Метод обнаружения изменений структуры веб-сайтов в системе сбора новостной информации

© А.М. Андреев © Д.В. Березкин © И.А. Козлов © К.В. Симаков
МГТУ им. Н.Э. Баумана, г. Москва
arkandreev@gmail.com dmitryb2007@yandex.ru kozlovilya89@gmail.com skv@ixlab.ru

Аннотация

Работа посвящена решению задачи обнаружения сбоев в работе системы сбора новостной информации, вызванных изменениями структуры веб-сайтов. Приведены модели документа и набора документов, предложен двухступенчатый метод обнаружения сбоев. Проведена экспериментальная оценка и даны направления по дальнейшему усовершенствованию предложенного подхода.

1 Введение

Сбор текстовой информации из открытых Интернет-источников, ее унификация и накопление, являются задачами, с которыми приходится сталкиваться при разработке промышленных систем интеллектуальной обработки текстов, например, класса Text Mining. Без наличия актуальной базы, постоянно пополняющейся текстами целевой предметной области, невозможно эффективно использовать методы автоматической обработки (такие как кластеризация, полнотекстовый поиск, выявление скрытых зависимостей).

В данной работе рассматривается решение задачи качественного сбора новостной информации. К такой информации относится текст новости, а также сопутствующие метаданные, включающие название, дату публикации, автора новости и др. Под качественным сбором в первую очередь подразумевается очистка текста новости от окружающей его служебной информации: меню сайта, рекламные баннеры, блоки социальных сетей, комментарии пользователей и т.д.

Основной акцент в данной статье делается на проблеме своевременного обнаружения изменения структуры опрашиваемых веб-сайтов, поэтому предлагаемый подход может быть использован не только для обработки новостных сайтов, он также распространяется на сбор сообщений из электронных библиотек, блогов, форумов и социальных сетей.

Труды 14-й Всероссийской научной конференции «Электронные библиотеки: перспективные методы и технологии, электронные коллекции» — RCDL-2012, Переславль-Залесский, Россия, 15-18 октября 2012 г.

2 Постановка задачи

2.1 Функционирование системы сбора

Существует множество подходов к организации сбора открытых текстовых материалов с веб-сайтов. Как правило, система сбора использует информацию об HTML-разметке целевых страниц для поиска в них нужной информации [12]. Эта информация используется правилами распознавания, записываемыми на принятом в системе формальном языке. Распространение получили как ручной способ описания правил, когда правила распознавания формирует программист [13], так и автоматизированный способ, когда правила формируются автоматически на основе обучающей выборки, подготовленной оператором [1,2,8]. Имея набор правил, система сбора выполняет периодический опрос веб-сайтов в поисках новых материалов.

В данной работе рассматривается система, выполняющая сбор новостей на основе правил, заданных вручную программистом. Основные функциональные элементы системы сбора представлены на рис. 1.

Рис.1. Функционирование системы сбора.

Система сбора выполняет чтение RSS-ленты сайта, откуда извлекается метаданные о каждой новости: название, аннотация, время публикации и URL текста новости. Далее по полученному URL осуществляется чтение страницы с текстом новости, выполняется построение DOM-модели этой страницы, откуда и выполняется извлечение чистого текста на основе имеющихся XPath правил. Результат сбора представляет собой чистый текст новости и XML-документ с метаданными. Далее эта информация заносится в базу данных, где осуществляется накопление и аналитическая обработка собираемых данных. Кроме этого, система выполняет постоянную регистрацию и накопление статистической информации о состоянии и структуре опрашиваемого веб-сайта.

2.2 Задача обнаружения сбоев

Все методы сбора информации из веб-сайтов, использующих особенности разметки страниц, объединяет то, что при изменении верстки сайта, возникает необходимость перенастраивать правила распознавания. При выполнении круглосуточного опроса целевых сайтов, своевременность обнаружения изменения верстки является весьма актуальной задачей, поскольку система сбора фактически перестает работать до тех пор, пока оператор не откорректирует набор правил распознавания.

В простейшем случае при существенном изменении структуры сайта система сбора станет выдавать в качестве результата пустые текстовые документы. Однако существуют достаточно сложные ситуации, когда при изменении верстки система сбора начинает извлекать тексты не полностью, либо фрагменты из других участков сайта, например, комментарии пользователей. Именно выявлению таких нетривиальных ситуаций посвящена данная статья.

2.3 Существующие подходы к решению задачи

В работах, посвященных теме выявления сбоев систем извлечения данных [7,9,11], представлено несколько подходов к решению вышеуказанной задачи. Большинство из них основано на оценке статистических характеристик документов, извлекаемых системой. При этом оценке может подвергаться как отдельно взятый документ [7] (в этом случае вычисляется вероятность его корректности, которая затем сравнивается с задаваемым пользователем пороговым значением), так и их набор [11] (оценке подвергается схожесть законов распределения случайных величин, соответствующих характеристикам документов из обучающей и тестовой выборок. Для сравнения используется критерий согласия Пирсона [20]).

В [11] также представлен подход, основанный на использовании методов machine learning для обучения системы обнаружения сбоев на наборах корректных документов для последующего определения правильности её работы на новых данных. В качестве таких методов используется, в частности, одноклассовая классификация (выявление аномалий) [5].

3 Принцип обнаружения сбоев

Для распознавания сбоев, связанных с изменением верстки, в систему сбора встраивается подсистема, осуществляющая контроль корректности поступающих документов и выявляющая сбои в верстке документов. Возможны два следующих подхода к обнаружению сбоев.

1. Анализ одной загруженной веб-страницы. Суть данного подхода заключается в использовании классификатора, который определяет принадлежность веб-страницы к классу корректных или некорректных страниц. В своей работе классификатор использует набор выделяемых из веб-страницы признаков. Обучение классификатора осуществляется на предопределенных наборах веб-страниц обоих классов. Преимуществом такого подхода является высокая скорость реакции детектора на сбой: «плохой» документ будет выявлен непосредственно после его поступления. Однако этот метод имеет и серьезный недостаток. Статьи, подвергающиеся анализу, могут сильно отличаться друг от друга. Так, иногда на вход детектора поступают «хорошие», но нетипичные для данного источника новости. Если подобных документов не было в обучающей выборке классификатора, они не могут быть корректно распознаны, и в результате происходит ложное срабатывание. При накапливании корректных документов и увеличении обучающей выборки частота возникновения таких ошибок постепенно уменьшается, но они продолжают периодически возникать.
2. Анализ контрольной серии из нескольких последних загруженных веб-страниц. Данный подход позволяет избавиться от ложных срабатываний. Даже если в контрольную серию попало несколько подозрительных статей, то усредненные характеристики этой коллекции останутся близкими к характеристикам эталонной обучающей выборки. Если же сомнительные документы будут поступать от источника регулярно, то через некоторое время, когда в контрольной серии таких статей будет накоплено достаточное количество, они будут составлять значительную долю анализируемого набора. В результате характеристики контрольной серии изменятся, и мы сможем обнаружить сбой. Такой подход к фиксации сбоев более надежен. Причём качество проверки будет возрастать с увеличением количества документов в контрольной серии. Но это приведёт к возникновению значительной задержки между моментом, в который произошел сбой, и временем его обнаружения.

Предложенный в данной работе метод, сочетает преимущества двух вышеописанных подходов (см. рис. 2): быструю реакцию на сбой и высокое качество проверки.

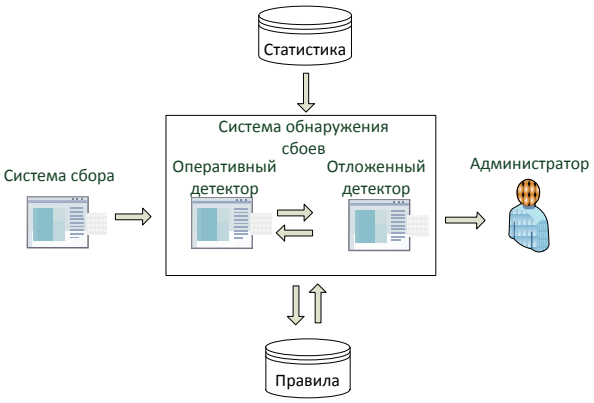

Рис. 2. Предложенный подход.

Данный метод положен в основу подсистемы контроля корректности загружаемых документов. Подсистема представляет собой двухступенчатый детектор сбоев. Один из его компонентов – «опера-

тивный детектор» – проверяет документы непосредственно в момент их поступления и делает предварительный вывод о вероятности сбоя. Если вероятность высока, выполняется проверка «отложенным детектором», уточняющая этот результат.

4 Предложенные модели документов

В основе системы обнаружения сбоев лежит модель анализируемых данных. Два основных компонента системы работают с разными входными данными и анализируют различные характеристики, поэтому для каждого из них предложена своя модель: модель документа, подвергающаяся обработке «оперативным детектором» и модель набора документов, анализируемая «отложенным детектором».

4.1 Модель документа

Под моделью документа понимается совокупность его характеристик, учитываемых «оперативным детектором» при его обработке. При создании детектора для системы сбора новостей выбор параметров производился с учетом некоторых особенностей функционирования системы. Статья извлекается из веб-страницы, где текст обычно разбит на параграфы (html-элемент <p>). Также внутри текстовых параграфов могут встречаться стилевые элементы разметки. С учётом этих факторов для оценки корректности новостей были выбраны следующие характеристики:

- объем веб-страницы, содержащей статью (*P*);
- суммарный размер параграфов статьи (*S*). Учитывается только текст, без html-элементов;
- количество параграфов в статье (*N*);
- дисперсия размера параграфа в рамках статьи (*V*);
- количество html-элементов различных типов, включенных в новость. Для сокращения типов html-элементов, они были сгруппированы по нескольким категориям. Были выделены классы наиболее часто встречающихся элементов: «Гиперссылки (H)» (в этот класс попал элемент href), «Текстовые блоки (B)» (br, div, span), «Форматирование текста (S)» (i, b, u, em, strong), «Изображения (I)» (img). Остальные теги попали в класс «Прочее (O)». Для каждой категории был введен параметр (соответственно, T_H,T_B,T_S,T_I и T_O), значение которого равно количеству элементов соответствующего класса, включенных в новость.

В отличие от дисперсии, среднее значение размера параграфа не включено в число параметров, поскольку оно может быть выражено через параметры *S* и *N* - суммарный размер и количество параграфов соответственно.

Таким образом, каждый документ характеризуется рядом параметров (в нашем случае – девятью), поэтому, с точки зрения детектора, документ представлен девятимерным случайным вектором, элементами которого являются значения перечисленных характеристик:

$$X=(P,S,N,V,T_H,T_B,T_S,T_I,T_O) \quad (1)$$

4.2 Модель набора документов

Для описания модели набора из нескольких документов заметим следующее. Группы характеристик (*P,S,N,V)* и *(T_H,T_B,T_S,T_I,T_O)* имеют разную природу. Характеристики первой группы описывают свойства текста документа, тогда как характеристики второй группы отражают свойства его разметки. Для описания свойств набора из нескольких документов, мы будем рассматривать эти группы характеристик отдельно.

Случайные величины группы (*P,S,N,V)* имеют разнородные области значений. Так величина N обычно принимает значения в диапазоне от 1 до 100, величина V непрерывна, а значения дискретной величины P могут достигать 10^5. В связи с этим, для последующего анализа удобно все величины привести к дискретному виду, а области их значений отобразить на множество фиксированной мощности. Для этого необходимо разбить область значений каждой величины группы (*P,S,N,V)* на фиксированное количество интервалов равной длины. Пусть *m* - количество таких интервалов. Это число выбирается в зависимости от объема выборки. Одним из наиболее распространенных способов определения оптимального числа интервалов является формула Стерджесса:

$$m = 1 + \log_2 n \quad (2)$$

где *n* – количество документов в наборе[14].

Для снижения вычислительной сложности алгоритмов, использующих предлагаемую нами модель, в контексте набора из нескольких документов мы будем рассматривать величины (*P,S,N,V)* независимо друг от друга. Поэтому, с точки зрения величин (*P,S,N,V),* модель для набора документов будет представлять собой следующие четыре статистических ряда:

$$P^n = (P_1, \dots, P_m), \quad S^n = (S_1, \dots, S_m),$$
$$N^n = (N_1, \dots, N_m),\ V^n = (V_1, \dots, V_m) \quad (3)$$

Где P_i, S_i, N_i, V_i–частота попадания в i-ый интервал значения величины *P, S, N* и *V* соответственно на выборке из n документов.

Для учета в модели (3) величин (T_H,T_B,T_S,T_I,T_O) рассмотрим другой подход к представлению информации о html-элементах. В i-ом документе выборки встречается определенное количество тэгов каждой из выделенных нами пяти категорий H, B, S, I и O. Обозначим эти количества $T_H^i, T_B^i, T_S^i, T_I^i, T_O^i$ соответственно. Просуммируем их по всем документам выборки и получим следующие значения $T_H = \sum_{i=1}^n T_H^i$, $T_B = \sum_{i=1}^n T_B^i$, $T_S = \sum_{i=1}^n T_S^i$, $T_I = \sum_{i=1}^n T_I^i$, $T_O = \sum_{i=1}^n T_O^i$, которые образуют пятиэлементный статистический ряд $T^n = (T_H, T_B, T_S, T_I, T_O)$, который мы будем рассматривать в качестве модели набора документов, с точки зрения, частоты встречаемости в нем тэгов из пяти выделенных категорий.

Таким образом, модель набора документов представляет собой совокупность из следующих пяти статистических рядов:

$$P^n = (P_1, \dots, P_m), \qquad S^n = (S_1, \dots, S_m),$$
$$N^n = (N_1, \dots, N_m), \qquad V^n = (V_1, \dots, V_m),$$
$$T^n = (T_H, T_B, T_S, T_I, T_O) \tag{4}$$

5 Оперативный детектор

5.1 Принцип работы оперативного детектора

Быстродействующий компонент детектирующей системы представляет собой бинарный классификатор, который на основании значений параметров документа делает вывод о его корректности или некорректности.

Такие популярные подходы к решению задачи бинарной классификации как нейронные сети [22], опорные векторы [3], логистическая регрессия [10] могут быть использованы лишь при наличии обучающей выборки с большим количеством как позитивных, так и негативных примеров. Но при обучении оперативного детектора в большинстве случаев получить такую выборку невозможно: количество «хороших» документов при сборе новостей намного больше числа «плохих». В некоторых случаях в обучающей выборке может вообще не содержаться некорректных документов. Поэтому было решено проводить обучение классификатора на позитивных примерах, но при этом его работа была организована следующим образом: в режиме проверки документов детектор должен считать корректными лишь статьи, похожие на элементы обучающей выборки. Определим эту схожесть в терминах выбранной модели документа.

Каждый документ представлен девятимерным вектором. Рассмотрим двумерную проекцию множества таких векторов, соответствующих набору новостей с сайта kp.ru, на плоскость, задаваемую параметрами *N* (количество параграфов в статье) и *P* (объем веб-страницы, содержащей статью) (рис. 3).

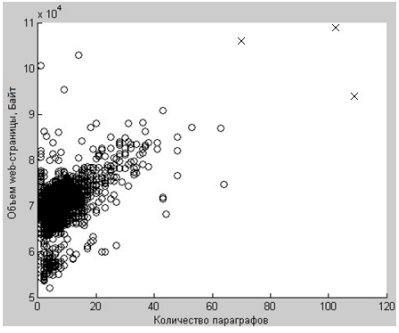

Рис. 3. Распределение значений параметров N и P.

Точки не распределены в пространстве равномерно, они сгруппированы в некоторых областях. Новый документ, поступающий на проверку, можно считать корректным, если соответствующая ему точка попадает в одну из таких областей. Если же точка находится в отдалении от этих зон (как, например, три точки в правой верхней части рисунка, помеченные крестиком), то соответствующая статья является подозрительной.

Таким образом, обучение оперативного детектора сводится к выделению таких областей, а классификация статей на корректные и некорректные – к определению, попадает ли документ в одну из выделенных областей.

Рисунок 3 демонстрирует применение предложенного подхода для определения корректности объектов с двумерными векторами характеристик, но аналогичным образом может осуществляться классификация и в случае большей размерности векторов. Однако с ростом размерности для формирования плотных областей требуется существенно увеличивать обучающую выборку. Учитывая предполагаемые объемы наборов документов (десятки тысяч новостей), при использовании девяти характеристик добиться высокой плотности при сохранении небольшого количества выделяемых зон невозможно.

Таким образом, описанная в (1) модель документа в виде 9-мерного вектора оказывается неудобной для непосредственного использования оперативным детектором, поэтому в неё были внесены изменения. Заменим девятимерный вектор X на набор векторов меньшей размерности (Y_1, Y_2, … , Y_k), каждый из которых содержит некоторое подмножество элементов X. Будем выбирать этот набор векторов исходя из следующих соображений:

- нужно по возможности использовать векторы наименьшей размерности (двумерные) для получения максимальной плотности кластеров
- нужно избегать использования векторов, которые могут оказаться бесполезными для некоторых источников

Второй пункт относится, прежде всего, к характеристикам, отражающим количество html-элементов, включенных в новость. Сайты обычно применяют для оформления новостей лишь небольшой набор тэгов, при этом некоторые группы html-элементов могут не использоваться вовсе. Поэтому только некоторые из параметров (T_H, T_B, T_S, T_I, T_O) будут принимать ненулевые значения. Каждый сайт использует собственный подход к оформлению новостей и выбору набора тэгов, что не позволяет определить универсальный критерий полезности каждой из этих характеристик и их совокупностей. Поэтому было решено все перечисленные величины включить в пятимерный вектор Y_1.

Каждый из оставшихся четырёх параметров является важной характеристикой структуры документа, поэтому в качестве элементов остальных векторов использовались все попарные сочетания величин P,S,N и V. Так были получены 6 двумерных векторов Y_2, … , Y_7.

Таким образом, модель документа, подвергающаяся обработке «оперативным детектором», представляет собой совокупность из следующих семи случайных векторов:

$$Y_1 = (T_H, T_B, T_S, T_I, T_O), \qquad Y_2 = (P, S),$$
$$Y_3 = (P, N), \qquad Y_4 = (P, V), \qquad Y_5 = (S, N),$$
$$Y_6 = (S, V), \quad Y_7 = (N, V) \tag{5}$$

5.2 Кластеризация документов

Выделение областей необходимо производить таким образом, чтобы максимально облегчить последующую проверку принадлежности точек этим областям. Поэтому нет смысла выбирать зоны сложной формы – более эффективным решением является нахождение плотных групп точек и построение простых ограничивающих поверхностей для этих групп. Для разбиения всего множества документов из обучающей выборки на группы нужно решить задачу кластеризации. Существует множество подходов к кластерному анализу, и применение различных алгоритмов к одним и тем же входным данным может дать совершенно разные результаты [21,18]. Основным требованием, определяющим пригодность метода для кластеризации новостей, является простая, гиперсферическая форма кластеров, позволяющая получить с помощью простых ограничивающих поверхностей плотные области без разреженных участков.

Одним из наиболее популярных методов кластеризации является k-means – итеративный метод кластерного анализа, основная идея которого заключается в минимизации суммарного квадратичного отклонения точек кластеров от центроидов этих кластеров [4,15]. Несмотря на низкую вычислительную сложность метод имеет существенный недостаток, связанный со спецификой обрабатываемых детектором данных. Некоторые из отдаленных точек – «выбросов» являются, всё же, корректными статьями, которые должны быть учтены при кластеризации. Для достижения минимальной разреженности такие точки должны быть по возможности помещены в отдельные кластеры. Однако этого сложно добиться с k-means, поскольку этот метод имеет тенденцию к выделению кластеров схожего размера.

Также широко распространены алгоритмы, использующие иерархический подход к кластеризации [19]. Они делятся на агломеративные (объединяющие объекты в множества) и дивизимные (разделяющие единое множество объектов на подмножества). При этом есть возможность выбора любого количества кластеров после осуществления кластеризации. Иерархические методы различаются по принципу определения двух ближайших кластеров [17]. Существует несколько подходов:

- Метод одиночной связи хорошо справляется с проблемой выбросов, но имеет тенденцию к образованию кластеров в виде длинных цепочек элементов. Такой подход эффективен в случае, когда кластеры имеют вытянутую или необычную форму, но для решения рассматриваемой задачи он неприменим.
- Метод полной связи склонен к выделению кластеров приблизительно равных размеров, что может приводить к появлению небольших разреженных кластеров вместо плотных, но крупных.
- Метод средней связи имеет склонность к образованию гиперсферических кластеров, кроме того, он даёт хороший результат при значительном варьировании размеров кластеров. Этот метод отвечает требованиям к виду формируемых кластеров, однако он имеет серьезный недостаток, характерный для всех иерархических методов – высокая вычислительная сложность ($O(n^2)$). Тем не менее, в данной работе за основу был взят этот метод, в который были внесены следующие модификации.

Ограничим число элементов, подвергающихся кластеризации методом средней связи, числом n.Тогда кластеризация N элементов ($N>n$) будет осуществляться следующим образом.

1. Выбрать из множества документов n элементов.
2. Произвести кластеризацию этих элементов методом средней связи.
3. Найти центроиды кластеров.
4. Поместить центроиды в множество точек в качестве новых элементов.
5. Повторять пункты 1-4 пока в множестве не останется необходимое число элементов.
6. Определить принадлежность исходных элементов найденным кластерам.

Для простоты будем считать, что при кластеризации всегда выделяется одинаковое число кластеров k. Найдём значение n, обеспечивающее минимальную вычислительную сложность алгоритма. При каждой итерации из множества удаляется n элементов и вместо них туда помещается k новых, то есть число элементов уменьшается на (n-k). Задачей алгоритма является замена N исходных элементов на k центроидов кластеров, то есть уменьшение числа элементов на (N-k). Поэтому выполнение кластеризации n объектов нужно произвести $\frac{(N-k)}{(n-k)}$ раз, и сложность алгоритма равна $O(n^2\frac{(N-k)}{(n-k)})$. Функция $f(n)=n^2\frac{(N-k)}{(n-k)}$ имеет минимум при n=2k. Таким образом, на каждой итерации кластеризации выполняется замена старых 2k элементов на новые k элементов. Результат кластеризации, произведенной описанным способом при k=10, приведён на рис. 4.

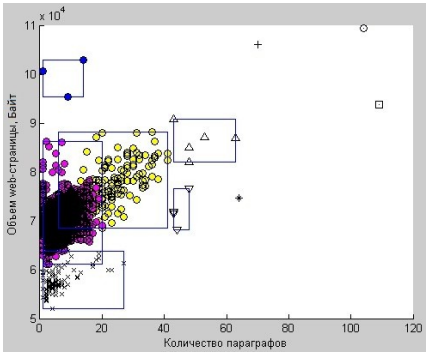

Рис. 4. Ограничивающие поверхности кластеров

В качестве ограничивающих поверхностей для областей рассматривались гиперпараллелепипед, гиперсфера и гиперэллипсоид. Выбор был сделан в пользу наиболее простых в построении гиперпараллелепипедов, показавших хорошие результаты при оценке плотности точек. Таким образом, каждый кластер задается набором пар (z_{min},z_{max}), определяющих граничные значения соответствующего гиперпараллелепипеда по параметру Z. Элемент

принадлежит кластеру, если для каждого параметра Z выполняется $z_{min} \le z \le z_{max}$, где z – значение параметра Z для рассматриваемого элемента. При классификации документ считается подозрительным, если он не попадает ни в один из кластеров.

Кластеризация и построение ограничивающих поверхностей и последующая классификация загружаемых документов производятся отдельно для каждого из семи выделенных векторов (5). Таким образом, результатом классификации является набор из семи двоичных значений. Возможны различные подходы к принятию решения о корректности документа на основании этого набора. Например, статья может считаться корректной, если она успешно прошла проверку не менее чем по *k* критериям из семи, где *k*–некоторое заданное значение. В разработанной системе используется наиболее строгий подход с параметром *k*=7.

6 Отложенный детектор

Второй компонент системы обнаружения сбоев осуществляет оценку набора документов. Оценка осуществляется на основе статистических рядов (4), которые можно рассматривать как приближения к функциям вероятности соответствующих случайных величин. Идея, лежащая в основе функционирования отложенного детектора, заключается в следующем: рассматриваемые нами случайные величины, составляющие вектор (1), подчиняются некоторым законам распределения, которые при отсутствии сбоя остаются неизменными. Изменение же вёрстки с высокой вероятностью повлияет на эти законы распределения. Следовательно, две разных выборки, состоящие из корректных документов, будут обладать высокой степенью сходства. Если же одна из них будет содержать «плохие» статьи, то различие между выборками будет значительно сильнее. Таким образом, задача детектора заключается в определении степени сходства проверяемой выборки и выборки, состоящей из гарантированно корректных статей, сформированной в процессе обучения (назовём её эталонной). На основе полученного результата принимается решение о наличии/отсутствии сбоя.

Для примера рассмотрим три выборки случайной величины *S* (суммарный размер параграфов статьи), соответствующие наборам новостей с сайта lenta.ru: эталонную (а); тестовую выборку, состоящую из «хороших» документов (б) и тестовую выборку, содержащую некорректные статьи (в). В качестве последних использовались новости с сайта cnews.ru.

На рис. 5 показаны гистограммы, соответствующие этим выборкам. Первые две из них обладают высокой степенью сходства, в то время как третья значительно от них отличается.

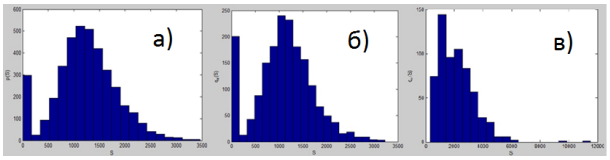

Рис. 5. Гистограммы выборок.

Для оценивания сходства выборок используется относительная энтропия (расстояние Кульбака–Лейблера, KLIC[6]). Для дискретных случайных величин с функциями вероятности p и q, принимающих значения в одном множестве $\mathcal{M} \subset \mathbb{R}$, это расстояние задается формулой

$$D_{KL}(p,q) = \sum_{x \subset \mathcal{M}} p(x) \ln \frac{p(x)}{q(x)} \qquad (6)$$

Вместо функций вероятности используются частоты рядов (4). При этом *p(x)* соответствует эталонной выборке, а *q(x)* – проверяемой.

Результатом расчёта KLIC для рядов (4) являются значения D_P, D_S, D_N, D_V, и D_T соответственно.

После расчёта расстояния Кульбака – Лейблера встаёт вопрос: как по найденному значению определить, произошел сбой или нет? Необходимо задать некоторое пороговое значение K, такое, что наличие сбоя можно определить как

$$f(D_{KL}) = \begin{cases} 0, & D_{KL} \le K - \text{сбоя нет} \\ 1, & D_{KL} > K - \text{произошел сбой} \end{cases} \qquad (7)$$

Данный порог не является фиксированной величиной, его значение зависит от числа документов в тестовой выборке. Поясним это утверждение на примере. Выберем множество $\mathbf{A} = \{A_i\}$ наборов документов A_i различной мощности и вычислим для каждого из них расстояние Кульбака – Лейблера D_iот эталонного закона распределения. Сопоставим натуральным числам *j*, соответствующим мощностям наборов из множества $\mathbf{A} = \{A_i\}$, числа K_j, определяемые как

$$K_j = \max_{A_i \in \mathbf{A}} \{D_i : |A_i| = j\} \qquad (8)$$

Рассмотрим зависимость максимального расстояния Кульбака – Лейблера от мощности набора. На рис. 6 приведена зависимость для новостей с kp.ru.

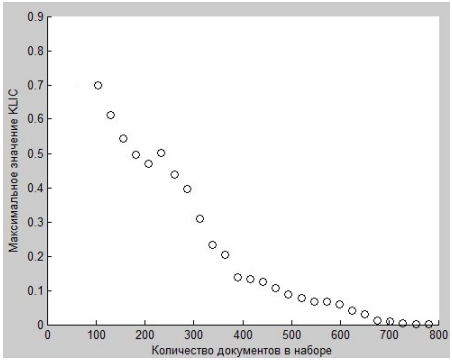

Рис. 6. Зависимость максимального значения KLIC от мощности набора

При этом использовалась оценка характеристики *P*, отражающей объем веб-страницы, но аналогичная зависимость имеет место и для других характеристик. При построении зависимости значения *j* выби-

рались равномерно в пределах от 0 до размера обучающей выборки. Наиболее точные результаты могут быть получены, если в качестве наборов с мощностью j рассматривать все возможные j-элементные подмножества документов обучающей выборки, однако эта задача имеет неполиномиальную сложность, поэтому полный перебор возможных комбинаций элементов был заменен анализом наборов, состоящих из j последовательных элементов выборки. Количество таких наборов равно r-j+1, где r – размер выборки.

Такой вид зависимости легко объясним: чем больше выборка, тем меньше на неё влияют локальные колебания значений параметров. Таким образом, при выборе порогового значения необходимо учитывать мощность анализируемого набора. Для этого необходимо определить пороговую функцию $K=h(x)$, устанавливающую соответствие между количеством документов в наборе и пороговым значением для этого набора. Для приведенных выше чисел j значение $h(j)$ должно быть максимально близко к K_j. Действительно: если $h(j)<K_j$, то появляется риск ложного срабатывания. Если же $h(j)$ значительно превышает K_j, то увеличивается вероятность ошибки пропуска сбоя (характеризующейся тем, что детектор не распознает ситуации возникновения сбоя в верстке опрашиваемого сайта). Таким образом, необходимо построить аппроксимирующую функцию по набору точек. При этом функция должна быть пригодной для экстраполяции, поскольку диапазон значений её аргумента ограничен размерами обучающей выборки, проверяемый же набор документов может иметь любую мощность.

Анализ рис. 6 ведёт к предположению об обратно пропорциональной зависимости значения K_j от j и целесообразности использования аппроксимирующей функции вида $h(x)=\frac{a}{x^b}$. Однако проведение подобного исследования для других источников и параметров показывает, что такая функция не всегда даёт приемлемый результат: в некоторых случаях зависимость имеет более сложный характер. Чтобы сделать метод определения пороговой функции пригодным для различных случаев и при этом учесть общую закономерность (постепенное уменьшение значения функции при возрастании аргумента), было решено использовать для аппроксимации функцию$h(x)=\sum_{i=0}^{k}\frac{a_i}{x^i}$, где коэффициенты a_i определяются в процессе обучения. Выбор числа k производится эмпирическим путём. Необходимо обеспечить возможность качественной аппроксимации сложной зависимости (что невозможно при малых k), но при этом по возможности избежать переобучения (возникающего при больших k). На основе исследования зависимостей, характерных для различных источников, было выбрано значение k=7. Таким образом, пороговая функция имеет вид

$$h(x)=\sum_{i=0}^{7}\frac{a_i}{x^i} \qquad (9)$$

Для определения параметров a_i пороговой функции использовался метод наименьших квадратов [16]. Оптимальная, с точки зрения МНК, функция имеет недостаток: МНК одинаково учитывает отклонение вычисленных данных от экспериментальных для всех узлов аппроксимации. Однако значения функции в точках с наименьшими и наибольшими значениями аргументов могут отличаться в тысячи раз, и отклонение, несущественное для одних узлов, будет недопустимым для других. Это может привести к значительному снижению точности на больших наборах документов и увеличению частоты ложных срабатываний. Для минимизации числа ложных срабатываний было решено подвергнуть функцию преобразованию, которое бы обеспечило выполнение условия $h_j \geq K_j$ для всех узлов. Для этого определим величину Δ:

$$\Delta = \max_j\{K_j - h_j\} \qquad (10)$$

Увеличим коэффициент a_0 на значение Δ. Теперь график пороговой функции лежит не ниже всех точек, использованных для аппроксимации. На рисунке 7 приведены графики пороговой функции до (пунктирная линия) и после (сплошная линия) коррекции.

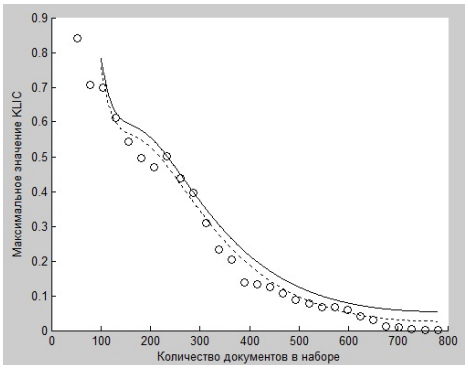

Рис. 7. График пороговой функции

С помощью приведённой пороговой функции на основании показателей D_P, D_S, D_N ,D_V и D_T получим набор из пяти двоичных значений: (F_P, F_S, F_N,F_V,F_T). В зависимости от количества единиц в этом наборе и от того, какие именно критерии приняли единичное значение, делается заключение о вероятности сбоя. В разработанной системе используется следующий подход:

- количество единиц в наборе равно 0 или 1: низкая вероятность (сбоя нет)
- 2 или 3: средняя вероятность (нельзя с уверенностью судить о наличии или отсутствии сбоя)
- 4 или 5: высокая вероятность (произошел сбой)

7 Взаимодействие детекторов

Отдельной задачей является организация взаимодействия двух детекторов с целью достижения максимально эффективного функционирования системы отслеживания сбоев. Поскольку отложенный детектор осуществляет более качественный анализ и менее склонен к ложным срабатываниям, он используется для контроля работы оперативного классификатора. Этот контроль подразумевает две основные функции:

1) Проверка правильности результатов, полученных классификатором оперативного детектора.
2) Обучение классификатора. Если оперативный детектор обнаружил подозрительный документ, а отложенный детектор в результате проверки установил отсутствие сбоя, значит, произошло ложное срабатывание. Это свидетельствует о недостаточной обученности оперативного детектора. Поэтому необходимо произвести его переобучение с использованием документов, определенных им в категорию подозрительных.

Проверка результатов оперативного детектора с помощью отложенного позволяет избежать большинства ложных срабатываний и свести к минимуму число ошибочных оповещений администратора системы о произошедших сбоях. Но в некоторых случаях ложные срабатывания могут быть обнаружены на более ранней стадии работы системы и устранены без участия отложенного детектора. Для этого оперативный классификатор был оснащен функцией самопроверки. Он способен самостоятельно отличить единичный выброс от массового поступления некорректных статей путём анализа частоты появления таких статей среди последних скачанных документов. Если эта частота меньше заданного порогового значения (например, 50%), делается вывод о ложном срабатывании и запускается переобучение. В качестве анализируемого набора при самопроверке используется группа документов, полученных в рамках последней транзакции, т.е. при последней загрузке новостей с сайта.

Рассмотрим итоговый метод обнаружения изменений структуры веб-сайтов, реализованный в работе подсистемы обнаружения сбоев с учетом выбранного подхода к реализации взаимодействия детекторов. Этапы функционирования подсистемы приведены на рис. 8.

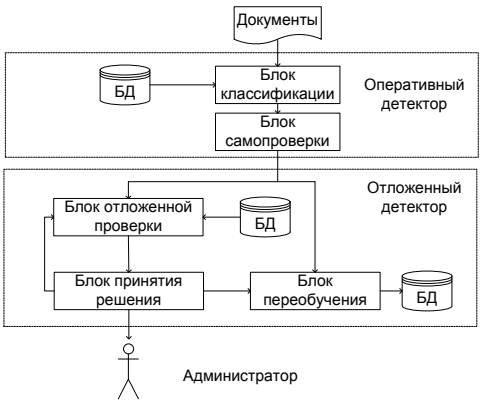

Рис. 8. Этапы обнаружения сбоев

На этапе классификации оперативный детектор проверяет поступающие статьи. Документы классифицируются на корректные и подозрительные. Необходимые для классификации данные о кластерах и ограничивающих поверхностях извлекаются из базы данных.

После поступления от источника группы новостей оперативный детектор выполняет самопроверку: вычисляется частота детектирования подозрительных статей в пределах текущей транзакции. Если она ниже порогового значения, но не равна нулю, делается заключение о ложном срабатывании и выполняется переход к блоку переобучения. Если частота выше порогового значения – к блоку отложенной проверки.

Работа блока отложенной проверки начинается с оповещения отложенного детектора о необходимости выполнения анализа. Выполнение проверки непосредственно после получения оповещения не имеет смысла, поскольку сбой может быть зафиксирован только после накапливания достаточного числа некорректных статей. После поступления необходимого числа документов отложенный детектор выполняет проверку этого набора. Для её проведения из базы данных извлекаются статистические ряды эталонных выборок и коэффициенты a_i пороговой функции. Результат проверки передается блоку принятия решения.

Блок принятия решения определяет дальнейшие действия подсистемы в зависимости от результата отложенной проверки. Если она показала высокую вероятность сбоя, администратор системы оповещается о необходимости корректировки системы сбора документов. Если вероятность сбоя низка, делается заключение о ложном срабатывании оперативного классификатора и выполняется переход к блоку переобучения. Если же результат анализа не позволяет с высокой долей уверенности судить о наличии или отсутствии сбоя, выполняется повторная отложенная проверка.

На этапе переобучения для оперативного детектора заново определяются кластеры и строятся ограничивающие поверхности с использованием нового, дополненного набора данных. Количество кластеров и граничные значения гиперпараллелепипедов заносятся в базу данных.

8 Экспериментальная проверка системы

В рамках данной работы были проведены эксперименты, направленные на анализ качества работы разработанной системы обнаружения сбоев. Эксперименты проводились на ПЭВМ со следующими основными параметрами: процессор Intel Core 2 Duo 1,8 ГГц, объем ОЗУ 2 Гб.

Для проведения экспериментов использовалась коллекция новостей, извлеченных со следующих сайтов: mail.ru, itar-tass.com, kp.ru, rbc.ru, kommersant.ru, ria.ru, rambler.ru. Для обучения использовалось в общей сложности 72888 корректных документов. При обучении оперативного детектора формировалось 10 кластеров.

Тестирование оперативного детектора производилось в течение трёх суток. При самопроверке было использовано пороговое значение, равное 10%. Накопленные за время тестирования документы использовались в качестве тестовой выборки для отложенного детектора.

Целью первого эксперимента была оценка работы системы на корректных данных. В качестве

входных данных использовались гарантированно корректные статьи, полученные с использованием правильных настроек системы сбора. Для проведения эксперимента использовалось в общей сложности 5169 документов.

Табл. 1. Ложные срабатывания оперативного детектора.

Источник	M_L	M_T	M_S	N_D	N_S
mail.ru	25296	2631	20	14	0
itar-tass.com	11548	560	76	0	0
kp.ru	7220	218	24	4	1
rbc.ru	3517	227	25	14	5
kommersant.ru	5288	260	47	4	0
ria.ru	16519	1115	29	12	5
rambler.ru	3500	158	15	17	13
Всего:	72888	5169	34	65	24

Где M_L - размер обучающей выборки, M_T - размер тестовой выборки, M_S - средний размер анализируемого набора документов при самопроверке, N_D - количество подозрительных статей, N_S - количество подозрительных статей после самопроверки.

В рамках эксперимента проверке были подвергнуты 5169 корректных статей. При первичной классификации 65 из них (1,26%) были определены как подозрительные. В результате самопроверки 41 из них была переведена в категорию корректных. Оставшиеся 24 (0,46% от общего числа) были ошибочно признаны некорректными.

Табл. 2. Ложные срабатывания отложенного детектора.

Источник	M_L	M_T	F_P	F_S	F_N	F_V	F_T	N_F	P_F
mail.ru	25296	2631	0	0	0	0	0	0 из 5	L
itar-tass.com	11548	560	0	0	0	0	0	0 из 5	L
kp.ru	7220	218	1	0	0	0	0	1 из 5	L
rbc.ru	3517	227	0	0	0	0	0	0 из 5	L
kommersant.ru	5288	260	0	0	0	0	0	0 из 5	L
ria.ru	16519	1115	0	0	0	0	0	0 из 5	L
rambler.ru	3500	158	0	0	0	0	0	0 из 5	L
Всего:	72888	5169	1	0	0	0	0	1 из 35	

N_F - количество критериев, показавших наличие сбоя, P_F - заключение детектора: вероятность сбоя (L-низкая, M – средняя, H - высокая)

Отложенный детектор показал правильный результат при проверке тестовой выборки каждого сайта. Ошибочное значение критерия было зафиксировано лишь в 1 случае из 35 (2,86%).

В рамках второго эксперимента оценивалась способность системы обнаруживать сбои. Ввиду отсутствия для многих сайтов достаточного числа негативных примеров, тестовые наборы были созданы искусственно: в качестве «плохих» документов использовались комментарии к новостям, полученные с сайта championat.com. Такой выбор тестовых данных обусловлен тем, что возможным последствием изменения верстки является извлечение из вебстраниц не новостей, а текстов с других участков сайта, в частности, комментариев. Для проведения эксперимента использовалось 356 документов (для всех источников использовался одинаковый тестовый набор).

В рамках эксперимента проверке были подвергнуты 356 некорректных статей. При первичной классификации все они были определены как подозрительные для каждого из семи источников. В результате самопроверки никаких изменений произведено не было.

Отложенный детектор показал правильный результат для 4 источников из 7. Для оставшихся 3 источников он не смог сделать вывод о наличии или отсутствии сбоя. В 8 случаях из 35 (22,85%) значение критериев было неверным. Данным ситуациям соответствуют значения 0 соответствующего критерия в таблице 4.

Если в ходе первого эксперимента система обнаружения сбоев продемонстрировала свою работоспособность при выполнении как оперативной, так и отложенной проверки корректных данных, то с задачей обнаружения сбоев она справилась значительно хуже. Возможной причиной низкого качества работы системы при анализе некорректных документов является неудачный подход к определению результата проверки. Анализ результатов экспериментов показывает необходимость понижения порога фиксации сбоя. Кроме того, при проведении второго эксперимента критерии F_P, F_S, F_N, F_V и F_T были приняты равнозначными, однако оказалось, что некоторые из них показывают наличие сбоя значительно точнее, чем другие. Так, критерии F_P и F_N приняли верное значение в 7 случаях из 7, а F_V – лишь в 3. Чтобы учесть различную значимость критериев, для каждого из них может быть установлен весовой коэффициент, определяющий влияние значения соответствующего критерия на результат проверки.

Табл. 3. Оценка пропуска сбоев оперативным детектором.

Источник	M_L	M_T	M_S	N_D	N_S
mail.ru	25296	356	25	356	356
itar-tass.com	3500	356	25	356	356
kp.ru	11548	356	25	356	356
rbc.ru	7220	356	25	356	356
kommersant.ru	16519	356	25	356	356
ria.ru	3517	356	25	356	356
rambler.ru	5288	356	25	356	356
Всего:	72888	2492	25	2492	2492

Табл. 4. Оценка пропуска сбоев отложенным детектором.

Источник	M_L	M_T	F_P	F_S	F_N	F_V	F_T	N_F	P_F
mail	25296	356	1	1	1	0	0	3 из 5	M
itar-tass	11548	356	1	1	1	0	0	3 из 5	M
kp	7220	356	1	0	1	0	1	3 из 5	M
rbc	3517	356	1	1	1	0	1	4 из 5	H
kommersant	5288	356	1	1	1	1	1	5 из 5	H
ria	16519	356	1	0	1	1	1	4 из 5	H
rambler	3500	356	1	1	1	1	1	5 из 5	H
Всего:	72888	2492	7	5	7	3	5	27 из 35	

Заключение

В работе предложен метод обнаружения изменений структуры новостных веб-сайтов. В основе метода лежит двухуровневая проверка корректности новостей, обеспечивающая быстроту реакции и высокое качество оценки документов.

В основе первичной классификации новостей лежит проверка схожести документа с элементами обучающей выборки. Это позволяет системе адекватно реагировать на любые нетипичные для сайта новости. Простота выполнения этой проверки достигается с помощью предложенного метода кластеризации. Он относится к иерархическим методам, но имеет меньшую вычислительную сложность по сравнению с другими алгоритмами этого класса.

Отложенная проверка корректности основана на сравнении законов распределения. Для правильной интерпретации полученного результата используется пороговая функция, полученная путём аппроксимации методом МНК. Такой подход обеспечивает высокую точность проверки в независимости от размера оцениваемой выборки.

Проведенные эксперименты показали эффективность совместного использования двух детекторов. Предложенный подход был реализован в виде подсистемы отслеживания сбоев в системе сбора новостной информации. Данная система успешно внедрена в Совете Федерации Федерального Собрания РФ в рамках комплекса «Обзор СМИ», решающего задачу сбора, накопления и классификации новостей общественно-политической тематики.

Литература

[1] Tobias Anton. XPath-Wrapper Induction by generalizing tree traversal patterns // Workshopwoche der GI-Fachgruppen/Arbeitskreise GI-Fachgruppen ABIS, AKKD, FGML – 2005, p. 126–133.

[2] Boris Chidlovskii, Jon Ragetli and Maarten de Rijke. Wrapper generation by reversible grammar induction // Machine Learning: ECML 2000, Lecture Notes in Computer Science, 2000, Vol. 1810 – P. 96-108.

[3] NelloCristianini and John Shawe-Taylor. An Introduction to Support Vector Machines and other kernel-based learning methods. Cambridge University Press, 2000

[4] Jain A. Dubs R., Clustering methods and algorithms,1988 // Prentice-Hall Inc.

[5] Hans-Peter Kriegel, Peer Kröger, Arthur Zimek. Outlier Detection Techniques. The Thirteenth Pacific-Asia Conference on Knowledge Discovery and Data Mining, 2009

[6] Kullback S., Leibler R.A. On information and sufficiency // The Annals of Mathematical Statistics. 1951. V.22. №1. P. 79-86.

[7] Nicholas Kushmerick, Dan S. Weld, and Robert B. Doorenbos. Wrapper induction for information extraction. In Proceedings of the Intl. Joint Conference on Artificial Intelligence (IJCAI), pages 729-737, 1997.

[8] Nicholas Kushmerick. Wrapper induction: Efficiency and expressiveness // Artificial Intelligence – 2000. – №118. – P. 15–68.

[9] Nicholas Kushmerick. Wrapper verification. World Wide Web Journal, 3(2):79-94, 2000.

[10] Lemeshow, David W. Hosmer, Stanley (2000). Applied logistic regression (2nd ed.). New York: Wiley

[11] K. Lerman, Steven Minton, and Craig Knoblock. Wrapper maintenance: A machine learning approach. Journal of Artificial Intelligence Research, 18:149-181, 2003.

[12] Daniel Nikovski, Alan Esenther. Semi-Supervised Information Extraction FromVariable-LengthWeb-PageLists // International Conf. onEnterprise Information Systems – 2009.

[13] Ermelinda Oro, Massimo Ruffolo, Steffen Staab. SXPath - Extending XPath towards Spatial Querying on Web Documents // Proceedings of the VLDB Endowment – 2011.–Vol. 4, No. 2 – P. 129–140.

[14] Sturges H. The choice of a class-interval., 1926, Journal of the American Statistical Association, Vol. 21, No. 153, P. 65-66.

[15] А.М. Андреев, Д.В. Березкин, В.В. Морозов, К.В. Симаков. Метод кластеризации документов текстовых коллекций и синтеза аннотаций кластеров // Электронные библиотеки: перспективные методы и технологии, электронные коллекции: Труды 10-ой Всероссийской научной конференции (RCDL'2008). – Дубна, 2008. – С. 220–229.

[16] Аппроксимация методом наименьших квадратов (МНК) [Электронный ресурс] – Режим доступа: http://alglib.sources.ru/interpolation/linearleastsquares.php, свободный

[17] Бериков В.Б., Лбов Г.С. Современные тенденции в кластерном анализе. – Новосибирск: Институт математики им. С.Л. Соболева

[18] Дюран Б., Оделл П. Кластерный анализ. М.: Статистика, 1977. 128 с.

[19] Жамбю М. Иерархический кластер-анализ и соответствия. М.: Финансы и статистика, 1988. 342 с.

[20] Кендалл М., Стьюарт А. Статистические выводы и связи. — М.: Наука, 1973

[21] Мандель И.Д. Кластерный анализ. М.: Финансы и статистика, 1988. 176 с.

[22] Хайкин С. Нейронные сети: полный курс 2-еиздание.: Пер. сангл. М.: Вильямс, 2006. 1104 с.

The method of detecting structure changes of news websites

Arkady Andreev, Dmitry Berezkin, Ilya Kozlov, Konstantin Simakov

This article describes unsupervised method of detecting structure's and markup's changes of targeted websites. This problem arises when we deal with maintenance of real-life HTML-wrapper applications.

We have proposed two stages of detection – online and offline. The former is based on clustering considering HTML-document as a vector of some features. The later builds statistical distributions of such features for learning and testing sets of HTML-documents. Comparing such distributions we can make decision of structure's or markup's change.

We have carried out several experiments with data obtained from real wrapper. The result shows the robustness of our approach.