минимальный размер хвоста. Выбор хвоста минимально возможной длины для экспериментов обусловлен, во-первых, соображениями производительности, во-вторых — относительно малым количеством возможных видов кривых, и, кроме того, если вероятность погрешности при оценке одного значения достоверности равна p, то вероятность погрешности на четырех значениях составит p^4, что дает нам ~6% вероятности погрешности при $p = \frac{1}{2}$.

Какие тенденции может описывать кривая? Это может быть тенденция к убыванию, что говорит в пользу исключения факта, либо тенденция к возрастанию. Также кривая может не иметь выраженной тенденции. Введенная нумерация кривых отражает их форму и записывается в виде ***a.b.c,*** где $a \in A = \{1,2\}; b \in B = \{1,2,3,4,5,6\}; c \in C = \{1,2\}$.

A:
1. кривая имеет точку перегиба
2. кривая не имеет перегибов

B:
1. кривая возрастает
2. кривая убывает
3. кривая возрастает, а затем убывает
4. кривая убывает, а затем возрастает
5. кривая убывает, возрастает, затем снова убывает
6. кривая возрастает, убывает, затем снова возрастает

C:
1. кривая выпукла вниз в начальной точке
2. кривая выпукла вверх в нач. точке

Согласно такой нумерации кривая с номером, например, **1.2.1** — это кривая, имеющая точку перегиба, выпуклая вниз в некоторой окрестности начальной точки и убывающая на всей области определения. Хотя таким образом можно пронумеровать 24 кривые, всего их 16. Это обусловлено выбранным числом опорных точек и порядком кривой и подтверждено экспериментальной проверкой выборки из ~10 миллионов хвостов, сгенерированных 500 тысячами случайных процессов. Для нас интересны, в первую очередь кривые вида ***.1.*** и ***.2.***, поскольку они показывают общую тенденцию.

Строго убывающая кривая может уничтожить факт, строго возрастающая — предотвратить его удаление. Пусть *MIN* — минимально допустимый уровень доверия, при котором на факт еще можно положиться без какой-либо дополнительной проверки, F — некоторый факт, $E^j = \{e_1^j, e_2^j, e_3^j, e_4^j\}$ — j-й хвост соответствующего ему случайного процесса, e_k^j это последние четыре значения достоверности F. Рассмотрим граничные случаи.

Допустим, что $e_4^j < MIN;\ e_3^j \geq MIN$. Это первый граничный случай, при этом факт удаляется, если кривая выражает тенденцию к убыванию, т.е. имеет вид ***.2.***. Если кривая имеет какой-либо другой вид, факт остается в системе. Точно таким же образом разрешаются промежуточные случаи, когда $e_4^j < MIN;\ e_3^j < MIN, e_2^j \geq MIN$ или $e_4^j < MIN;\ e_3^j < MIN, e_2^j < MIN, e_1^j \geq MIN$. Второй граничный случай наступает, когда $e_k^j < MIN$ $(k = 1,2,3,4)$. Все значения находятся ниже минимального порога. Здесь на первый план выходят кривые вида ***.1.***. Соответственно, если кривая строго возрастает, то факт F все равно *не* будет удален. Если кривая имеет вид, отличный от указанного, то факт исключается из информационной системы как утративший доверие.

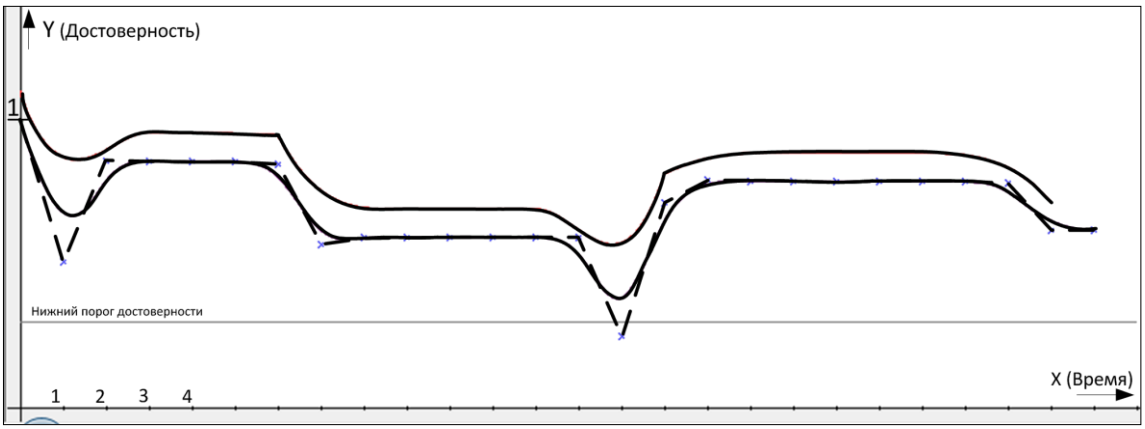


Рис. 2 Различия полной и кусочной аппроксимации

Новые хвосты могут вычисляться со смещением в одну точку. На Рис. 2 показана полная аппроксимирующая кривая в сравнении с кривой, составленной из отдельных хвостов [8, 10]. Пунктиром обозначена ломаная линия, проведенная через опорные точки. Для наглядности сегменты второй кривой построены со смещением в три точки по оси абсцисс, чтобы избежать наложения частей кривых друг на друга, и со смещением 0.1 по оси ординат (единица по шкале О*y* относится к единице шкалы О*x* как 20/3).

5 Заключение

Предлагаемые методы и подходы призваны обеспечить автоматическую обработку входящего потока данных и смоделировать изменение достоверности фактов в информационной системе, предметная область которой формально описана с помощью онтологии, а также описать механизм удаления ненадежной информации из системы.

Поиск референциальных связей между объектами и идентификация помогают отсеять нежелательные данные при пополнении системы, в то время как контроль достоверности и механизм отсеивания утративших доверие фактов способствуют сохранению целостности и непротиворечивости информации, прошедшей эту проверку. В особенности это касается фактов, требующих регулярного подтверждения. Примером таких фактов являются статьи кодексов (уголовного, гражданского, налогового и пр.). Каждое переиздание кодекса может подтвердить силу той или иной статьи, либо скорректировать или удалить ее, вводя новые. Данные, не подтверждаемые в течение долгого времени или подтверждаемые сомнительными источниками, постепенно будут удаляться из информационной системы.

Предлагаемые методы применяются при разработке информационной системы с документально подтверждаемой информацией. Ожидается, что результаты практического применения поспособствуют выявлению ошибок и недочетов, укажут на особенности и дадут опыт настройки процесса под различные виды фактов.

Литература

[1] Когаловский М. Р. Перспективные технологии информационных систем. — М.: ДМК Пресс; М: Компания АйТи, 2003. — 288 с.

[2] Когаловский М.Р. Системы доступа к данным на основе онтологий // Труды Второго симпозиума «Онтологическое моделирование», Казань 2010 – М: ИПИ РАН, 2011. – С. 45–78

[3] Загорулько Ю.А., Боровикова О.И., Кононенко И.С., Сидорова Е.А. Подход к построению предметной онтологии для портала знаний по компьютерной лингвистике. // Компьютерная лингвистика и интеллектуальные технологии: Труды международной конференции «Диалог 2006». М.: РГГУ, 2006.

[4] Heeyoung Lee, Yves Peirsman, Angel Chang, Nathanael Chambers, MihaiSurdeanu and Dan Jurafsky. Stanford’s Multi-Pass Sieve Coreference Resolution System at the CONLL-2011 Shared Task. // In Proc. of the 15th Conference on Computational Natural Language Learning: Shared Task. Portland. Oregon. USA. 2011. P. 28–34.

[5] Ермаков А.Е. Референция обозначений персон и организаций в русскоязычных текстах СМИ: эмпирические закономерности для компьютерного анализа. // Компьютерная лингвистика и интеллектуальные технологии: Труды международной конференции «Диалог 2005». М: Наука, 2005. С. 131–135.

[6] Серый А.С., Сидорова Е.А. Поиск референциальных отношений между информационными объектами в процессе автоматического анализа документов // Труды XIV Всероссийской научной конференции RCDL-2012 Электронные библиотеки: перспективные методы и технологии, электронные коллекции. – Переславль-Залесский, 2012. С.206–212

[7] Серый А.С., Сидорова Е.А. Идентификация объектов в задаче автоматической обработки документов. // Компьютерная лингвистика и интеллектуальные технологии: Труды

международной конференции «Диалог 2011». М.: РГГУ, 2011. С. 580-591.

[8] Ли, К. Основы САПР (CAD/CAM/CAE) / Кунву Ли. – СПб. : Питер, 2004. – 560 с.

[9] Роджерс, Д.Ф. Математические основы машинной графики: пер с англ. / Д. Роджерс, Дж. Адамс. – М.:Мир, 2001. – 604 с.

[10] Кокс, Д. Идеалы, многообразия и алгоритмы. Введение в вычислительные аспекты алгебраической геометрии и коммутативной алгебры : пер.с англ. / Д. Кокс, Дж. Литтл, Д. О`Ши ; под ред. В.Л. Попова. – М.: Мир, 2000. – 687 с.

Developing methods for maintaining data reliability in an information system based on facts

Alexey S. Sery

The paper will discuss methods and approaches for automating the process of the incoming data analysis in ontology based information systems where data is presented as a set of information objects. It is proposed how to establish a referential identity or co-reference between objects and how to maintain information reliability, which means defining its trust metric and monitoring up-to-dateness. The former depends on the trust metrics of information sources, the latter — on the lifetime mostly. The proposed trust management technique also includes removing spotted unreliable data from the system data storage, and by doing so reduces expert participation in the data verifying process and number of errors in the system.

Декларативная аналитика в мультидиалектной среде

Д.Ю. Ковалев
Институт проблем информатики РАН
г. Москва
dm.kovalev@gmail.com

Аннотация

Развитие наук с интенсивным использованием данных требует создания новых систем поддержки научных исследований. Подобные системы позволяют исследователям специфицировать задачи в терминах исследуемой предметной области, осуществлять масштабируемый анализ разнородных данных из различных источников, при этом поддерживая современные платформы больших данных. В Институте проблем информатики Российской Академии Наук исследуется мультидиалектная среда, позволяющая концептуально описывать научные задачи, при этом обеспечивая интероперабельность различных систем на правилах и средств интеграции данных. В данной работе представлен пример концептуальной спецификации задачи по анализу финансовых данных. Пример демонстрирует необходимость использования разных систем на правилах и обмена правилами между ними. Предлагаются усовершенствованный алгоритм выполнения концептуальной программы, а также соответствующий этому изменению подход для описания декларативной семантики работы среды.

1 Введение и описание проблемы

В течение последних нескольких лет в большинстве научных областей, а также в бизнесе произошел значительный скачок в количестве производимых и накопленных данных. Во многих научных областях работа с данными стала занимать значительную часть рабочего времени исследователей. Изменились сам формат и подход к научной деятельности, во главу угла была поставлены работа с данными. Это привело к возникновению новой парадигмы в науке, а сами такие разделы науки были названы науками с интенсивным использованием данных (data-intensive sciences (DIS)) [2, 17]. Примерами научных направлений с интенсивным использованием данных являются астрономия, науки о Земле, молекулярная биология [17].

Труды 15-й Всероссийской научной конференции «Электронные библиотеки: перспективные методы и технологии, электронные коллекции» — RCDL-2013, Ярославль, Россия, 14-17 октября 2013 г.

Новая парадигма требует создания новых систем и средств поддержки всего цикла научных исследований, начиная от получения данных и заканчивая анализом данных и визуализацией полученных результатов [2]. При создании подобной системы неизбежно предстоит решить следующие задачи:

- интеграция разнородных, распределенных информационных ресурсов;
- разработка масштабируемых алгоритмов для работы в распределенной и параллельной средах;
- описание задачи в терминах предметной области (концептуально), сокращение кода программы;
- избавление пользователя от необходимости вручную распараллеливать алгоритмы.

При работе с большими объемами данных алгоритмы анализа данных реализуются над распределенной инфраструктурой. Это могут быть как параллельные машины баз данных, так и Hadoop. Такая реализация не требует изменений кода программы при увеличении объема данных.

Существует два подхода к обеспечению масштабируемости аналитики [15]. Целью первого подхода является создание параллельной среды выполнения программы для языка достаточно высокого уровня (R, Matlab). Примерами такого подхода могут служить System ML [11], Revolution Analytics[22], Snow [30]. Для императивных языков распараллеливание производится вручную (например, с помощью MPI-интерфейса). Для декларативных языков эта же задача решается автоматически самой системой. Например, в System ML программы на R-подобном языке автоматически транслируются в программу над Hadoop.

Целью второго подхода является предоставление пользователю системы со встроенными примитивами низкого уровня совместно с примитивами для координации выполнения программы. Примерами такого подхода являются Apache Mahout [5], SciDB [29], MADlib [15].

Другой сложностью, возникающей в DIS, являются разнообразие форматов и моделей данных, представлений и способов хранения большого объема данных, а также множественность распределенных источников данных. Все это приводит к необходимости интеграции разнородных данных. При реализации материальной и виртуальной интегра-

ции данных обычно используются подходы GAV [31, 14] и LAV [14] или их комбинация GLAV [9].

Реализация информационных систем в DIS осуществляется в комбинированных архитектурах ИТ средств, включающих платформы поддержки больших данных, суперкомпьютеры, многопроцессорные системы, грид и облачные архитектуры [2]. Спецификация и реализация распределенных программ анализа данных над такой комбинированной распределенной архитектурой являются нетривиальной задачей [16]. Низкоуровневые примитивы и абстракции императивных языков программирования неэффективны в таких архитектурах для выражения алгоритмов задач анализа данных, требующих концептуального представления, независимого от конкретных данных и обеспечивающих возможность их повторного использования в других применениях. В качестве альтернативы использованию императивных языков, подобно семантическому Вебу [27], для решения задач анализа данных представляется целесообразным использовать языки на правилах. Помимо естественной концептуализации проблем в таких языках разнообразие их семантик является существенным расширением возможностей традиционных методов анализа данных при решении сложных задач. Также декларативные языки являются хорошо распараллеливаемыми [16].

В результате формулирования проблем и требований к новым системам поддержки DIS были выявлены некоторые свойства, наличие которых необходимо учитывать при их создании [2, 16]. Такие системы должны сочетать в себе средства интеграции и анализа данных, позволяющие эффективным образом выражать аналитические алгоритмы в терминах концептуального представления предметных областей. Кроме того, рост объема данных приводит к тому, что функции анализа необходимо выполнять как можно ближе к данным. Системы должна быть легко разворачиваемы над комбинированными распределенными ИТ-средствами, сохраняя при этом приемлемую эффективность работы.

Ряду перечисленных требований соответствует разрабатываемая в ИПИ РАН среда, позволяющая декларативным образом концептуально описывать задачи из различных предметных областей [18]. В ней использован высокоуровневые подходы к программированию задач на основе языков на правилах в сочетании со средствами интеграции данных, также выражаемых декларативно.

Первый из подходов технически базируется на использовании стандарта W3C по обмену правилами Rule Interchange Format (RIF) [27]. Этот стандарт позволяет обеспечить синтаксическую и семантическую интероперабельность программ, представленных в различных языках на правилах, на основе стандартизованных диалектов RIF. Согласно стандарту, для каждой из конкретных систем на правилах требуется построить отображение из языка системы в подходящий диалект RIF и обратно. Отображение должно сохранять семантику отображаемых языков. Правила передаются от одной системы к другой на промежуточном языке, в качестве которого используется некоторый диалект RIF.

В существующей редакции RIF определено несколько диалектов. Так, RIF-BLD является основным логическим диалектом в стандарте RIF и соответствует хорновским правилам с некоторыми синтаксическими и семантическими расширениями [24]. Диалект RIF-PRD [28] обобщает примитивы различных продукционных систем. Диалект RIF-Core разрабатывался с целью максимизации пересечения продукционных и логических диалектов. Существуют и другие диалекты RIF, пока не являющиеся рекомендацией W3C: CASPD [25] – для систем, обладающих семантикой стабильных моделей [10], CLPWD [26] – для систем с хорошо-обоснованной семантикой [33]. По причине того, что при спецификации задач может быть использовано несколько диалектов RIF, разрабатываемая среда получила название мультидиалектной.

Средства интеграции данных в рассматриваемой среде обеспечивают виртуальную интеграцию и совместное использование неоднородных ресурсов на основе технологии предметных посредников [34, 20]. Посредник располагается между пользователем, который формулирует проблемы концептуально, независимо от ресурсов, и разнородными информационными ресурсами.

Несмотря на то, что уже сейчас в этой среде можно решать интересные задачи [18], учитывая требования задач анализа данных в DIS, требуется развить ее, встроив средства аналитики и сохранив при этом высокий уровень используемых абстракций и примитивов.

Статья структурирована следующим образом: в разделе 2 кратко описана существующая среда, в разделе 3 представлены основная цель исследования и сопутствующие задачи. Раздел 4 описывает подход к проблеме и основные идеи работы. Наконец, в разделе 5 представлено заключение.

2 Мультидиалектная среда

Программой на правилах называется некоторый конечный набор правил. Движок, который осуществляет выполнение программы на правилах, называется системой на правилах. Ключевым при выводе является отношение выполнимости формул. Логический язык имеет теоретико-модельную семантику, согласно которой определяется, какие модели существуют для каждой программы. Существует несколько различных семантик для работы с отрицанием. Основными являются стратифицированный даталог [32], семантика стабильных моделей и хорошо-обоснованная семантика.

Среда [18] обеспечивает взаимную работу средств интеграции и систем вывода на правилах. Она состоит из нескольких узлов, осуществляющих отображение из языка системы в диалект RIF и наоборот. В среду также входят супервизор, отвечающий за исполнение программы на правилах, и посредник. Так как посредник специфицируется на логическом языке, можно относиться к нему как к

полноправному узлу. Задача формулируется и решается в терминах предметной области (описанной концептами из подключенной онтологии). Программа, написанная в концептах предметной области, называется концептуальной. При добавлении информации о принадлежности правил и предикатов конкретному узлу системы концептуальная программа становится распределенной. При исполнении программы происходят следующие действия: 1) переписывание концептуальной программы в распределенную с добавлением для каждого правила и предиката в правиле, соответствующего ему имени узла, 2) рассылка частей программы на узлы, 3) исполнение программы на узлах, 4) получение ответа.

Ключевым механизмом при исполнении программы является делегирование [1] – передача правил и фактов от одного узла к другому. Ответственным за передачу правил и фактов является супервизор среды. На данный момент осуществлена лишь частичная поддержка механизма, например, можно обмениваться фактами.

3 Цель и задачи исследования

Целью работы является исследование и развитие программной среды для сопровождения научных исследований, в которой концептуально и декларативно используются разнообразные инструменты – языки и системы на правилах, средства интеграции данных, аналитические методы.

Для достижения поставленной цели необходимо решить ряд задач.

Во-первых, требуется определить задачу, решение которой продемонстрирует необходимость использования нескольких систем на правилах с разной семантикой, а также обмена правилами между системами вывода.

Во-вторых, требуется развить и автоматизировать алгоритмы переписывания программы из концептуальной в распределенную. При этом отображение должно сохранять отношение логического следования программ. Другим направлением развития является совершенствование алгоритма выполнения распределенной программы. Необходимо сформулировать требования, которым должен удовлетворять этот алгоритм, реализовать его в соответствии с этими требованиями, провести ряд экспериментов по сравнению эффективности предложенного алгоритма с другими подобными системами.

В-третьих, необходимо описать декларативную (теоретико-модельную) семантику выполнения распределенной программы и провести доказательство того, что все полученные модели являются моделями и для концептуальной программы, т. е. установить отношение выполнимости программ.

В-четвертых, среду планируется расширить в сторону поддержки хранилищ данных. При этом они могут содержать в себе инструменты аналитики. При такой интеграции требуется сохранить декларативность и концептуальность подхода. Среда должна учитывать возможности современных платформ, а также распределенных сред – грид и облаков. Необходимо продемонстрировать возможность эффективного использования предлагаемого подхода в этих средах.

В-пятых, требуется оценить сложность выполнения распределенной программы и определить насколько сильно предложенная среда усложняет решение задач.

4 Предлагаемый подход

4.1 Пример задачи для решения в мультидиалектной среде

Для лучшего понимания необходимости использования мультидиалектной среды предложена задача, иллюстрирующая преимущества использования нескольких систем вывода с разными семантиками и обмена правилами между ними [18].

Требуется построить диверсифицированный портфель максимального размера. Портфелем называется множество ценных бумаг, таких как акции и облигации. Диверсифицированность означает, что движение цены бумаги не зависит от других бумаг. Таким образом, снижаются риски сильного падения совокупной цены портфеля.

Прежде всего, требуется решить проблему интеграции данных из различных финансовых источников (google finance, yahoo finance) и проблему по построению портфеля максимального размера. Показано, что последняя проблема сводится к поиску максимальной клики в графе. Данная задача является NP-сложной. Для решения подобных задач (по скорости выполнения и простоте написания программы) хорошо подходят ASP системы на правилах со стабильной семантикой моделей, например система DLV. Для интеграции данных из разных источников используется логический язык СИНТЕЗ. Взаимодействие систем вывода в мультидивалектной среде между собой представлено на рис. 1.

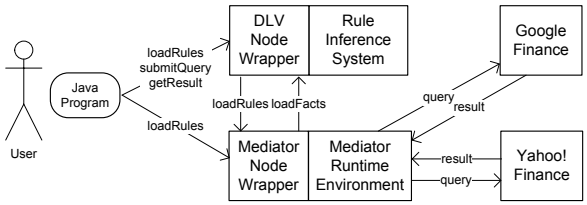


Рис. 1. Мультидиалектная среда для задачи поиска диверсифицированного портфеля

Концептуальная программа определяется на диалектах RIF-BLD (интеграция данных) и RIF-CASPD (поиск максимальной клики). Задачу поиска максимальной клики можно считать одной из класса задач на графах. При использовании языка DLV спецификация задачи о поиске максимальной клики становится простым и практически тривиальным действием. Описание поиска происходит концептуально и декларативно:

```
Document( Dialect(RIF-CASPD)
  ...
```

```
Group (
Forall ?X(Or(prt:portfolio(?X)
  prt:nonPortfolio(?)):-tickers@gex(?X))
Forall ?X ?Y( :- And(prt:portfolio(?X)
  prt:nonPortfolio(?X)))
Forall ?X ?Y( :- And(prt:portfolio(?X)
  prt:portfolio(?Y) (Naf noncorrelat-
  ed@gex(?X ?Y))) )
Forall ?X( :~ prt:nonPortfolio(?X))  ) )
```

Формальное описание процесса решения задачи представлено в [18].

Задачу поиска максимальной клики можно считать одной из класса задач на графах. Подход, подобный изложенному, можно применять и для других задач из этого класса, а также других классов задач.

Второй задачей является интеграция данных из различных источников. Для ее решения используется концепция предметных посредников и язык СИНТЕЗ [20]. Спецификация осуществляется на диалекте RIF-BLD:

```
Document( Dialect(RIF-BLD)
  ...
  Group(
  Forall ?t ?tp ?symbol, ?ticker(
    Exists ?ts( And(?ts#gex:tickers
      ?ts[symbol -> ?symbol]) ):-
      And(?t#srt:stockRates ?t[
        ticker->?symbol])
  Forall ?m ?n ?ticker1 ?ticker2 ?start
    ?end ?rates1 ?rates2 ?dv1 ?dv2 ?date1
    ?date2 ?series1 ?series2 ?corr (
      Exists ?e (
        And(?e#gex:noncorrelated ?e[
          start->?ticker1 end->?ticker2])):-
          ?m#srt:stockRates  ?m[
            ticker->?ticker1  rates->?rates1]
          ?n#srt:stockRates  ?n[
            ticker->?ticker2  rates->?rates2]
          ?dv1#?rates1  ?dv1[date -> ?date1]
          ?dv2#?rates2  ?dv2[date -> ?date2]
          External(pred:date-greater-than-or-
            equal(?date1  2012-01-01))
          External(pred:date-less-than-or-
            equal(?date1  2012-12-31))
          ?c#srt:correlation ?c[corr->?corr
            series1->?rates1 series2
              ->?rates2]
          External(pred:numeric-greater-
            than(?corr  -0.25)
          External(pred:numeric-less-
            than(?corr  0.25))
        External(pred:numeric-less-
            than(?ticker1  ticker2))  )))
```

Подробное описание правил и взаимодействия между модулями представлено в [18]. Вся программа занимает всего 6 правил. В мультидиалектной среде становится возможным выделить несколько частей программы, для которых требуется различные системы на правилах с разной семантикой, при этом обеспечив их взаимодействие посредством обмена правилами через диалекты RIF.

4.2 Алгоритм переписывания и исполнения распределенной программы

На данный момент предлагаемая среда обладает рядом ограничений. Фактически при исполнении алгоритма на каждом из узлов программа исполняется не более одного раза. Предлагается определить вычисление распределенной программы как последовательность локальных вычислений на узлах среды [3, 21]. Узлы среды активируются один за другим в каком-либо порядке. Это происходит, пока не достигнута неподвижная точка (fixpoint) распределенной программы. В течение каждого локального исполнения части программы узел получает и отправляет сообщения, а также совершает некоторый логический вывод, определенный системой вывода в данном узле. Последовательность вычислений не прекращается до тех пор, пока не достигнута неподвижная точка.

Пусть I_L – конечный набор фактов на узле L, P_L – часть распределенной программы на узле L, $Facts_{rcv}$ – набор фактов, полученных от других узлов, а $Rules_{rcv}$ – множество соответствующих полученных правил. Пусть также *induc(P)[I]* – это набор edb-фактов, принадлежащий данному узлу и полученных в результате выполнения программы P над базой данных I; *facts(P)[I]* – набор edb-фактов, не принадлежащих данному узлу, *rules(P)[I]* – набор правил, не принадлежащих данному узлу. Тогда в результате выполнения программы получится набор фактов, которые пополнят локальную базу данных

$$J = I \cup induc(P_L \cup Rules_{rcv})[I \cup Fatcs_{rcv}],$$

а также набор фактов и правил, которые необходимо отправить на соответствующие узлы:

$$Fatcs_{snd} = facts(P_L \cup Rules_{rcv})[I \cup Fatcs_{rcv}]$$

$$Rules_{snd} = rules(P_L \cup Rules_{rcv})[I \cup Fatcs_{rcv}]$$

Соответственно после отправки сообщений запустится очередной локальный шаг вычислений. Последовательность локальных вычислений строится следующим образом:

$$\rho_1 \xrightarrow[Facts,Rules]{} \rho_2 \xrightarrow[Facts,Rules]{} \cdots \xrightarrow[Facts,Rules]{} \rho_n.$$

Алгоритм завершается, если набор фактов и правил, подлежащих отправлению, пуст.

На данном этапе в среде предполагается передача фактов коллекцией. Предлагается усовершенствовать алгоритм в соответствии с предложенной схемой.

Внедрение этого алгоритма выполнения распределенной программы позволит исполнять произвольные программы в мультидиалектной среде. Кроме того, становится возможным в соответствии с предложенным алгоритмом описать теоретико-

модельную семантику исполнения программы и доказать ее соответствие семантике диалекта RIF.

Обмен правилами между продукционными системами на правилах с использованием диалекта RIF-PRD представлен в работах [12, 8]. Основным отличием является то, что продукционные системы работают с единой операционной семантикой и, следовательно, пропадает проблема совместной работы систем на правилах с различной логической семантикой.

Автору не известны упоминания в литературе описаний механизмов обмена правилами между системами с операционной и декларативной семантиками в RIF. Однако существует ряд работ по использованию языков баз данных для спецификации декларативных распределенных программ и манипулирования данными в распределенной среде [21, 13, 3, 1]. В отличие от мульти-диалектного подхода, в упоминаемых системам используется единый язык на правилах на всех узлах. Обычно это даталог с отрицанием с добавлением понятия локализации – принадлежности факта или правила определенному узлу. В отличие от этих языков, в предложенном подходе используются стандартные, определенные в RIF удаленные и импортированные термы.

4.3 Семантика

При переписывании концептуальной программы в распределенную может измениться набор выводимых моделей. Определение декларативной семантики распределенной программы позволит установить отношение логического следования между ней и концептуальной программой.

Для переписанной программы P_{dist} из концептуальной P_{conc} должно быть верно отношение логического следования $P_{conc} \models P_{dist}$, то есть все модели P_{dist} должны быть и моделями P_{conc}.

Для достижения этого может быть применено несколько подходов. Можно определить теоретико-модельную семантику концептуальной и распределенной программ и установить отношение логического следования. Другим вариантом является определение набора операций переписывания программы, сохраняющих семантику. Остается определить, что это за операции и каким образом установить сохранение семантики при отображении.

Для большинства распределенных языков на правилах не определяется декларативная семантика по причине наличия нелогических конструкций. Исключением является язык Dedalus [3], для которого определена полностью декларативная семантика.

5 Заключение

Смена парадигмы в некоторых областях науки требует создания современны средств и систем поддержки научных исследований. В ИПИ РАН исследуется система, позволяющая декларативно специфицировать научные задачи. С целью ее развития в данной работе представлены пример задачи по анализу финансовых данных, новый алгоритм выполнения распределенной программы и соответствующий ему подход по определению декларативной семантики программ. Пример демонстрирует потенциальные возможности мультидиалектной среды, в том числе интеграцию данных из различных источников и декларативный анализ полученных данных.

Литература

[1] Abiteboul S., Bienvenu M., Galland A. et al. A rule-based language for Web data management. In: Proc. 30th ACM Symposium on Principles of Database Systems, ACM Press, 2011. – P. 283–292.

[2] Agrawal D., Bernstein P., Bertino E., Davidson S., Dayal U., Franklin M., Gehrke J., Haas L., Halevy A., Han J., Jagadish H.V., Labrinidis A., Madden S., Papakonstantinou Y., Patel J. M., Ramakrishnan R., Ross K., Shahabi C., Suciu D., Vaithyanathan S., Widom J. Challenges and Opportunities with Big Data. A community white paper developed by leading researchers across the United States, 2012. – http://cra.org/ccc/docs/init/ bigdatawhitepaper.pdf

[3] Alvaro P., Marczak W.R. et al. Dedalus: Datalog in time and space// Technical Report EECS-2009-173, University of California, Berkeley, 2009.

[4] Apache Hadoop. – http://hadoop.apache.org/.

[5] Apache Mahout. – http://mahout.apache.org/.

[6] Apache Pig. – http://pig.apache.org/.

[7] Beyer K.S., Ercegovac V., Gemulla R., Balmin A., Eltabakh M., Kanne C.C., Shekita E.J. Jaql: A scripting language for large scale semistructured data analysis// In: Proc. of the VLDB Endowment, 2011. – V. 4, No 12. – P. 1272-1283.

[8] Cosentino V., Del Fabro M. D., El Ghali A. A model driven approach for bridging ILOG Rule Language and RIF. In: Proc. of the 6th International Symposium on Rules, RuleML2012, 2012. – CEUR-ws. – V. 874. – P. 96-102.

[9] Friedman M., Levy A., Millstein T. Navigational plans for data integration. In: National Conference on Artificial Intelligence (AAAI) Proc., 1999.

[10] Gelfond M., Lifschitz V. The Stable Model Semantics for Logic Programming. In: Proc. Fifth Intl. Conference and Symposium Logic Programming, MIT Press, Cambridge, 1988. – P. 1070–1080.

[11] Ghoting A., Krishnamurthy R., Pednault E. et al. SystemML: Declarative machine learning on MapReduce. In: ICDE, 2011. – P. 231–242.

[12] Gonzalez-Moriyon G. Final steel industry public demonstrators// ONTORULE Deliverable D5.5, 2012.

[13] Grumbach S., Wang. F. Netlog, a rule-based language for distributed programming. In: (Eds.) M. Carro and R. Pena, Proc. 12th International Symposium on Practical Aspects of Declarative Languages, LNCS, 2010. – V. 5937. – P. 88–103.

Halevy A. Y. Answering queries using views: A survey. In: VLDB J., 2001. – V. 10, No. 4.

[14] Hellerstein J. M., Ré C., Schoppmann F., Wang D. Z., Fratkin E., Gorajek A., Kumar A. The MADlib analytics library: or MAD skills, the SQL. In: Proc. of the VLDB Endowment, 2012. – V. 5, No. 12. – P. 1700-1711.

[15] Hellerstein J. The declarative imperative: experiences and conjectures in distributed logic. In: ACM SIGMOD Record, 2010. – V. 39, No. 1. – P. 5–19.

[16] Hey T., Tansley S., Tolle K. (Eds.). The Fourth Paradigm: Data-intensive Scientific Discovery. Microsoft Research, Redmond, Washington, 2009.

[17] Ihaka R., Gentleman R. R: A language for data analysis and graphics. In: Journal of computational and graphical statistics, 1996. – V. 5, No. 3. – P. 299-314.

[18] Kalinichenko L., Stupnikov S., Vovchenko A., Kovalev D. Rule-based Multi-dialect Infrastructure for Conceptual Problem Solving over Heterogeneous Distributed Information Resources. In: Kacprzyk, Janusz (eds.), Advances in Intelligent Systems and Computing, 2013. – V. 241. – P. 61-68.

[19] Kalinichenko L.A., Stupnikov S.A., Martynov D.O. SYNTHESIS: A language for canonical information modeling and mediator definition for problem solving in heterogeneous information resource environments. In: M. IPI RAS, 2007.

[20] Loo B. T., Condie T., Garofalakis M., Gay D. E., Hellerstein J. M., Maniatis P., Ramakrishnan R., Roscoe T., Stoica I. Declarative networking: language, execution and optimization. In: ACM SIGMOD Conference Proceedings, 2006. – P. 97–108.

[21] Revolution Analytics. – http://www.revolutionanalytics.com/

[22] RHadoop package. – http://github.com/RevolutionAnalytics/RHadoop/

[23] RIF Basic Logic Dialect (Second Edition)// (Eds.) H. Boley, M. Kifer.W3C Recommendation, 2013.

[24] RIF Core Answer Set Programming Dialect// (Eds.) S. Heymans, M. Kifer, 2009. – http://ruleml.org/rif/RIF-CASPD.html

[25] RIF Core Logic Programming Dialect Based on the Well-founded Semantics// (Ed.) Michael Kifer RuleML specification, 2010. – http://ruleml.org/rif/RIF-CLPWD.html

[26] RIF Overview (Second Edition)// (Eds.) H. Boley, M. Kifer. W3C Working Group Note, 2013.

[27] RIF Production Rule Dialect// (Eds.) Christian de Sainte Marie, Gary Hallmark, Adrian Paschke. W3C Recommendation, 2013. – http://www.w3.org/TR/2013/REC-rif-prd-20130205/

[28] Stonebraker M., Brown P., Poliakov A., Raman S. The architecture of SciDB. In: Proc. of the 23rd international conference on Scientific and statistical database management SSDBM, 2011. – P. 1–16.

[29] Tierney L., Rossini A. J., Li N. Snow: A parallel computing framework for the R system. In: International Journal of Parallel Programming, 2009. – V. 37, No. 1. – P. 78-90.

[30] Ullman J. D. Information integration using logical views. In: 6th International Conference on Database Theory (ICDT'97) Proc., 1997.

[31] Ullman J. D. Principles of Database and Knowledge-Base Systems. W. H. Freeman & Co., New York, 1990. – V. 2.

[32] Van Gelder A., Ross K. A., Schlipf J. S. The well-founded semantics for general logic programs. In: Journal of the ACM (JACM), 1991. – V. 38, No. 3. – P. 619-649.

[33] Д.О. Брюхов, А.Е. Вовченко, В.Н. Захаров, О.П. Желенкова, Л.А. Калиниченко, Д.О. Мартынов, Н.А. Скворцов, С.А. Ступников. Архитектура промежуточного слоя предметных посредников для решения задач над множеством интегрируемых неоднородных распределённых информационных ресурсов в гибридной грид-инфраструктуре виртуальных обсерваторий. Информатика и её применения, 2008. – т. 2, вып. 1. – с. 2-34.

Declarative Analytics in Multidialect Infrastructure

D. Kovalev

Data intensive sciences require new systems to support scientific research, starting from data acquisition and finishing with data analysis and visualization. At the Institute of problems of informatics RAS a new multidialect infrastructure is investigated to satisfy these needs. Problems could be specified conceptually. System supports interoperability of rule systems and integration facilities. A use-case of conceptual specification of a problem in the financial domain is presented.