2nd Annual International Symposium on Information Management and Big Data

2nd, 3th and 4th September 2015

Cusco - Peru

PROCEEDINGS

Proceeding Editors: J. A. Lossio-Ventura and H. Alatrista-Salas

Table of contents

The SIMBig 2015 Organizing Committee confirms that full and concise papers accepted for this publication:

- Meet the definition of research in relation to creativity, originality, and increasing humanity's stock of knowledge;
- Are selected on the basis of a peer review process that is independent, qualified expert review;
- Are published and presented at a conference having national and international significance as evidenced by registrations and participation; and
- Are made available widely through the Conference web site.

Disclaimer: The SIMBig 2015 Organizing Committee accepts no responsibility for omissions and errors.

Organizing Committee of SIMBig 2015

General Organizers

- **Juan Antonio LOSSIO-VENTURA**, University of Montpellier - LIRMM, FRANCE
- **Hugo ALATRISTA-SALAS**, Pontificia Universidad Católica del Perú - GRPIAA Labs, PERU

Local Organizers

- Cristhian GANVINI VALCARCEL Universidad Andina del Cusco, PERU
- Armando FERMIN PEREZ, Universidad Nacional Mayor de San Marcos, PERU

Track Organizers: WTI *(Web and Text Intelligence Track)*

- Jorge Carlos VALVERDE-REBAZA, University of São Paulo, Brazil
- Alneu DE ANDRADE LOPES, ICMC – University of São Paulo, Brazil
- Ricardo Bastos CAVALCANTE PRUDENCIO, Federal University of Pernambuco, Brazil
- Estevam HRUSCHKA, Federal University of São Carlos, Brazil
- Ricardo CAMPOS, Polytechnic Institute of Tomar - LIAAD–INESC Technology and Science, Portugal

SIMBig 2015 Program Committee

- **Nathalie Abadie**, French National Mapping Agency, COGIT, FRANCE
- **Elie Abi-Lahoud**, University College Cork, Cork, IRELAND
- **Salah AitMokhtar**, Xerox Research Centre Europa, FRANCE
- **Sophia Ananiadou**, NaCTeM - University of Manchester, UNITED KINGDOM
- **Marcelo Arenas**, Pontificia Universidad Catolica de Chile, CHILE
- **Jérôme Azé**, LIRMM - University of Montpellier, FRANCE
- **Pablo Barceló**, Universidad de Chile, CHILE
- **Cesar A. Beltrán Castañón**, GRPIAA - Pontifical Catholic University of Peru, PERU
- **Albert Bifet**, Huawei Noah's Ark Research Lab, Hong Kong, CHINA
- **Sandra Bringay**, LIRMM - Paul Valéry University, FRANCE
- **Oscar Corcho**, Ontology Engineering Group - Polytechnic University of Madrid, SPAIN
- **Gabriela Csurka**, Xerox Research Centre Europa, FRANCE
- **Frédéric Flouvat**, PPME Lab - University of New Caledonia, NEW CALEDONIA
- **André Freitas**, Dept. of Computer Science and Mathematics, University of Passau, GERMANY
- **Adrien Guille**, ERIC Lab - University of Lyon 2, FRANCE
- **Hakim Hacid**, Zayed University, UNITED ARAB EMIRATES
- **Sébastien Harispe**, LGI2P/EMA Research Centre, Site EERIE, Parc Scientifique, FRANCE
- **Dino Ienco**, Irstea, FRANCE
- **Diana Inkpen**, University of Ottawa, CANADA
- **Clement Jonquet**, LIRMM - University of Montpellier, FRANCE
- **Alípio Jorge**, Universidade do Porto, PORTUGAL
- **Yannis Korkontzelos**, NaCTeM - University of Manchester, UNITED KINGDOM

- **Eric Kergosien**, GERiCO Lab - University of Lille 3, FRANCE
- **Peter Mika**, Yahoo! Research Labs - Barcelone, SPAIN
- **Phan Nhat Hai**, Oregon State University, UNITED STATES of AMERICA
- **Jordi Nin**, Barcelona Supercomputing Center (BSC) - BarcelonaTECH, SPAIN
- **Miguel Nuñez del Prado Cortez**, Intersec Lab - Paris, FRANCE
- **Thomas Opitz**, Biostatistics and Spatial Processes - INRA, FRANCE
- **Yoann Pitarch**, IRIT - Toulouse, FRANCE
- **Pascal Poncelet**, LIRMM - University of Montpellier, FRANCE
- **Julien Rabatel**, LIRMM, FRANCE
- **José Luis Redondo García**, EURECOM, FRANCE
- **Mathieu Roche**, Cirad - TETIS - LIRMM, FRANCE
- **Nancy Rodriguez**, LIRMM - University of Montpellier, FRANCE
- **Arnaud Sallaberry**, LIRMM - Paul Valéry University, FRANCE
- **Nazha Selmaoui-Folcher**, PPME Labs - University of New Caledonia, NEW CALEDONIA
- **Maguelonne Teisseire**, Irstea - LIRMM, FRANCE
- **Paulo Teles**, LIAAD-INESC Porto LA, Porto University, PORTUGAL
- **Julien Velcin**, ERIC Lab - University of Lyon 2, FRANCE
- **Maria-Esther Vidal**, Universidad Simón Bolívar, VENEZUELA
- **Boris Villazon-Terrazas**, Expert System Iberia - Madrid, SPAIN
- **Osmar R. Zaïane**, Department of Computing Science, University of Alberta, CANADA

WTI 2015 Program Committee *(Web and Text Intelligence Track)*

- **Adam Jatowt**, Kyoto University, Japan,
- **Claudia Orellana Rodriguez**, Insight Centre for Data Analytics - University College Dublin, Ireland,
- **Ernesto Diaz-Aviles**, IBM Research, Ireland,
- **Jannik Strötgen**, Heidelberg University, Germany,

- **João Paulo Cordeiro**, University of Beira Interior, Portugal,
- **Lucas Drumond**, University of Hildesheim, Germany,
- **Leon Derczynski**, University of Sheffield, UK,
- **Miguel Martinez-Alvarez**, Signal, UK,
- **Brett Drury**, USP, Brazil,
- **Celso Antônio Alves Kaestner**, UTFPR, Brazil,
- **Edson Matsubara**, UFMS, Brazil,
- **Flavia Barros**, UFPe, Brazil,
- **Hércules Antonio do Prado**, PUC Brasília, Brazil,
- **Huei Lee**, UNIOESTE, Brazil,
- **João Luís Rosa**, USP, Brazil,
- **José Luís Borges**, Universidade do Porto, Portugal,
- **José Paulo Leal**, Universidade do Porto, Portugal,
- **Maria Cristina Ferreira de Oliveira**, USP, Brazil,
- **Ronaldo Prati**, UFABC, Brazil,
- **Thiago A. S. Pardo**, USP, Brazil,
- **Solange Rezende**, USP, Brazil,
- **Marcos Aurélio Domingues**, USP, Brazil.

List of contributions

Overview of SIMBig 2015: 2nd Annual International Symposium on Information Management and Big Data

Juan Antonio Lossio-Ventura
LIRMM, University of Montpellier
Montpellier, France
juan.lossio@lirmm.fr

Hugo Alatrista-Salas
Pontificia Universidad Católica del Perú
Lima, Peru
halatrista@pucp.pe

Abstract

Big Data is a popular term used to describe the exponential growth and availability of both structured and unstructured data. The aim of the symposium is to present the analysis methods for managing large volumes of data through techniques of artificial intelligence and data mining. Bringing together main national and international actors in the decision-making field to state in new technologies dedicated to handle large amount of information.

1 Introduction

Big Data is a popular term used to describe the exponential growth and availability of both structured and unstructured data. This has taken place over the last 20 years. For instance, social networks such as Facebook, Twitter and Linkedin generate masses of data, which is available to be accessed by other applications. Several domains, including biomedicine, life sciences and scientific research, have been affected by Big Data[1]. Therefore there is a need to understand and exploit this data. This process can be carried out thanks to "Big Data Analytics" methodologies, which are based on Data Mining, Natural Language Processing, etc. That allows us to gain new insight through data-driven research (Madden, 2012; Embley and Liddle, 2013). A major problem hampering Big Data Analytics development is the need to process several types of data, such as structured, numeric and unstructured data (e.g. video, audio, text, image, etc)[2].

Therefore, the second edition of the Annual International Symposium on Information Management and Big Data - SIMBig 2015[3], aims to present the analysis methods for managing large volumes of data through techniques of artificial intelligence and data mining. Counting with main national and international actors in the decision-making field to state in new technologies dedicated to handle large amount of information.

Our first edition, SIMBig 2014[4] took place in Cuzco Peru too in September 2015. SIMBig 2014 has been indexed on DBLP[5] (Lossio-Ventura and Alatrista-Salas, 2014) and on CEUR Workshop Proceedings[6].

1.1 Keynote Speakers

SIMBig 2015 second edition has welcomed five keynote speakers experts in Big Data, Data Mining, Natural Language Processing (NLP), and Social Networks:

- PhD. Pr. **Albert Bifet**, from HUAWEI Noah's Ark Lab, China;
- PhD. Pr. **Diana Inkpen**, from University of Ottawa, Canada;
- PhD. Pr. **Pascal Poncelet**, from LIRMM Laboratory and University of Montpellier, France;
- PhD. Pr. **Mathieu Roche**, from Cirad and TETIS Laboratory, France;
- PhD. Pr. **Osmar R. Zaïane**, from University of Alberta, Canada.

1.2 Scope and Topics

To share the new analysis methods for managing large volumes of data, we encouraged participation from researchers in all fields related to Big Data, Data Mining, and Natural Language Processing, but also Multilingual Text Processing, Biomedical NLP. Topics of interest of SIMBig 2015 included but were not limited to:

- Big Data
- Data Mining
- Natural Language Processing

[1] By 2015 the average of data annually generated in hospitals is 665TB: http://ihealthtran.com/wordpress/2013/03/infographic-friday-the-body-as-a-source-of-big-data/.

[2] Today, 80% of data is unstructured such as images, video, and notes

[3] http://simbig.org/SIMBig2015/

[4] https://www.lirmm.fr/simbig2014/

[5] http://dblp2.uni-trier.de/db/conf/simbig/simbig2014

[6] http://ceur-ws.org/Vol-1318/index.html

- Bio NLP
- Text Mining
- Information Retrieval
- Machine Learning
- Semantic Web
- Ontologies
- Web Mining
- Knowledge Representation and Linked Open Data
- Social Networks, Social Web, and Web Science
- Information visualization
- OLAP, Data Warehousing
- Business Intelligence
- Spatiotemporal Data
- Health Care
- Agent-based Systems
- Reasoning and Logic
- Constraints, Satisfiability, and Search

2 Latin American and Peruvian Academic Goals of the Symposium

The academic goals of the symposium are varied, among which we can list the following:

- Meet Latin American and foreign researchers, teachers, and students belonging to several domains of computer sciences, specially related to Big Data.
- Promote the production of scientific articles, which will be evaluated by the international scientific community, in order to receive a feedback from experts.
- Foster partnerships between Latin American universities, local universities and European universities.
- Promote the creation of alliances between Peruvian universities, enabling decentralization of education.
- Motivate students to learn more about computer sciences research to solve problems related to the management of information and Big Data.
- Promote the research in Peruvian universities, mainly those belonging to the local organizing committee.
- Create connections, forming networks of partnerships between companies and universities.
- Promote the local and international tourism, in order to show to the participants the architecture, gastronomy and local cultural heritage.

3 Track on Web and Text Intelligence (WTI 2015)

Web and text intelligence are related areas that have been used to improve human computer interaction both in general and in particular to explore and analyze information that is available on the Internet. With the advent of social networks and the emergence of services such as Facebook, Twitter, and others, research in these areas has been greatly enhanced. In recent years, shared knowledge and experiences have established new and different types of personal and communal relationships which have been leveraged by social networks scientists to produce new insights. In addition there has been a huge increase in community activities on social networks.

The Web and Text Intelligence (WTI) track of SIMBig 2015 have provided a forum that brought together researchers and practitioners for exploring technologies, issues, experiences and applications that help us to understand the Web and to build automatic tools to better exploit this complex environment. The WTI track has fostered collaborations, exchange of ideas and experiences among people working in a variety of highly cross-disciplinary research fields such as computer science, linguistics, statistics, sociology, economics, and business.

The WTI track is a follow up of the 4th International Workshop on Web and Text Intelligence[7], which took place in Curitiba, Brazil, October 2012, as a workshop of BRACIS 2012; the 3rd International Workshop on Web and Text Intelligence[8], which took place in São Bernardo, Brazil, October 2010, as a workshop of SBIA10; the 2nd International Workshop on Web and Text Intelligence[9], which took place in São Carlos, Brazil, September 2009, as a workshop of STIL09; the 1st Web and Network Intelligence[10], which took place in Aveiro, Portugal, October 2009, as a thematic track of EPIA09; and the 1st International Workshop on Web and Text Intelligence, which took place in

[7] http://www.labic.icmc.usp.br/wti2012/
[8] http://www.labic.icmc.usp.br/wti2010/
[9] http://www.labic.icmc.usp.br/wti2009/
[10] http://epia2009.web.ua.pt/wni/

Salvador, Brazil, October 2008, as a workshop of SBIA08.

3.1 Scope and Topics

The topics of WTI include, but are not limited to:

- Web and Text Mining
- Link Mining
- Web usability
- Web automation and adaptation
- Graph and complex network mining
- Communities analysis in social networks
- Relationships analysis in social networks
- Applications of social networks and social media
- Data modeling for social networks and social media
- Location-based social networks analysis
- Big data issues in social network and media analysis
- Modeling of user behavior and interactions
- Temporal analysis of social networks and social media
- Pattern analysis in social networks and social media
- Privacy and security in social networks
- Propagation and diffusion of information in social network
- Social information applied to recommender systems
- Search and Web Mining
- Multimedia Web Mining
- Visualization of social information

4 Sponsors

We want to thank our wonderful sponsors! We extend our sincere appreciation to our sponsors, without whom our symposium would not be possible. They showed their commitment to making our research communities more active. We invite you to support these community-minded organizations.

4.1 Organizing Institutions

- Université de Montpellier, France[11]
- Laboratoire de Informatique, Robotique et Microélectronique de Montpellier, France[12]
- Universidad Andina del Cusco, Perú[13]

[11] http://www.umontpellier.fr/
[12] http://www.lirmm.fr/
[13] http://www.uandina.edu.pe/

4.2 Collaborating Institutions

- iMedia[14]
- Bioincuba[15]
- TechnoPark[16]
- Grupo de Reconocimiento de Patrones e Inteligencia Artificial Aplicada, PUCP, Perú[17]
- Universidad Nacional Mayor de San Marcos, Perú[18]
- Escuela de Post-grado de la Pontificia Universidad Católica del Perú[19]

4.3 WTI Organizing Institutions

- Instituto de Ciências Matemáticas e de Computação, USP, Brasil[20]
- Instituto Politécnico de Tomar, Portugal[21]
- Labóratorio de Intêligencia Computacional, ICMC, USP, Brasil[22]
- Labóratorio de Intêligencia Artificial e Apoio à Decisão, INESC TEC, Portugal[23]
- Machine Learning Lab (MaLL), UFSCar, Brasil[24]
- Universidade Federal de São Carlos, Brasil[25]

References

David W Embley and Stephen W Liddle. 2013. Big data—conceptual modeling to the rescue. In *Conceptual Modeling*, ER'13, pages 1–8. LNCS, Springer.

Juan Antonio Lossio-Ventura and Hugo Alatrista-Salas, editors. 2014. *Proceedings of the 1st Symposium on Information Management and Big Data - SIMBig 2014, Cusco, Peru, September 8-10, 2014*, volume 1318 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Sam Madden. 2012. From databases to big data. volume 16, pages 4–6. IEEE Educational Activities Department, Piscataway, NJ, USA, may.

[14] http://www.imedia.pe/
[15] http://www.bioincuba.com/
[16] http://technoparkidi.org/
[17] http://inform.pucp.edu.pe/~grpiaa/
[18] http://www.unmsm.edu.pe/
[19] http://posgrado.pucp.edu.pe/la-escuela/presentacion/
[20] http://www.icmc.usp.br/Portal/
[21] http://portal2.ipt.pt/
[22] http://labic.icmc.usp.br/
[23] http://www.inesctec.pt/liaad
[24] http://ppgcc.dc.ufscar.br/pesquisa/laboratorios-e-grupos-de-pesquisa
[25] http://www2.ufscar.br/home/index.php

Real-Time Big Data Stream Analytics

Albert Bifet
Université Paris-Saclay
Télécom ParisTech
Dṕartement Informatique et Réseaux
46 rue Barrault
75634 Paris Cedex 13, FRANCE
albert.bifet@telecom-paristech.fr

Abstract

Big Data is a new term used to identify datasets that we cannot manage with current methodologies or data mining software tools due to their large size and complexity. Big Data mining is the capability of extracting useful information from these large datasets or streams of data. New mining techniques are necessary due to the volume, variability, and velocity, of such data. MOA is a software framework with classification, regression, and frequent pattern methods, and the new APACHE SAMOA is a distributed streaming software for mining data streams.

1 Introduction

Big Data is a new term used to identify the datasets that due to their large size, we can not manage them with the typical data mining software tools. Instead of defining "Big Data" as datasets of a concrete large size, for example in the order of magnitude of petabytes, the definition is related to the fact that the dataset is too big to be managed without using new algorithms or technologies. There is need for new algorithms, and new tools to deal with all of this data. Doug Laney (Laney, 2001) was the first to mention the 3 V's of Big Data management:

- Volume: there is more data than ever before, its size continues increasing, but not the percent of data that our tools can process
- Variety: there are many different types of data, as text, sensor data, audio, video, graph, and more
- Velocity: data is arriving continuously as streams of data, and we are interested in obtaining useful information from it in real time

Nowadays, there are two more V's:

- Variability: there are changes in the structure of the data and how users want to interpret that data
- Value: business value that gives organizations a competitive advantage, due to the ability of making decisions based in answering questions that were previously considered beyond reach

For velocity, data stream real time analytics are needed to manage the data currently generated, at an ever increasing rate, from such applications as: sensor networks, measurements in network monitoring and traffic management, log records or click-streams in web exploring, manufacturing processes, call detail records, email, blogging, twitter posts and others. In fact, all data generated can be considered as streaming data or as a snapshot of streaming data, since it is obtained from an interval of time.

In the data stream model, data arrive at high speed, and algorithms that process them must do so under very strict constraints of space and time. Consequently, data streams pose several challenges for data mining algorithm design. First, algorithms must make use of limited resources (time and memory). Second, they must deal with data whose nature or distribution changes over time.

We need to deal with resources in an efficient and low-cost way. In data stream mining, we are interested in three main dimensions:

- accuracy
- amount of space necessary
- the time required to learn from training examples and to predict

These dimensions are typically interdependent: adjusting the time and space used by an algorithm can influence accuracy. By storing more precomputed information, such as look up tables, an algorithm can run faster at the expense of space. An algorithm can also run faster by processing less information, either by stopping early or storing less, thus having less data to process.

2 MOA

Massive **O**nline **A**nalysis (MOA) (Bifet et al., 2010) is a software environment for implementing algorithms and running experiments for online learning from evolving data streams. MOA includes a collection of offline and online methods as well as tools for evaluation. In particular, it implements boosting, bagging, and Hoeffding Trees, all with and without Naïve Bayes classifiers at the leaves. Also it implements regression, and frequent pattern methods. MOA supports bidirectional interaction with WEKA, the Waikato Environment for Knowledge Analysis, and is released under the GNU GPL license.

3 APACHE SAMOA

APACHE SAMOA (SCALABLE ADVANCED MASSIVE ONLINE ANALYSIS) is a platform for mining big data streams (Morales and Bifet, 2015). As most of the rest of the big data ecosystem, it is written in Java.

APACHE SAMOA is both a framework and a library. As a framework, it allows the algorithm developer to abstract from the underlying execution engine, and therefore reuse their code on different engines. It features a pluggable architecture that allows it to run on several distributed stream processing engines such as Storm, S4, and Samza. This capability is achieved by designing a minimal API that captures the essence of modern DSPEs. This API also allows to easily write new bindings to port APACHE SAMOA to new execution engines. APACHE SAMOA takes care of hiding the differences of the underlying DSPEs in terms of API and deployment.

As a library, APACHE SAMOA contains implementations of state-of-the-art algorithms for distributed machine learning on streams. For classification, APACHE SAMOA provides a Vertical Hoeffding Tree (VHT), a distributed streaming version of a decision tree. For clustering, it includes an algorithm based on CluStream. For regression, HAMR, a distributed implementation of Adaptive Model Rules. The library also includes meta-algorithms such as bagging and boosting.

The platform is intended to be useful for both research and real world deployments.

3.1 High Level Architecture

We identify three types of APACHE SAMOA users:

1. Platform users, who use available ML algorithms without implementing new ones.
2. ML developers, who develop new ML algorithms on top of APACHE SAMOA and want to be isolated from changes in the underlying SPEs.
3. Platform developers, who extend APACHE SAMOA to integrate more DSPEs into APACHE SAMOA.

4 Conclusions

Big Data Mining is a challenging task, that needs new tools to perform the most common machine learning algorithms such as classification, clustering, and regression.

APACHE SAMOA is is a platform for mining big data streams, and it is already available and can be found online at `http://www.samoa-project.net`. The website includes a wiki, an API reference, and a developer's manual. Several examples of how the software can be used are also available.

Acknowledgments

The presented work has been done in collaboration with Gianmarco De Francisci Morales, Bernhard Pfahringer, Geoff Holmes, Richard Kirkby, and all the contributors to MOA and APACHE SAMOA.

References

Gianmarco De Francisci Morales and Albert Bifet. 2015. SAMOA: Scalable Advanced Massive Online Analysis. *Journal of Machine Learning Research*, 16:149–153.

Albert Bifet, Geoff Holmes, Richard Kirkby, and Bernhard Pfahringer. 2010. MOA: Massive Online Analysis. *Journal of Machine Learning Research*, 11:1601–1604, August.

Doug Laney. 2001. 3-D Data Management: Controlling Data Volume, Velocity and Variety. *META Group Research Note, February 6*.

Detecting Locations from Twitter Messages

Invited Talk

Diana Inkpen

University of Ottawa
School of Electrical Engineering and Computer Science
800 King Edward Avenue, Ottawa, ON, K1N6N5, Canada
Diana.Inkpen@uottawa.ca

1 Extended Abstract

There is a large amount of information that can be extracted automatically from social media messages. Of particular interest are the topics discussed by the users, the opinions and emotions expressed, and the events and the locations mentioned. This work focuses on machine learning methods for detecting locations from Twitter messages, because the extracted locations can be useful in business, marketing and defence applications (Farzindar and Inkpen, 2015).

There are two types of locations that we are interested in: location entities mentioned in the text of each message and the physical locations of the users. For the first type of locations (task 1), we detected expressions that denote locations and we classified them into names of cities, provinces/states, and countries. We approached the task in a novel way, consisting in two stages. In the first stage, we trained Conditional Random Field models with various sets of features. We collected and annotated our own dataset for training and testing. In the second stage, we resolved cases when more than one place with the same name exists, by applying a set of heuristics (Inkpen et al., 2015).

For the second type of locations (task 2), we put together all the tweets written by a user, in order to predict his/her physical location. Only a few users declare their locations in their Twitter profiles, but this is sufficient to automatically produce training and test data for our classifiers. We experimented with two existing datasets collected from users located in the U.S. We propose a deep learning architecture for the solving the task, because deep learning was shown to work well for other natural language processing tasks, and because standard classifiers were already tested for the user location task. We designed a model that predicts the U.S. region of the user and his/her U.S. state, and another model that predicts the longitude and latitude of the user's location. We found that stacked denoising auto-encoders are well suited for this task, with results comparable to the state-of-the-art (Liu and Inkpen, 2015).

2 Biography

Diana Inkpen is a Professor at the University of Ottawa, in the School of Electrical Engineering and Computer Science. Her research is in applications of Computational Linguistics and Text Mining. She organized seven international workshops and she was a program co-chair for the AI 2012 conference. She is in the program committees of many conferences and an associate editor of the Computational Intelligence and the Natural Language Engineering journals. She published a book on Natural Language Processing for Social Media (Morgan and Claypool Publishers, Synthesis Lectures on Human Language Technologies), 8 book chapters, more than 25 journal articles and more than 90 conference papers. She received many research grants, including intensive industrial collaborations.

Acknowledgments

I want to thank my collaborators on the two tasks addressed in this work: Ji Rex Liu for his implementation of the proposed methods for both tasks and Atefeh Farzindar for her insight on the first task. I also thank the two annotators of the corpus for task 1, Farzaneh Kazemi and Ji Rex Liu.

This work is funded by the Natural Sciences and Engineering Research Council of Canada.

References

Atefeh Farzindar and Diana Inkpen. 2015. Natural Language Processing for Social Media. Morgan and Claypool Publishers. Synthesis Lectures on Human Language Technologies.

Diana Inkpen, Ji Liu, Atefeh Farzindar, Farzaneh Kazemi and Diman Ghazi. 2015. Location Detection and Disambiguation from Twitter Messages. In Proceedings of the 16th International Conference on Intelligent Text Processing and Computational Linguistics (CICLing 2015), LNCS 9042, Cairo, Egypt, pp. 321-332.

Ji Liu and Diana Inkpen. 2015. Estimating User Location in Social Media with Stacked Denoising Autoencoders. In Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing, NAACL 2015, Denver, Colorado, pp. 201-210.

Opinion Mining: Taking into account the criteria!

Pascal Poncelet
LIRMM - UMR 5506
Campus St Priest - Bâtiment 5
860 rue de Saint Priest
34095 Montpellier Cedex 5 - France
and
UMR TETIS Montpellier - France
Pascal.Poncelet@lirmm.fr

Abstract

Today we are more and more provided with information expressing opinions about different topics. In the same way, the number of Web sites giving a global score, usually by counting the number of stars for instance, is also growing extensively and this kind of tools can be very useful for users interested by having a general idea. Nevertheless, sometimes the expressed score (e.g. the number of stars) does not really reflect what it is expressed in the text of a review. Actually, extracting opinions from texts is a problem that have been extensively addressed in the last decade and very efficient approaches are now proposed to extract the polarity of a text. In this presentation we focus on a topic related with opinion but rather than considering the full text we are interested with the opinions expressed on specific criteria. First we show how criteria can be automatically learnt. Second we illustrate how opinions are extracted. By considering criteria we illustrate that it is possible to propose new recommender systems but also to evaluate how opinions expressed on the criteria evolve over time.

1 Introduction

Extracting opinions that are expressed in a text is a topic that have been addressed extensively in the last decade (e.g. (Pang and Lee, 2008)). Usually proposed approaches mainly focus on the polarity of a text: *this text is positive, negative or even neutral*. Figure 1 shows an example of a review on a restaurant.

Actually this review has been scored quite well: 4 stars over 5. Any opinion mining tools will show that the review is much more negative than positive. Let us go deeper on this exemple. Even if

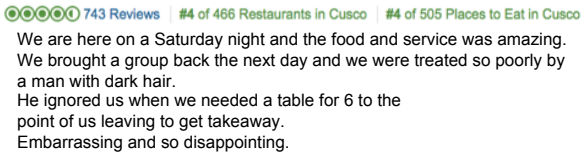


Figure 1: An example of a review

the review is negative it clearly illustrates that the reviewer was mainly disappointed by the service: he was in the Restaurant and found it amazing. We could imagine that, at that time, the service was not so bad. This exemple illustrates the problem we address in the presentation: we do not focus on a whole text rather we would like to extract opinions related to some specific criteria. Basically, by considering a set of user-specified criteria we would like to highlight (and obviously extract opinions) only on the relevant parts of the reviews focusing on these criteria. The paper is organized as follows. In Section 2 we give some ideas on how to automatically learn terms related to a criterium. We give also some clues for extracting opinions to the criteria in Section 3. Finally Section 4 concludes the paper.

2 Automatic extraction of terms related to a criterium

First of all we assume that the end user is interested in a specific domain and some criteria. Let us imagine that the domain is movie and the two criteria are actor and scenario. For each criterium we only need to have several keywords or terms of the criterium (seed of terms). For instance in the movie domain: **Actor=** {actor, acting, casting, character, interpretation, role, star} and **Scenario=**{scenario, adaptation, narrative, original, scriptwriter, story, synopsis}. Intuitively two different sets may exist. The first one corresponding to all the terms that may be used for a criterium. Such a set is called a *class*. The second

one corresponds to all the terms which are used in the domain but which are not in the class. This set is called *anti-class*. For instance the term *theater* is about movie but is not specific neither to the class actor nor scenario. Now the problem is to automatically learn the set of all terms for a class. Using experts or users to annotate documents is too expensive and error-prone. By the way there are many documents available on the internet having the terms of the criteria that can be learned. In a practical way by using a research engine it is easy and possible to get these documents. For instance, the following query expressed in Google: "+movie +actor -scenario adaptation narrative original -scriptwriter story -synopsis" will extract a set of documents of the domain movie (character $+$), having actor in the document and without (character $-$) scenario, adaptation, etc. In other terms we are able to automatically extract movie documents having terms only relative to the class actor. By performing some text preprocessing and taking into account a frequency of a term in a specific window of terms (see (Duthil et al., 2011) for a full description of the process as well as the measure that can be used to score the terms) we can extract quite relevant terms: the higher the score, the higher the probability of this term belonging to a class. Nevertheless as the number of documents to be analyzed is limited, some terms may not appear in the corpus. Usually these terms will have more or less the same score both in the class and the anti-class. They are called *candidates* and as we do not know the most appropriate class, a new query on the Web will extract new documents. Here again, a new score can be computed and all the terms with their associated scores can finally be stored in lexicons. Such lexicon can then be used to automatically segment a document for instance.

3 Extracting opinions

A quite similar process may be adapted for extracted terms used to express opinions: adjectives, verbs and even grammatical patterns such as $<$adverb $+$ adjective $>$ in order to automatically learn positive and negative expressions. Then by using the new opinion lexicon extracted we can easily detect the polarity of a document. In the same way by using the segmentation performed in the previous step it is now possible to focus on criteria and then extract the opinion for a specific criterium. Interested reader may refer to (Duthil et al., 2012).

4 Conclusion

In the presentation we will present more in detail the main approach. Conducted experiments that will be presented during the talk will show that such an approach is very efficient when considering Precision and Recall measures. Furthermore some practical aspects will be addressed: how many documents? how many seed terms? the quality of the results for different domains? We will also show that such lexicons could also be very useful for recommending systems. For instance we are able to focus on the criteria that are addressed by newspapers and then recommend the end user only with a list of newspapers he/she could be interested in. In the same way, evaluating how opinions evolve over time on different criteria is of great interest for many different applications. Interested reader may refer to (Duthil, 2012) for different applications that can be defined.

Acknowledgments

The presented work has been done mainly during the Ph.D of Dr. Benjamin Duthil and in collaboration with Gérard Dray, Jacky Montmain, Michel Plantié from the Ecole des Mines d'Alès (France) and Mathieu Roche from the University of Montpellier (France).

References

B. Duthil, F. Trousset, M. Roche, G. Dray, M. Plantié, J. Montmain, and Pascal Poncelet. 2011. Towards an automatic characterization of criteria. In *Proceedings of the 22nd International Conference on Database and Expert Systems Applications (DEXA 2011)*, pages 457–465, Toulouse, France. Springer Verlag.

B. Duthil, F. Trousset, G. Dray, J. Montmain, and Pascal Poncelet. 2012. Opinion extraction applied to criteria. In *Proceedings of the 23rd International Conference on Database and Expert Systems Applications (DEXA 2012)*, pages 489–496, Vienna, Austria. Springer Verlag.

B. Duthil. 2012. *Détection de critères et d'opinion sur le Web (In French)*. Ph.D. thesis, Université Montpellier 2, France.

B. Pang and L. Lee. 2008. Opinion mining and sentiment analysis. *Foundations and Trend in Information Retrieval*, 2(1-2):1–135.

How to deal with heterogeneous data?

Mathieu Roche
UMR TETIS (Cirad, Irstea, AgroParisTech) – France
mathieu.roche@cirad.fr
LIRMM (CNRS, University of Montpellier) – France
mathieu.roche@lirmm.fr
Web: www.textmining.biz

1 Introduction

The Big Data issue is traditionally characterized in terms of 3 V, i.e. volume, variety, and velocity. This paper focuses on the variety criterion, which is a challenging issue.

Data heterogeneity and content

In the context of Web 2.0, text content is often heterogenous (i.e. lexical heterogeneity). For instance, some words may be shortened or lengthened with the use of specific graphics (e.g. emoticons) or hashtags. Specific processing is necessary in this context. For instance, with an opinion classification task based on the message `SimBig is an aaaaattractive conference!`, the results are generally improved by removing repeated characters (i.e. `a`). But information on the sentiment intensity identified by the character elongation is lost with this normalization. This example highlights the difficulty of dealing with heterogeneous textual data content.

The following sub-section describes the heterogeneity according the document types (e.g. images and texts).

Heterogeneity and document types

Impressive amounts of high spatial resolution satellite data are currently available. This raises the issue of fast and effective satellite image analysis as costly human involvement is still required. Meanwhile, large amounts of textual data are available via the Web and many research communities are interested in the issue of knowledge extraction, including spatial information. In this context, image-text matching improves information retrieval and image annotation techniques (Forestier et al., 2012). This provides users with a more global data context that may be useful for experts involved in land-use planning (Alatrista Salas et al., 2014).

2 Text-mining method for matching heterogenous data

A generic approach to address the heterogeneity issue consists of extracting relevant features in documents. In our work, we focus on 3 types of features: thematic, spatial, and temporal features. These are extracted in textual documents using natural language processing (NLP) techniques based on linguistic and statistic information (Manning and Schütze, 1999):

- The extraction of **thematic information** is based on the recognition of relevant terms in texts. For instance, terminology extraction techniques enable extraction of single-word terms (e.g. `irrigation`) or phrases (e.g. `rice crops`). The most efficient state-of-the-art term recognition systems are based on both statistical and linguistic information (Lossio-Ventura et al., 2015).

- Extracting **spatial information** from documents is still challenging. In our work, we use patterns to detect these specific named entities. Moreover, a hybrid method enables disambiguation of spatial entities and organizations. This method combines symbolic approaches (i.e. patterns) and machine learning techniques (Tahrat et al., 2013).

- In order to extract **temporal expressions** in texts, we use rule-based systems like HeidelTime (Strötgen and Gertz, 2010). This multilingual system extracts temporal expressions from documents and normalizes them. HeidelTime applies different normalization strategies depending on the text types, e.g. news, narrative, or scientific documents.

These different methods are partially used in the projects summarized in the following section. More precisely, Section 3 presents two projects

that investigate heterogeneous data in agricultural the domain.[1]

3 Applications in the agricultural domain

3.1 Animal disease surveillance

New and emerging infectious diseases are an increasing threat to countries. Many of these diseases are related to globalization, travel and international trade. Disease outbreaks are conventionally reported through an organized multilevel health infrastructure, which can lead to delays from the time cases are first detected, their laboratory confirmation and finally public communication. In collaboration with the CMAEE[2] lab, our project proposes a new method in the epidemic intelligence domain that is designed to discover knowledge in heterogenous web documents dealing with animal disease outbreaks. The proposed method consists of four stages: data acquisition, information retrieval (i.e. identification of relevant documents), information extraction (i.e. extraction of symptoms, locations, dates, diseases, affected animals, etc.), and evaluation by different epidemiology experts (Arsevska et al., 2014).

3.2 Information extraction from experimental data

Our joint work with the IATE[3] lab and AgroParisTech[4] deals with knowledge engineering issues regarding the extraction of experimental data from scientific papers to be subsequently reused in decision support systems. Experimental data can be represented by n-ary relations, which link a studied topic (e.g. food packaging, transformation process) with its features (e.g. oxygen permeability in packaging, biomass grinding). This knowledge is capitalized in an ontological and terminological resource (OTR). Part of this work consists of recognizing specialized terms (e.g. units of measures) that have many lexical variations in scientific documents in order to enrich an OTR (Berrahou et al., 2013).

[1] `http://www.textmining.biz/agroNLP.html`

[2] Joint research unit (JRU) regarding the control of exotic and emerging animal diseases – `http://umr-cmaee.cirad.fr`

[3] JRU in the area of agro-polymers and emerging technologies – `http://umr-iate.cirad.fr`

[4] `http://www.agroparistech.fr`

4 Conclusion

Heterogenous data processing enables us to address several text-mining issues. Note that we integrated the knowledge of experts in the core of research applications summarized in Section 3. In future work, we plan to investigate other techniques dealing with heterogeneous data, such as visual analytics approaches (Keim et al., 2008).

References

H. Alatrista Salas, E. Kergosien, M. Roche, and M. Teisseire. 2014. ANIMITEX project: Image analysis based on textual information. In *Proc. of Symposium on Information Management and Big Data (SimBig), Vol-1318, CEUR*, pages 49–52.

E. Arsevska, M. Roche, R. Lancelot, P. Hendrikx, and B. Dufour. 2014. Exploiting textual source information for epidemio-surveillance. In *Proc. of Metadata and Semantics Research - 8th Research Conference (MTSR) - Communications in Computer and Information Science, Volume 478*, pages 359–361.

S.L. Berrahou, P. Buche, J. Dibie-Barthelemy, and M. Roche. 2013. How to extract unit of measure in scientific documents? In *Proc. of International Conference on Knowledge Discovery and Information Retrieval (KDIR), Text Mining Session*, pages 249–256.

G. Forestier, A. Puissant, C. Wemmert, and P. Gançarski. 2012. Knowledge-based region labeling for remote sensing image interpretation. *Computers, Environment and Urban Systems*, 36(5):470–480.

D.A. Keim, F. Mansmann, J. Schneidewind, J. Thomas, and H. Ziegler. 2008. Visual analytics: Scope and challenges. In *Visual Data Mining*, pages 76–90. Springer-Verlag.

J.A. Lossio-Ventura, C. Jonquet, M. Roche, and M. Teisseire. 2015. Biomedical term extraction: overview and a new methodology. *Information Retrieval Journal (IRJ) - special issue "Medical Information Retrieval", to appear*.

C.D. Manning and H. Schütze. 1999. *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, MA, USA.

J. Strötgen and M. Gertz. 2010. Heideltime: High quality rule-based extraction and normalization of temporal expressions. In *Proc. of International Workshop on Semantic Evaluation*, pages 321–324.

S. Tahrat, E. Kergosien, S. Bringay, M. Roche, and M. Teisseire. 2013. Text2geo: from textual data to geospatial information. In *Proc. of International Conference on Web Intelligence, Mining and Semantics (WIMS)*.

Rich Data: Risks, Issues, Controversies & Hype

Osmar R. Zaïane
Department of Computing Science
University of Alberta, Canada
zaiane@cs.ualberta.ca

Abstract

Big data technology is being adopted in industry and government at a record rate. A large number of enterprises believe big data analytics will redefine the competitive landscape of their industries within the next few years. Adoption is now perceived as a matter of survival. The unprecedented accumulation of data in almost all industries and government is unquestionable, but is the extravagant promotion of the technology justifiable? Is the technology ready to deliver on the promises? And is the fear driving the technology adoption reasonable? We will try to shed some light on the current state of rich data.

1 Introduction

The continuously increasing deluge of complex data we experience today is undeniable. If there is a hype about big data it is not about whether it is already upon us, but possibly on the expectations about what we can currently attain from its analytics. The buzzword Big Data is unfortunately a misnomer. It is an inaccurate term since it is misleading to understand the real significance of the idiom, even for specialists in information technology. Most focus on the qualifier "Big" to emphasize solely the size and miss the most important nature of the medium, the complexity of the data. Big Data refers to the massive amounts of complex data that are difficult to manipulate and understand using traditional processing methods. The complexity is not only due to the size but many other factors we highlight later. Thus, we advocate the designation Rich Data. What added to the confusion is the issue of the journal Nature on Big Data (Doctorow, 2008) that mainly centered on the issue of size. Big data was originally used rhetorically (Anderson, 2008) indicating that big is a fast moving target when it comes to data. What we considered large is not anymore, and what we consider huge today will not be soon. For the originators of the term (Anderson, 2008) Big Data typically meant applying tools such as Machine Learning to vast data beyond that captured in standard databases. Examples of such data include web browsing trails, social media, sensor data, surveillance data, etc. Based on this definition, big data is today ubiquitous and inescapable. This definition also hints to the moving target again; big data refers to rich complex data for which existing methods for storage, indexing, processing and analyzes are inadequate and new methods are required. As soon as solutions are found, big data is again something else for which methods have yet to be devised. The clever IBM marketing team has presented Big Data in terms of four dimensions now commonly known as the 4 Vs: Volume, Velocity, Variety, and Veracity (IBM, 2015). Today, most see big data as a mix of structured, semi-structured, and unstructured data, which typically breaks barriers for traditional relational database storage and breaks the limits of indexing by "rows". Hence the emergence of No-SQL and NewSQL data stores using a simple data model based on key-value pairs (Grolinger, 2013). This data also typically requires intensive pre-processing before each query to extract "some structure", particularly when it comes to text, and entails massively parallel and distributed computing with Map-Reduce type operations; to the point that Hadoop, an open source framework implementing the Map-Reduce processing, is becoming synonymous with Big Data (White, 2012). In reality, there is no standard recipe or architecture for big data. Big Rich Data is when we have complex data coming from disparate data sources that require integration to extract real value. Each problem is unique, hence the need for data scientists, who are not only data analysts but specialists contemplating holistic solutions considering infrastructures for data storage and management, as well as methods for aggregating,

analyzing and visualizing data and patterns. Data scientists do not work in isolation but in teams bringing together different skills in data analytics.

2 The Famous Vs of Big Data

IBM is credited for introducing the dimensions of Big Data. They were initially three (Volume, Velocity and Variety) and later augmented with Veracity (IBM, 2015). In fact, other Vs have been proposed later by other data science experts. We introduce herein 7 Vs.

Volume: refers to the size of the data which is typically very large. We are indeed awash with data, be it scientific data, data generated from activities on the web, acquired from sensors or collected from social media. We have an enormous volume of data at our disposal and are witnessing an exponential growth. However, not all problems with large volume of data are big data problems, and not all big data problems are concerned with very large data.

Velocity: is concerned with the speed of data creation and the speed of change. Sensors continuously transmit their measures; trades are done in milliseconds; credit card transactions are conducted world-wide uninterruptedly; social media messages go constantly viral in minutes. This velocity of the data is equated to a firehose of data from which we can read the data only once and having to analyze it while it is being generated. Velocity for rich data refers also to the speed of required analysis. Analysis and reporting of the results are also constraint with time.

Variety: refers to the different types of data we can now use, but more importantly refers to the vast array of data sources at our disposal. In the past, applications mainly exploited numerical and categorical data stored in relational tables, called structured data; with Rich Data applications we need to harness differed types of data including, images, video sequences, voice, time series, text messages from social media, and last but not least the relationships between data objects such as in social networks. Variety comes also from the availability of myriad independent data sources sometimes even from the public domain, such as open-data or from the Web. Acquiring and integrating additional data to the available one enhances the insights that can be obtained from the original data.

Veracity: Available data is often uncertain, particularly when acquired from sources over which we do not have control, such as social media. Veracity refers to ascertaining the accuracy of the analysis results or understanding of the discovered information when uncertainty prevails in the source data. The volume of data often makes up for the lack of quality or accuracy, but models that provide probabilistic results are preferred to measure some trust in the results.

Value: refers to the capacity to transform data into value, and more often the value is in the integration of data from different autonomous sources. The power of big data is to leverage additional independent data sources to better extract actionable knowledge and new information from an original dataset to bring more value in a decision making process.

Visualization: encompasses the reporting of the results of the analysis and effectively communicating actionable knowledge to decision makers. Visualization is the art of coalescing complex information into one 2D or 3D possibly interactive image. It is the essential lens through which one can see and understand the patterns and the relationships in the data.

Vulnerability: pertains to the privacy of the data that could be jeopardized. This is often the forgotten V. Even when dealing with anonymized data, when combining with additional data from other separate sources, the integration can reveal previously undisclosed information and thus expose private evidence. Data anonymization is typically attacked and compromised by combining data sources, which is the essence of big data. Privacy preserving techniques need to be intrinsic to big data analytics.

3 The Value is in Data Integration

A concrete example can illustrate the spirit of big data analytics. In 2006, wanting to improve on its Cinematch recommender system, the Netflix company launched a $1M challenge to whom would improve the results of their algorithm by at least 10%. The competition was clear on not to use other data sources but the 100M ratings in a sparse matrix of 500k users and 17k movies. It took about 3 years to win the prize with an improvement equivalent to 1/10 of a star. The solution was too convoluted for Netflix to implement. It was not the complexity of the solution the main reason for dropping it, but the realization that using additional information such as the Internet Movie Database (IMDB) with information on actors, directors, etc. and their relationships as well as sentiment in reviews could provide additional value to the ratings in the matrix to deliver better results for a recommender system with a more powerful

predictive model (Amatriain, 2013). The lesson learned is that one should always exploit all obtainable data, not just the data available at hand.

4 The Pitfalls & Challenges of Big Data

There is hype when the rate of adoption out paces the ordinary evolution of the technology and to avoid a quick disillusionment towards the technology one must manage to balance between the expectations and the promises. This same imbalance led to the disappointment toward Artificial Intelligence and its relinquishment by the major funders in the 1970s and again in the late 1980s, periods known as the winters of AI. It is debated whether Big Data would know such winter with a serious dwindling of the investment. The value of data is commonly agreed upon, yet very few know how to profit of this data for competitive advantage. Where big data has undeniably seen success is in consumer behaviour prediction but the very quick adoption is touching all industries and government. Many have invested significant amounts of money in the technology mainly by fear of missing the train of opportunity, but the interest can fade since many are failing to realize and operationalize the value that lies in big data and the voluminous investment that comes with it. For the adoption to endure and to drive more innovation, the community must be more mindful of the technology and cognizant of the pitfalls and challenges. We highlight some herein.

Few years ago an authoritative report created a stir in the industry. The McKinsey Report asserted that in the US alone there will be a shortage by 2018 of up to 190,000 data scientists (Manyika, 2011). This led the Harvard Business Review to state data scientist as being the "Sexiest Job" in this century (Davenport, 2012). Training more data scientists with deep analytical skills is becoming a necessity. Meanwhile, with the current void, we have many that deceptively claim knowledge and skills in data science which could contribute to the disillusionment. The McKinsey Report also stressed the necessity to educate managers in the know-how to use the analysis of big data to make effective decisions. Educating managers gives them the opportunity to leverage the skills of their data science team and surely take advantage of big data analytics.

Another important downside is one of the least insisted upon V of big data: Veracity. The voluminous size is a curse for big data as with vast data, patterns can happen by chance but these patterns may have no predictive power. Like with statistics, facts in data are vulnerable to misuse and with this pliability of data one can make it mean anything. Data per se does not create meaning but data analysts make it express the hidden information and bring forth the crucial interpretation. As Susan Etlinger articulated it: "Critical Thinking is the killer app for Big Data" (Etlinger, 2014). Hence the need for the data context, known as metadata. Metadata, describing the data, should be created at the source, should journey with the data, managed with the data, exploited during analysis, and used for interpretation. A pillar component of big data is data fusion, but integrating data cannot be safely accomplished without using metadata. Metadata is also paramount for the interpretation of patterns as well as visualizing and clarifying analysis results. Likewise, visualization is still a neglected limitation while it is of paramount importance in any complete data mining process as it conveys the final discoveries (Fayyad, 2001). Visualization, the visual reporting of discoveries from data, is not a real science but an art; the art of conveying patterns from a high dimensional space in 2D representation, possibly interactively, without losing information while highlighting the essential and actionable. One typical mistake is not to work with skilled artists and trained communication specialists who have different perspectives and think outside the box to produce such required visualizations.

Big Data carries challenges for the scientific community. The challenges are numerous which represent huge opportunities for research and innovation. The first challenge is obviously the scale. The trend is going towards collecting even more data and the advent of the Internet of Things will only be a multiplier (Greengard, 2015). The challenge is not only building the required infrastructure to store and manage the data but also analyzing it efficiently and obtain the valuable insights. The popular MapReduce concept has become the generic programming model used to store and process large scale datasets on commodity hardware clusters. However, not all problems are "map-reducible". New initiatives for massive distributed computing, such as the Spark framework (Karau, 2013), are already being introduced. Another defiant problem is due to data heterogeneity from various sources and data inconsistencies between sources. While data integration and record linking has a long tradition in the database research community, it is still in its infancy when

it comes to rich data and its complexities. Combining different data sources brings additional challenges such as data incompleteness and uncertainty, which again highlight the importance of Veracity. Last but not least, combining data sources also creates a possible confrontation with data privacy. Truly privacy-preserving data mining techniques can compromise data utility (Wu, 2013). Anonymization approaches add perturbations to generate altered versions of the data with additional uncertainties. It remains that data sharing in big data raises many security and privacy concerns. Another overlooked challenge is the one due to the data dimensionality explosion (Wu, 2014). Big Data is also concerned with large dynamic and growing complex data. In this active data, not only are we faced with high and diverse dimensionality issues, but the dimensions keep changing with new additions, disappearances and modifications. The ultimate challenge is automation of the big data analytics such as with autonomous vehicles. There is a trend towards analysis for non-data scientists; creating generic mechanized systems taking disparate data sources as input and producing reports with a push of a button.

5 Conclusion

We are constantly bombarded by stories about how much data there is in the world and how traditional solutions are too slow, too small, or too expensive to use for such large data, but when it comes to Rich Data and the challenges of interpreting it, size is not everything. There is also speed at which it is created and the variety of it and its complexity of types and sources. Because Big Data could improve decision making in myriad fields from business to medicine allowing decisions to be based on data and data analysis, large corporations have adopted Big Data in their decision making, predominantly in marketing and customer behavior analysis. Big Data is only getting worse in terms of volume, speed, availability of sources and complexity, and most sectors of the economy are data-driven decision making. Therefore, big data is not just a buzzword anymore, but to avoid a hype we must manage the realistic expectations. Otherwise, people may be quickly disappointed by not getting what is promised and what is currently possible. A common gaffe is to focus on infrastructure, yet a holistic solution is required: data linking is part of the solution, new hardware is part of the solution, and new algorithms are part of the solution. The key is to deploy all means to be able to exploit all the data that is obtainable to enhance insights and possible actions.

Reference

Chris Anderson, 2008, The Petabyte Age: Because More Isn't Just More — More Is Different, Wired Magazine, Issue 16.07

Xavier Amatriain, 2013, Mining large streams of user data for personalized recommendations. ACM SIGKDD Explorations Newsletter, 14(2):3748,

Randal E. Bryant, Randy H. Katz, Edward D. Lazowska, 2008, Big-Data Computing: Creating revolutionary breakthroughs in commerce, science, and society, Computing Community Consortium.

Thomas Davenport, D.J. Patil, 2012, Data Scientist: The sexiest Job of the 21st Century, Harvard Business Review

Cory Doctorow, 2008, Welcome to the petacentre, Big Data Special issue, Nature 455, 1

Susan Etlinger, 2014, Critical Thinking: The Killer App for Big Data, TED Talk, https://www.ted.com/talks/susan_etlinger_what_do_we_do_with_all_this_big_data

Usama Fayyad, Georges Grinstein, Andreas Wierse, 2001, Information Visualization in Data Mining and Knowledge Discovery, Morgan Kaufmann.

Samuel Greengard, 2015, The Internet of Things, MIT Press

Katarina Grolinger, Wilson Higashino, Abhinav Tiwari, Miriam Capretz, 2013, Data management in cloud environments: NoSQL and NewSQL data stores, Journal of Cloud Computing: Advances, Systems and Applications, 2:22, Springer.

IBM, 2015, The Four V's of Big Data, http://www.ibmbigdatahub.com/infographic/four-vs-big-data

Holden Karau, 2013, Fast Data Processing with Spark, Packt Publishing

James Manyika, Michael Chui, Brad Brown, Jacques Bughin, Richard Dobbs, Charles Roxburgh, Angela Hung Byers, 2011, Big data: The next frontier for innovation, competition, and productivity, McKinsey Global Institute

Tom White, 2012, Hadoop: The definitive guide, O'Reilly Media, Inc.

Lengdong Wu, Hua He, Osmar Zaiane, 2013, Utility Enhancement for Privacy Preserving Health Data Publishing, International Conference on Advanced Data Mining and Applications

Xindong Wu, Xingquan Zhu, Gong-Qing Wu, Wei Ding, 2014, Data Mining with Big Data, IEEE Transactions on Knowledge & Data Engineering, vol.26, Issue 1.

Dependency-based Topic-Oriented Sentiment Analysis in Microposts

Prasadith Buddhitha and Diana Inkpen

University of Ottawa
School of Information Technology and Engineering
Ottawa, ON, K1N6N5, Canada
pkiri056@uottawa.ca and Diana.Inkpen@uottawa.ca

Abstract

In this paper, we present a method that exploits syntactic dependencies for topic-oriented sentiment analysis in microposts. The proposed solution is based on supervised text classification (decision trees in particular) and freely-available polarity lexicons in order to identify the relevant dependencies in each sentence by detecting the correct attachment points for the polarity words. Our experiments are based on the data from the Semantic Evaluation Exercise 2015 (SemEval-2015), task 10, subtask C. The dependency parser that we used is adapted to this kind of text. Our classifier that combines both topic- and sentence-level features obtained very good results (comparable to the best official SemEval-2015 results).

1 Introduction

Identifying opinionated factual information has become widely popular during the current years. The growth of social media has enhanced the amount of information being shared among groups of people as and when it is being generated due to various activities. With the availability of various less-complex and economical telecommunication media, human expression has become frequently embedded within the information being transmitted. Such freely available information has attracted many stakeholders with a wide range of interdisciplinary interests. Microblogs have become one of the most popular and widely-used sources of information, where users share real-time information on many topics.

Twitter has become one of the most popular microblogging platforms in recent years. According to Twitter (2015), 500 million tweets are being posted per day with 302 million monthly active users. As more and more interest has emerged in identifying the key information contained within the messages, greater difficulties are being introduced due to the informal nature of the message representation. With the limitation of 140 characters, the informal nature of the messages has introduced slang, new words and terminology, URLs, creative spelling, misspelling, punctuations and abbreviations such as #hashtag and "re-tweet" (RT).

With the representation of valuable information about one or more interests enriched with user perception and the sheer amount of volume has challenged the researchers in Natural Language Processing and Machine Learning to generate mechanisms to extract the valuable information, which could be beneficial for the interested parties from different domains, such as marketing, finance, and politics. Identification of the perception, which could also be termed as opinion mining or sentiment analysis has resulted in many researches based on supervised and unsupervised learning methods.

The widely-spread enthusiasm in the Twitter sentiment analysis is supported by various research-based events such as the Semantic Evaluation Exercise (SemEval). The research we present in this paper is based on the SemEval 2015 task 10, dedicated to Sentiment Analysis in Twitter. The task is subdivided into four subtasks emphasizing different levels such as expression, message, topic and trend (SemEval-2015).

We focus on "topic-based message polarity classification"; that is, given a message and a topic, we classify whether the message is of positive, negative, or neutral sentiment towards the given topic (SemEval-2015). The task will be approached through the use of sentiment lexicons at both topic and sentence level. Our solution

will use several freely available general-purpose sentiment lexicons and tweet-specific sentiment lexicons, the latter provided by the National Research Council (NRC) of Canada.

The following paragraphs will briefly define the different terminologies being used in the rest of this paper.

Tokenization: Text normalization is a key part in Natural Language Processing (NLP), which is commonly being used in many NLP tasks including the proposed solution. Tokenization can be considered as one of the initial and key functions in text normalization where a given text is divided in to a sequence of characters, which can be treated as a group. The character sequence can be treated as a word or any other token such as a punctuation mark or a number (Jurafsky & Martin, 2008).

Sentiment analysis: As described in Scherer's typology of affective states, sentiment analysis can be defined as detecting attitudes (Scherer, 1984). The polarity can be identified as a type of attitude, which could be categorized into one of the states such as positive, negative or neutral, as well as being assigned with a weighted value indicating the strength within the assigned category (Manning & Jurafsky, 2015).

N-grams: N-grams can be broadly defined as a contiguous sequence of words (Jurafsky & Martin, 2008). The N-grams can be represented as N-tokens, where the tokens could be words, letters, etc. Depending on the number of tokens, N-gram models can be termed as unigrams (N-gram with the size of one), 2-gram (bigram), 3-gram (trigram), four-gram or five-gram, which can be considered as the most commonly-used in statistical language models. The number of items within the language model can be based on the processing task. Our proposed solution mainly considers unigrams and bigrams.

Decision Trees: Decision trees can be explained in the most abstract form as if-then-else statements arranged in a tree. The most informative features extracted from the training data are according to their information gain (Quinlan, 1986). They have the advantage that the model leant is interpretable; the user can inspect the tree in order to understand why a certain decision was made. Decision trees do not always get the best results compared to other machine learning algorithms (but they happened to work very well for our particular task). The key step in making decision trees effective will be the selection of suitable features for our task. In our solution, the selected features are based on the polarity words from the sentence that are in dependency relations to the targeted topic.

2 Related Work

There has been a large volume of research on sentiment analysis. It started with identifying subjective and objective statements. Subjective statement can further be classified into positive, negative, or neutral, possibility with intensity level for the first two. Many researches have been done on opinion mining and sentiment analysis for customer reviews (Pang & Lee, 2008) and, more recently, on Twitter messages (Jansen, Zhang, Sobel, & Chowdury, 2009; Kouloumpis, Wilson, & Moore, 2011; Pak & Paroubek, 2010; Bifet, Holmes, Pfahringer, & Gavaldà, 2011).

Over the years many techniques have been adopted by researchers on Twitter sentiment analysis, such as lexical approaches and machine learning approaches (Fernandez, Gutierrez, G'omez, & Martinez-Barco, 2014) (Bin Wasi, Neyaz, Bouamor, & Mohit, 2014). Lexicon-based systems focused on creating repositories of words labeled with polarity values, possibly calculated based on the association of these words and with other words with known polarities (Fernandez et al., 2014). In addition, well-performing hybrid systems have also been proposed for Twitter sentiment analysis by combining hierarchical models based on lexicons and language modeling approaches (Balage Filho, Avanco, Pardo, & Volpe Nunes, 2014).

The large impact of using polarity lexicons in supervised learning can also be seen in the top seven-ranked participants in the SemEval-2015, task 10, subtask C. According to Boag, Potash, & Rumshisky (2015); Plotnikova et al. (2015); Townsend et al. (2015); Zhang, Wu, & Lan (2015) put emphasis on publicly available lexicons such as the NRC Hashtag Sentiment Lexicon, the Sentiment 140 Lexicon, the NRC Emotion Lexicon and SentiWordNet for feature engineering. In addition to lexicon features, many of the top scored systems used linguistic and Twitter-specific features. These systems have mainly used supervised machine learning implemented through classifiers such as Support Vector Machine (SVM) and logistic regression to obtain the best results. It is interesting to note that Townsend et al. (2015), ranked sixth for subtask C, have used the Stanford parser configured for short documents with the use of a

caseless parsing model. The authors have argued that TweeboParser (Kong et al., 2014), which is explicitly created for parsing Twitter messages, lacks in dependency type information due to the use of a simpler annotation scheme rather than using an annotation scheme like Penn Treebank. Kong et al. (2014) have argued that Penn Treebank annotations produce low accuracies specifically with informal text such as tweets and it is more suited for structured data, and due to this reason they have used a syntactically-annotated corpus of tweets (TWEEBANK). Despite these claims, the TweeboParser has achieved an accuracy of over 80% on unlabelled attachments. The parser has contributed nearly 10% accuracy increase in our proposed solution through topic-level feature extraction, which has accumulated towards a comparable Macro F1-measure of 0.5310 in contrast to a lower Macro F1 measure of 0.2279 obtained by Townsend et al. (2015) using the reconfigured Stanford parser.

As many effective sentiment analysis solutions are based on machine learning and lexicon-based techniques (Balage Filho et al., 2014), our proposed solution will also be focused on supervised machine learning that use features computed by using freely available lexicons, while focusing on general and Twitter-specific language constructs.

Many of the proposed solutions in sentiment analysis have used key natural language processing techniques such as tokenizing, part-of-speech tagging, and bag-of-words representation for preliminary preparation tasks (Bin Wasi et al., 2014; Mohammad & Turney, 2013; Kiritchenko, Zhu, & Mohammad, 2014). Due to the informal nature of the Twitter messages, text-preprocessing techniques have to be given special consideration. Bin Wasi et al. (2014), Mohammad & Turney (2013) and Kiritchenko et al. (2014) used the Carnegie Mellon University (CMU) ARK tools for tasks such as tokenizing and part-of-speech tagging, which handles text with Twitter-specific characteristics such as identifying special characters and tokens according to Twitter-specific requirements (Owoputi, O'Connor, Dyer, Gimpel, Schneider, & Smith, 2013). In addition to the CMU ARK tokenizer, our proposed solution uses the TweeboParser for Twitter text dependency parsing, which allows us to identify the syntactic structure of the Twitter messages.

It could be argued that supervised or semi-supervised machine learning techniques provide higher accuracy levels compared to unsupervised machine learning techniques and also the consideration must be given to the specific domain which the task is implemented on (Villena-Roman, Garcia-Morera, & Gonzalez-Cristobal, 2014). This is why we build a supervised classifier based on the SemEval training data, and we are planning to extend it in future work with a large amount of unlabeled Twitter data.

Many systems in the past gave little consideration to hashtags, but this has changed recently, as their impact on the sentiment value of a message was demonstrated. Research has been conducted by using hashtags as seeds representing positive and negative sentiment (Kouloumpis et al., 2011) and also by creating hashtag sentiment lexicons from a hashtag-based emotion corpus (Mohammad & Kiritchenko, 2015). The same lexicon created by Mohammad & Kiritchenko (2015) is being used by our proposed classifier to identify hashtags associated to opinions and emotions; we add a stronger emphasis on the hashtag representation.

According to Zhao, Lan, & Zhu (2014), emoticons are also considered to be providing considerable sentiment value towards the overall sentiment of a sentence. Emoticons were identified using different mechanisms such as through the use of Christopher Potts' tokenizing script (Mohammad, Kiritchenko, & Zhu, 2013). Our proposed solution has adopted the MaxDiff Twitter sentiment lexicon to identify both the emoticons and their associated sentiment values (Kiritchenko et al., 2014), as it will be described in section 4.2.

Many proposed solutions normalize the informal messages in order to assist in sentiment identification (Zhao et al., 2014; Bin Wasi et al., 2014). We do not need to do this, because the lexicons we used contain many such Twitter-specific terms and their associated sentiment values (Mohammad et al., 2013; Kiritchenko at al., 2014).

3 Data

The dataset is obtained from the SemEval-2015 Task 10 for subtask C[1]. The dataset constitute of trial and training data. The training data includes the Twitter ID, the target topic and the polarity towards the topic (SemEval-2015). Due to privacy reasons, the relevant Twitter messages were

[1] We did not participate in the task, we downloaded the data after the evaluation campaigned

not included and a separate script has being provided in order to download the messages. After executing the script, the message "Not Available" is being printed if the relevant tweet is no longer available.

Our final dataset contains 391 Twitter messages, out from 489 given Twitter IDs for the task, where 96 IDs were removed due to unavailability of the Twitter messages, one record due to a mismatched ID and one record because it was a duplicate ID. The original dataset included around 44 topics and approximately ten tweets per topic (SemEval-2015). From the extracted 391 tweets, 110 tweets were labeled with positive topic polarity, 44 as negative, 235 as neutral and 2 were off-topic. According to Rosenthal et al. (2015), having access to less training tweets does not have a substantial impact on the results being generated, because the task participants that used less training data have produced higher results.

In order to make the dataset more relevant and accurate, both URLs and usernames were normalized, where the URLs are renamed as http://someurl and the usernames as @someuser. The tweets were also tokenized using the tokenizing tool provided by Carnegie Mellon University (CMU), known as Tweet NLP.

The Twitter messages in our dataset were composed of only one sentence (and one target topic in the message), this is why in this paper, the terms "sentence-level" and "message-level" are used interchangeably. This is due to the short nature of the tweets (also they are rarely fully-grammatical sentences due to the informal communication style). In rare case, when a tweet might contain more than one sentence, for future test data, our method will use only the sentence(s) that contain the topic(s) of interest.

4 Experiments

Our experiments had the goal of building a supervised classifier that can decide whether the message is positive, negative or neutral towards the given topic.

The experiments were conducted in two parts where features were extracted at sentence level and topic level, using different lexicons. The following sections will describe our features and the tools that we used to extract them, mainly the dependency parser and the lexicons.

4.1 Dependency Parsing

The dependency parser for English tweets, TweeboParser from CMU, was used to generate the syntactic structure of each tweet. Given an input, a single tweet per line, an output of the tokenized tweet is generated with their associated part-of-speech tags and syntactic dependencies. The generated prediction file is structured according to the "CoNLL" format representing different columns such as, token position (ID), word form (FORM), coarse grained part-of-speech tag (CPOSTAG), fine grained part-of-speech tag (POSTAG), most importantly the head of the current token (HEAD) indicating the dependencies and the type of dependency relation (DEPREL) (Buchholz, 2006). The generated syntactic structure for the following tweet:
"They say you are what you eat, but it's Friday and I don't care! #TGIF (@ Ogalo Crows Nest) http://t.co/l3uLuKGk"
is presented in Table 1. For this example, there are several conjunctions (CONJ), and one multi-word expression (MWE) is identified. Some other dependency relations were missed in this case, due to the imperfect training of the parser on small amounts of Twitter data.

This example tweet is from our dataset, and according to the annotations provided by the SemeEval task, the target topic is "Crows Nest", the general message polarity is "positive", and the polarity towards the given topic is "neutral".

4.2 Feature Extraction

Feature extraction was conducted at sentence level and at topic level. Feature extraction was mainly based on sentiment lexicons. NRC Canada has provided several tweet-specific sentiment lexicons, which were used in capturing many Twitter specific content displaying different levels of polarity such as positive, negative and neutral, and also accompanied with a finite set of values representing evaluative intensity towards specific polarity categories (Kiritchenko et al., 2014). Mentioned below are the different lexicons being used at both sentence and topic levels.

ID	FORM	CPOS TAG	POS TAG	HEAD	DEPREL
1	They	O	O	2	_
2	Say	V	V	9	CONJ
3	You	O	O	4	_
4	Are	V	V	2	_
5	What	O	O	7	_
6	You	O	O	7	_
7	Eat	V	V	4	_
8	,	,	,	-1	_
9	But	&	&	0	_
10	it's	L	L	9	CONJ
11	Friday	^	^	10	_
12	And	&	&	0	_
13	I	O	O	14	_
14	don't	V	V	12	CONJ
15	Care	V	V	14	_
16	!	,	,	-1	_
17	#TGIF	#	#	-1	_
18	(@	P	P	0	_
19	Ogalo	^	^	21	MWE
20	Crows	^	^	21	MWE
21	Nest	^	^	18	_
22	)	,	,	-1	_
23	http://t.co/l3uLuKGk	U	U	-1	_

Table 1. TweeboParser output for a tweet.

The *NRC hashtag emotion lexicon* consists in a list of words and their association with eight emotions: anger, fear, anticipation, trust, surprise, sadness, joy and disgust. The association between the tweets and the emotions were calculated through the identification of emotion-word hashtags in tweets (Mohammad et al., 2013). The file is formatted according to category (e.g. anger, fear, etc.), the target word, and the associated score. The relevant score indicates the strength of the association between the category and the target word (Mohammad et al., 2013). Higher scores indicate stronger associations between the category and the target word (Mohammad et al., 2015).

The *NRC word-emotion association lexicon* contains a list of words and their association with eight emotions, anger, fear, anticipation, trust, surprise, sadness, joy and disgust, and also the polarity towards the relevant word represented either as positive or negative (Mohammad et al., 2013). The lexicon is structured according to the target word, the emotion category and the association value indicating to which category the word belongs. The value 1 indicates that it belongs to the relevant category; the value is 0 if it does not (Mohammad et al., 2013).

The *MaxDiff Twitter sentiment lexicon* represents unigrams with associative strength towards positive sentiment. The data was obtained by manual annotation through Amazon Mechanical Turk (Kiritchenko et al., 2014). Each entry of the lexicon consists of the term and its relevant associative values ranging from -1 indicating the most negative score and +1 indicating the most positive score (Mohammad et al., 2013).

Sentiment140 lexicon is a collection of words with the associated positive and negative sentiment (Mohammad et al., 2013). The lexicon is divided into unigrams and bigrams, where each entry contains the term, the sentiment score and the number of times the term appeared with a positive emoticon and the number of times the term appeared with a negative emoticon. The sentiment score is calculated using the pointwise mutual information (PMI), by subtracting the associated score of the term with negative emoticons from the associated score with positive emoticons (Mohammad et al., 2013).

SentiWordNet 3.0 was designed for assisting in opinion mining and sentiment classification in general (not for Twitter messages). SentiWordNet is a result of annotating WordNet synonym entries according to their polarity weighting (Sebastiani & Esuli, 2010). The scores given for positive, negative and neutral classes range between zero and one, and the summation of all three scores is 1. SentiWordNet 3.0 is based on WordNet version 3.0 and the entries include POS and ID columns identifying the associated WordNet synonym set.

4.3 Sentence level feature extraction

Sentence-level feature extraction is conducted mainly using the above-mentioned lexicons. Hashtags in a tokenized Twitter message were looked up in the hashtag emotion lexicon, and the scores were aggregated according to the associated values for each category of emotion. If the given hashtags are not being associated with any of the emotion classes, a value of zero is being returned for the sentence for the specific emotion class.

As an additional attribute, the aggregated emotion values were compared to the maximum value, which is being assigned as the representative nominal class for the given sentence.

In order to compute the features based on the word emotion lexicon, the tokenized Twitter message was matched against the lexicon and the associated values were aggregated according to each individual emotion class in order to represent the sentence

The MaxDiff Twitter sentiment lexicon is used to identify the aggregated scores for a sentence with the associated values given for unigrams. As the values represent positive sentiment towards a given word calculated using the MaxDiff method of annotation, positive and negative value aggregation has resulted in a representation of a sentiment value for the given tweet.

Also, SentiWordNet is used to obtain an aggregated value for the sentence by matching words between the tokenized tweet and the SentiWordNet synonym sets. In addition to the sentence level SentiWordNet score, the given topic in a message is also being evaluated against the synonym set to identify if it carries a sentiment value.

Tokenized Twitter bigrams are also being used to identify related bigram lexical entries against the "Sentiment140" lexicon. In total, at message level, the classifier was trained on nine features using the hashtag emotion lexicon, ten features using the word-emotion association lexicon, and one feature each using the MaxDiff Twitter sentiment lexicon and SentiWordNet. Also the Sentiment140 lexicon for unigrams and bigrams was used in identifying one feature each at message level.

4.4 Topic-level feature extraction

Topic-level feature extraction is implemented similar to sentence-level feature extraction using the above-mentioned lexicons. The key motivation behind the identification of the dependent words is the nature of the task, where it is required to identify the sentence polarity towards a given topic. It is noted that the sentence level polarity and the sentence polarity towards a given topic can be different, as the topic might or might not contribute towards the overall polarity of the sentence. Dependency parsing is being used mainly to identify the sentiment contribution made by the dependent tokens towards the topic, as all the tokens within the sentence might not contribute equally towards the sentiment of a sentence. In contrast to the feature extraction based on the associated tokens towards the left and right of the specific topic (Kiritchenko et al., 2014), the dependency token identification can be intuitively considered as an effective methodology due to the following reasons: the neighbouring tokens might emphasize less the sentiment value; and, most importantly, the token selection can be limited based on their dependency relation to the topic.

The output obtained from the TweeboParser is analyzed to identify both tokens being dependent on the topic and the relevant dependencies that the topic has with the rest of the tokens within a given sentence. The multiword topics are considered as units and the dependencies towards and from them are identified. Extracted topic dependencies are evaluated using the given lexicons to identify different attributes, as mentioned above under different lexicon features. The features are identified against both unigrams and bigrams according to the given lexicon.

In total, at topic level the classifier was trained on nine features using the hashtag emotion lexicon, ten features using the word-emotion association lexicon, two features using the SentiWordNet and one feature using the MaxDiff Twitter sentiment lexicon. In addition, the Sentiment140 lexicon with unigrams and bigrams was used to identify one feature each at topic level.

In summary, a total of 47 features were used to train the classifier (23 at message level and 24 at topic level) and considerable improvement was obtained by using both sentence- and topic-level features, as it will be described in the next section.

5 Results

The evaluation measure that we report is the one used in the SemEval task: the macro average F1 measure for the positive class and for the negative class (excluding the neutral class). The key reason that could be identified as the motivation behind the use of this macro F1-measure is the unequal distribution of the classes (the neutral class being dominant in the dataset).

As the first step in evaluation the most efficient and effective machine-learning algorithm to be used as the main classifier was identified as decision trees, compared with the results[2] obtained for different classifiers such as Support Vector Machines (SVM) and Naïve Bayes. Decision trees resulted[3] with the highest macro average F1 measure for both positive and the negative classes, given all the feature vectors.

[2] Comparing the weighted average F1 measure, the results obtained using a t-test with both sentence- and topic- level features for decision trees (0.64) was noticeably higher than SVM- (0.60) and statistically significant than Naïve Bayes-algorithm (0.44).

[3] Decision trees macro average F1 measure (0.48) was substantially higher than both SVM (0.39) and Naïve Bayes (0.35) macro average F1 measure.

To understand the impact of different features identified through lexicons and the impact sentence- and topic-level features has on the overall classifier performance, we separately ran the decision trees algorithm on sentence- and topic-level features. Table 2 illustrates the impact each lexicon has on the classification results, which could be identified by removing individual lexicons (one or more features) at sentence-level, and then at topic-level and by comparing with the results when using all the features.

Features	Macro F1-measure	
	Sentence level	Topic level
All	0.4435	0.3500
Remove Hashtag emotion lexicon	0.4925	0.3000
Remove MaxDiff Twitter sentiment lexicon	0.4435	0.3615
Remove Word-emotion association lexicon	0.4435	0.3500
Remove Sentiment140 lexicon (unigrams)	0.4270	0.0870
Remove Sentiment140 lexicon (bigrams)	0.4035	0.1770
Remove SentiWordNet	0.2115	0.2685

Table 2. Classification results based on different lexicons illustrated separately on sentence- and topic-level, by removing one lexicon at a time.

By analyzing the Table 2 results (compared with the baseline accuracies) as well as through attribute subset evaluation and also by calculating the information gain[4] with respect to the class on separate features at sentence- and topic-level, we could identify that SentiWordNet and Sentiment140 lexicon features have more influence on the classifier performance followed by Word-emotion, MaxDiff and Hashtag emotion lexicons.

By implementing different combinations of features, both at sentence- and topic-level, we showed that the most influential features were extracted using the following lexicons: SentiWordNet, Sentiment140 lexicon and NRC emotion lexicon.

Table 3 summarizes the results obtained for different combinations of features, at both sentence and topic level. The first line includes all features at both levels. The second line removes all the sentence-level features and keeps only topic-level features in the first column of results and removes all the topic-level features but keep the sentence level features in the second columns of results (the same as the first line of results in Table 2). Then the next lines remove one or more lexicons at a time from each level, and in the last three lines from both levels.

Features (Lexicons)	Macro F1-measure	
	Sentence level	Topic level
Include all features	0.4845	
Remove all features at one level but keep them for the other level	0.3500	0.4435
Sentiment140 lexicon (bigrams)	0.4680	0.4805
Sentiment140 lexicon (unigrams)	0.4730	0.4945
SentiWordNet	0.4745	0.4825
MaxDiff Twitter sentiment lexicon	0.4845	0.4995
Word-emotion association lexicon	0.4845	0.4845
Hashtag emotion	0.5140	0.4745
Hashtag + Word-emotion	0.5140	0.4745
Hashtag **+** Word-emotion + MaxDiff	0.5165	-
Hashtag + Word-emotion + MaxDiff + Sentiment140 lexicon (unigrams) (topic)	0.5230	
Hashtag + Word-emotion + MaxDiff (sentence) + MaxDiff (topic)	0.5275	
Hashtag + MaxDiff (sentence) + MaxDiff (topic)	0.5310	

Table 3. Comparison of the classification results generated using sentence- and topic-level features together, while removing subsets of features at sentence-level, at topic-level or at both levels.

[4] Information gain and attribute subset evaluation were not solely considered due to macro average F1 measure where it only considers the positive and negative polarities.

	Team	Twitter 2015
1	TwitterHawk	0.5051
2	KLUEless	0.4548
3	Whu_Nlp	0.4070
4	whu-iss	0.2562
5	ECNU	0.2538
6	WarwickDCS	0.2279
7	UMDuluth-CS8761	0.1899

Table 4. Official SemEval-2015 task 10 subtask C results.

6 Discussion

The results obtained were compared against the official results of the SemEval 2015 task 10 subtasks C. The top seven results from SemEval are presented in Table 4.

Comparatively, the proposed classifier using sentence and topic level features based on lexicons has obtained a macro-F1 score higher than the best result from Table 4. The good results that we obtained were mainly due to the use of the publicly available lexicons and the rich set of lexicons provided by NRC Canada through extensive research on sentiment analysis of short informal text. Both sentence and topic level features have contributed to the higher accuracy level while sentence level features can be identified as the main contributor. Use of topic level features identified through topic dependencies has provided a substantial improvement to the overall results by increasing the macro-F1 measure from 0.4435 (using only sentence level features) to 0.5310 (using both sentence- and topic-level features, but with less sentence level features compared to topic level features). We also showed that use of all the emotion features as a single feature with separate nominal classes achieved better results compared to having separate nominal classes for each emotion (sadness, fear, anger, etc.).

The best results of 0.5310 macro F1-score were obtained with the use of a combination of topic-level and sentence-level features. Although the topic-level features' contribution on top of the sentence-level features was small, the macro-F1 score for topic-level features only was 0.35, a good score in itself for this difficult task.

We also note that the impact on the F1-measure from the emotion and NRC MaxDiff lexicons at both sentence and topic level was at a lower range, while the majority of the impact was contributed by the SentiWordNet and Sentiment140 lexicons. It could be identified that the use of lexicon-based features within a classification task resulted in generating an accurate classifier as long as features at both sentence- and topic- level were considered.

7 Conclusions and Future Work

The identification of both sentence level features and topic level dependencies with the use of lexicons designed especially for short informal texts, such as tweets, have made our proposed solution to achieve very good results. It was also identified that introducing more features based on lexicons at both sentence- and topic- level could further increase the accuracy of the classifier.

In future work, in addition to lexicon-based features, factors that have high impact on sentiment such as identification of negation, part-of-speech tagging and tag frequencies could also be considered in order to improve the accuracy of the classifier. Further identification of dependency relations by training the dependency parser with additional dependency relation labels, could also improve the accuracy level of the classifier. We also plan to do more extensive testing, on large amounts of tweets that arrive in real time for various target topics.

References

Balage Filho, P., Avanco, L., Pardo, T., & Volpe Nunes, M. d. (2014). NILC_USP: an improved hybrid system for sentiment analysis in Twitter messages. ACL Special Interest Group on the Lexicon - SIGLEX (p. 0). Dublin: Dublin City University - DCU.

Bifet, A., Holmes, G., Pfahringer, B., & Gavaldà, R. (2011). Detecting Sentiment Change in Twitter Streaming Data. Journal of Machine Learning Research, 5-11.

Bin Wasi, S., Neyaz, R., Bouamor, H., & Mohit, B. (2014). CMUQ@Qatar:Using Rich Lexical Features for Sentiment Analysis on Twitter. Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014) (pp. 186-191). Dublin: Association for Computational Linguistics.

Boag, W., Potash, P., & Rumshisky, A. (2015). TwitterHawk : A Feature Bucket Approach to Sentiment Analysis. Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), (SemEval), 640–646. Retrieved from http://www.aclweb.org/anthology/S15-2107

Buchholz, S. (2006, 06 14). CoNLL-X Shared Task: Multi-lingual Dependency Parsing. Retrieved 04 18, 2015, from CoNLL-X Shared Task: Multi-lingual Dependency Parsing: http://ilk.uvt.nl/conll/#dataformat

Fernandez, J., Gutierrez, Y., G'omez, M. J., & Martinez-Barco, P. (2014). GPLSI: Supervised Sentiment Analysis in Twitter using Skipgrams. Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014) (pp. 294-299). Dublin: Association for Computational Linguistics.

Jansen, B. J., Zhang, M., Sobel, K., & Chowdury, A. (2009). Twitter power: Tweets as electronic word of mouth. Journal of the American Society for Information Science and Technology, 60, 2169-2188.

Jurafsky, D., & Martin, J. H. (2008). Speech and Language Processing (2nd Edition). California: Pearson.

Kiritchenko, S., Zhu, X., & Mohammad, S. (2014). Sentiment Analysis of Short Informal Texts. Journal of Artificial Intelligence Research (JAIR), 723-762.

Kong, L., Schneider, N., Swayamdipta, S., Bhatia, A., Dyer, C., & Smith, A. N. (2014). A Dependency Parser for Tweets. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP) (pp. 1001-1012). Doha: Association for Computational Linguistics.

Kouloumpis, E., Wilson, T., & Moore, J. (2011). Twitter Sentiment Analysis: The Good the Bad and the OMG! Proceedings of the Fifth International Conference on Weblogs and Social Media, Barcelona, Catalonia, Spain, July 17-21, 2011 (pp. 538-541). Catalonia: AAAI Press.

Manning, C., & Jurafsky, D. (2015, 03 28). Sentiment Analysis. Retrieved 03 28, 2015, from Natural Language Processing: http://spark-public.s3.amazonaws.com/nlp/slides/sentiment.pdf

Mohammad, S. M., & Kiritchenko, S. (2015). Using Hashtags to Capture Fine Emotion Categories from Tweets. Computational Intelligence, 31, 301-326.

Mohammad, S. M., Kiritchenko, S., & Zhu, X. (2013). NRC-Canada: Building the State-of-the-Art in Sentiment Analysis of Tweets. Proceedings of the seventh international workshop on Semantic Evaluation Exercises (SemEval-2013). Atlanta: SemEval-2013.

Mohammad, S., & Turney, P. D. (2013). Crowdsourcing a Word-Emotion Association Lexicon. Computational Intelligence, 436-465.

Owoputi, O., O'Connor, B., Dyer, C., Gimpel, K., Schneider, N., & Smith, A. N. (2013). Improved Part-of-Speech Tagging for Online Conversational Text with Word Clusters. Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (pp. 380-390). Atlanta: Association for Computational Linguistics.

Pak, A., & Paroubek, P. (2010). Twitter Based System: Using Twitter for Disambiguating Sentiment Ambiguous Adjectives. Proceedings of the 5th International Workshop on Semantic Evaluation (pp. 436-439). Stroudsburg: Association for Computational Linguistics.

Pang, B., & Lee, L. (2008). Opinion Mining and Sentiment Analysis. Foundations and Trends® in Information Retrieval, 2, 1-135.

Plotnikova, N., Kohl, M., Volkert, K., Lerner, A., Dykes, N., Ermer, H., & Evert, S. (2015). KLUEless : Polarity Classification and Association. Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), 1(SemEval), 619–625. Retrieved from http://www.aclweb.org/anthology/S15-2103

Quinlan, J.R. (1986). Induction of Decision Trees. Machine Learning 1(1). Kluwer Academic Publishers.

Rosenthal, S., Nakov, P., Kiritchenko, S., Mohammad, S., Ritter, A., & Stoyanov, V. (2015). SemEval-2015 Task 10 : Sentiment Analysis in Twitter. Proceedings of the 9th International Workshop on Semantic Evaluation, (SemEval), 451–463. Retrieved from http://www.aclweb.org/anthology/S15-2078

Sebastiani, F. & Esuli, A. (2010). SentiWordNet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining. Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10) (pp. 19-21). Valletta: European Language Resources Association (ELRA).

SemEval 2015. (2015, 01 01). SemEval-2015. Retrieved 03 02, 2015, from Data and Tools: http://alt.qcri.org/semeval2015/task10/index.php?id=data-and-tools
http://alt.qcri.org/semeval2015/task10/

Scherer, K. R. (1984). Emotion as a multicomponent process: A model and some cross-cultural data. Personality & Social Psychology 5, 37-63.

Townsend, R., Tsakalidis, A., Wang, B., Liakata, M., Cristea, A., & Procter, R. (2015). WarwickDCS : From Phrase-Based to Target-Specific Sentiment

Recognition. Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), (SemEval), 657–663. Retrieved from http://www.aclweb.org/anthology/S15-2110

Twitter. (2015, 01 01). About. Retrieved 05 17, 2015, from Twitter: https://about.twitter.com/company

Villena-Roman, J., Garcia-Morera, J., & Gonzalez-Cristobal, C. J. (2014). DAEDALUS at SemEval-2014 Task 9: Comparing Approaches for Sentiment Analysis in Twitter. Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014) (pp. 218-222). Dublin: Association for Computational Linguistics.

Zhang, Z., Wu, G., & Lan, M. (2015). ECNU : Multi-level Sentiment Analysis on Twitter Using Traditional Linguistic Features and Word Embedding Features. Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), (SemEval), 561–567. Retrieved from http://www.aclweb.org/anthology/S15-2094

Zhao, J., Lan, M., & Zhu, T. (2014). ECNU: Expression- and Message-level Sentiment Orientation Classification in Twitter Using Multiple Effective Features. Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014) (pp. 259-264). Dublin: Association for Computational Linguistics and Dublin City University.

Specializations for the Peruvian Professional in Statistics: A Text Mining Approach

Luis Cajachahua Espinoza
UNI, Perú
lcajachahua@gmail.com

Andrea Ruiz Guerrero
UC, Colombia
randreag@gmail.com

Tomás Nieto Agudo
UCLM, España
Tomas.nieto.agudo@gmail.com

Abstract

The objective of this study was to identify the specialization profiles which are most required by companies and organizations in Lima, through the analysis of job postings published in the Internet. Text Mining techniques were used to extract relevant information and to identify some generic skills for the Peruvian statisticians.

For purposes of this study, we analyzed 2,809 job postings published in the Blog "Estadísticos de Perú" [2], between 2009 and 2014. We have identified many requirements, knowledge and specific skills that companies and organizations were looking for. After that, job postings were segmented using Singular Value Decomposition (SVD) of the Terms and Documents Matrix. In addition, five segments were discovered, corresponding to specific competency profiles of statisticians, where each one has different types of knowledge and specific skills.

Keywords: Job postings, Statistician, Professional, Competencies, Abilities. SVD, Clustering, Text Mining.

1 Introduction

The employment trends are changing a lot in recent years. A report published by the social network LinkedIn in 2014, after analyzing 259 million professional profiles, have identified ten professions that did not exist five years ago, but they are very popular today [11, 10]. This produces great uncertainty about the future of young people job opportunities.

On the other hand, there are many careers having accelerated growth in recent years. One of those careers is Statistics. According to reports in several countries around the world, the annual demand for professionals in Statistics has been increasing until having the highest employment rate. One example is Spain, where Statistics is the second career with the lowest unemployment rate in the country [6].

Statisticians are also required in Brazil [8], United States [1] and many other countries. According to another report, made by LinkedIn, statistical skills and data analysis are at the top of the 25 skills most sought by companies in the majority of countries considered in the study [9].

Considering these facts, there are some very interesting questions: What kind of statistics professionals are seeking companies and organizations? Have these requirements changed in recent years? Is there a unique statistician profile, or are several types? Where can we find useful information to clarify these doubts? We tried to answer these questions through analysis of job postings.

2 Background

To understand the demand for professionals and the skills required, we need to find some useful information sources. Previous research related to the issue, were made through in-depth studies, talking with some subject experts [14].

On the other hand, a group of Italian students developed a segmentation technique based on centroids [4] on the database of jobs for college SOUL (University Orientation and Job System, a network that contains jobs posted by 8 different universities in Italy) where they took more than 1,650 job postings. All kinds of them were analyzed, resulting segments from all university careers.

Another related work is the iSchool of Illinois, where they performed a segmentation analysis of Indeed job postings, in order to find the profiles that are most demanded for their students in these subjects [15]. In this case, 15,000 job postings were analyzed, all of them related to professionals in the data analysis field. But, segmentation was performed inside the contents of each job posting, so the resulting segments are referred to generic skills for all professionals.

The two last studies aimed not only to identify the most requested profiles, but also see the status of the current job market and its evolution over time, finding important patterns and can be implemented as actions either within the company or college.

2.1 Objectives

The main objectives of this study are:

- Identify the more important requirements, competencies and demands that companies include in their job postings.
- Detect the existence of professional profiles through all the job postings available through text mining techniques.
- Compare the evolution of the requirements and skills by dividing the dataset in two periods (2009-2011 and 2012-2014).

Once all previous goals achieved, we can make some recommendations to the agents involved in the job market: companies, educational institutions and potential employees, statisticians.

2.2 Limitations

By the nature of the study, it should be noted limitations implied in its realization:

- The main information source is the Blog where the job postings are published. If there were errors or omissions in the posts, they will influence the accuracy of the results.
- There are job opportunities that are not being published, causing a bias in the analysis results. Moreover, many leadership and senior positions are sent to headhunting companies. Consequently, they could not be included in this analysis.
- The postings are mostly from companies and organizations located in the city of Lima. Peru is still a very centralized country, nearly a third of Peruvian population lives in Lima, so the results could not be extrapolated to the whole country.

3 Methodology

According to the literature reviewed, there are several methods of text analysis, but these methods work well in other languages, so we needed to adapt some tools to Spanish. On the other hand, our aim, unlike previous studies, is to segment the job postings, in order to know the different types of specialties for a statistician.

3.1 Study scope

The population considered was formed by 2,809 job postings published in the blog "Estadísticos de Perú" [2]. All the postings were analyzed, so it was unnecessary to use sampling techniques. The number of postings published per year is shown in the next graph.

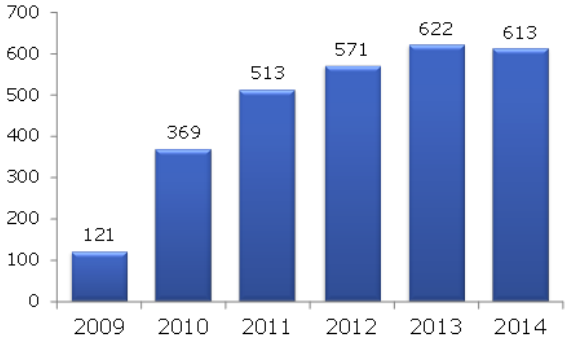


Fig. 1. Job postings per year

3.2 Text Mining

As a part of Data Mining, Text Mining is the intensive process of information extraction, where a user interacts with a collection of documents using specialized analysis tools. As a process, it deals with the discovery of knowledge in the content of several texts and after passing through several stages.

Text Mining seeks to extract useful information from multiple data sources through the identification and exploration of interesting patterns. One remarkable difference with numeric data analysis is that the documents analyzed do not have a defined structure. That is why in text mining the pre-processing tasks are very important. These operations are focused on the features identification and extraction of natural language and are responsible for transforming unstructured data in a structured intermediate format.

Text mining is used for:

- **Classify and organize documents based on their content:** With the information overload in companies, it is necessary a

method to facilitate the classification of documents that enter daily to the system. Text mining has several algorithms to do this automatically using index classification.

- **Organize depots for search and retrieval:** This problem spots the need of an efficient system search, through the submission of a request for recovering specific information. This query sends keywords to help identify the documents that best fit, sorts by relevance and the best matches are displayed. There are techniques that help to measure the similarity between documents in order to calculate the similarities and return information.
- **Automated addition and comparison of information:** Many times, when researchers have many documents on the same subject, it is necessary to group the information automatically to facilitate analysis. Text clustering is a useful technique to build the groups in these cases.
- **Extract relevant information from a document:** Text mining has methods that deals with unstructured texts, analyzes them and identifies groups of concepts. That is, it transforms plain texts into valuable and relevant knowledge.
- **Prediction and evaluation:** One of the concerns expressed sophisticated text mining is to create predictive models and evaluation from textual information that you count. These models are based on a model already raised issues of modeling and assembly, to predict for new documents entering the collection items or more suitable groups according to their contents. This type of problem is one of the most common text mining.

3.3 Text Mining Elements

Text Mining, as many other disciplines, have some recognizable elements that characterize it.

- **Repository of documents:** Any set of documents containing text, regardless of size, can be 10 or 100 billion texts. One of the main sources of documents, with more than 12 million items open to the public, with a wide variety of subjects and in different languages is PubMed. These characteristics have become one of the databases most used by computer professionals in data analysts or interested in the implementation of text mining tasks on a large scale. This collection is dynamic and are added over 40,000 items biomedical each month [17]. In a collection of this size, try to correlate the data between documents, mapping relationships or identify trends, could be extremely complex and demanding, in terms of time and machine. But there are some techniques that perform these tasks automatically that improve the speed and efficiency in the analysis.
- **Document:** For practical purposes, a document is a unit of text data (e.g. news, a report of business, emails, research articles, manuscript, stories, tweets, books, among others).
- **Corpus:** A collection of documents, usually stored electronically and on which the analysis is performed. Its elements are known as documents which store the current text and the local metadata.
- **Terms and documents matrix:** It is the most common way to represent text for future comparisons. This matrix is composed of document ID's as rows and terms as columns. Its elements are the frequencies of each term within that document.
- **Vector space model:** It is a matrix whose coefficients are functions of term frequency.

3.4 Text Mining Tools

On this study, we used R libraries and SAS Text Miner in order to obtain the results, because each one offers some advantages and useful tasks that the other one doesn't have. Another reason to choose these platforms is that the other ones do not have text Stemming and Lemmatization tools in Spanish. We can see a comparison of these tools in the next diagram:

Tasks	R*	SAS Text Miner
Ortographic Correction	✔	✔
Text Filtering	✔	✔
Multi-Words		✔
Lemmatization	✔	✔
Stemming	✔	
Term Matrix	✔	
SVD Decomposition	✔	✔
Text Segmentation		✔
Word Clouds	✔	

Fig. 2. Comparison of R and SAS Text Miner Tasks

Following this comparison, we decided to use both packages. R to clean the data and generate Word clouds for the segments and SAS Text Miner to the SVD decomposition and Segmentation.

The scheme of the Text Mining process is shown in the following image:

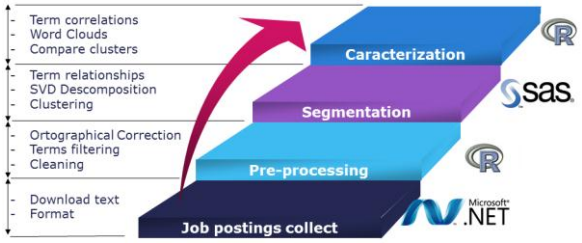


Fig. 3. Tasks and tools used in the analysis

In the terms filtering step, some stopwords were used, in order to avoid some obvious findings, like statistics, statistician, job, salary, enterprise, etc. ("estadística", "estadístico", "empleo", "salario", "empresa", etc.) Then, we performed the SVD decomposition and finally, the text clustering step. After this process, we obtained some interesting findings, which are explained in the next section.

4 Results

After textual analysis, we can answer the research questions. For example: What are the requirements and skills that students and professionals in Statistics are requested on employment notices published?
For the first answer, we could see the Word cloud of the complete database in order to discover the main requirements founded.

Fig. 4. WordCloud considering the entire Corpus (Total: 2,809 job postings)

As observed, the most prevalent and relevant terms in the job appear larger. That is, in a high percentage of postings, these words appeared which leads us to believe that one of the first things required of a statistic is the experience ("Experiencia"). We can see other some basic and generic skills, data analysis and information management ("datos", "análisis", "información" y "manejo"). Then, some other words make references to specific skills, such as SPSS or Excel. So, it is necessary to use clustering techniques, since there are several groups of words representing different capabilities related to statistical profiles.

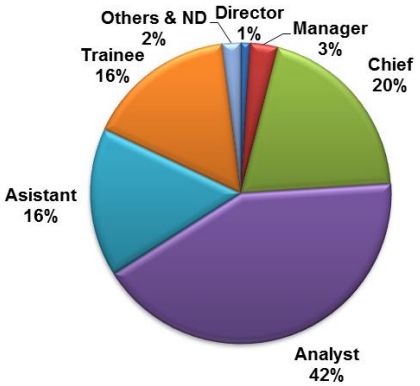


Fig. 5. Distribution of job postings for level (Total: 2,809 job postings)

It's clear that analysts' position dominates, because as we said, the job postings correspond to basic or intermediate positions.

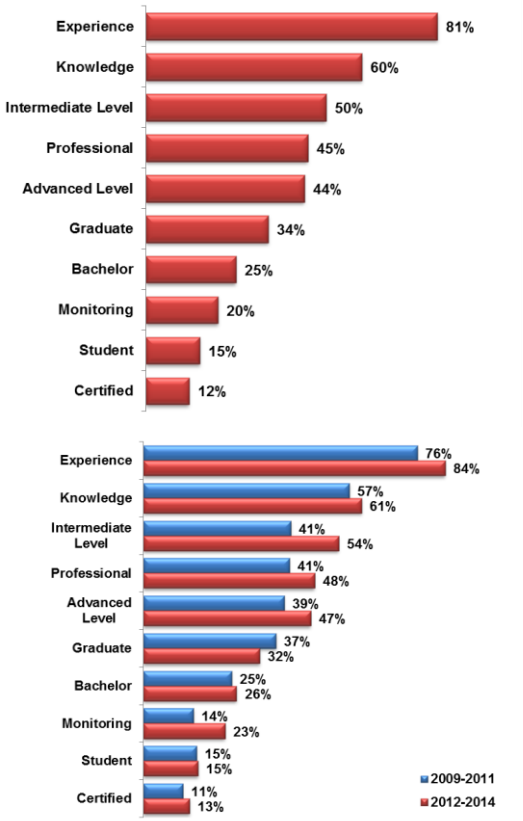


Fig. 6. Job postings distribution by Requirements and period

It is remarkable that 81% of job postings mention the word “Experience” in them. It means that this is one of the most important requirements (along with knowledge or intermediate and advanced levels). Furthermore, they have experienced increasing importance in recent years.

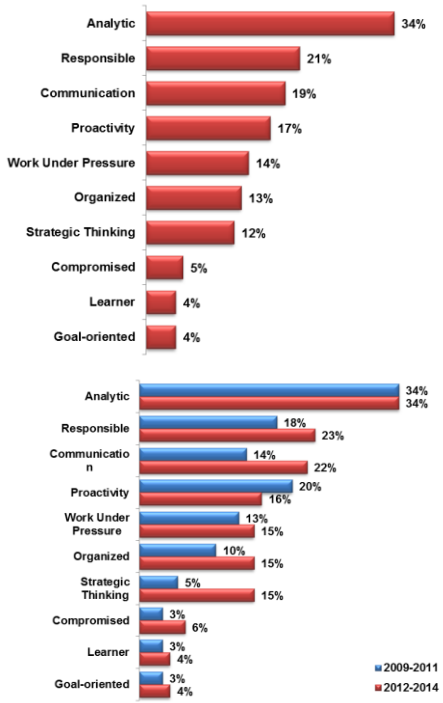


Fig. 7. Job postings distribution by Competencies and period

As for the Competencies, we highlight the character or analytical profile along with other basic skills in business such as responsibility and communication skills. The increase of good communication, responsibility and strategic thinking is valuable. Clearly, the organizations seek Statisticians that are not only good at technical level, but also have the ability to think about the best solution for the organization as a whole.

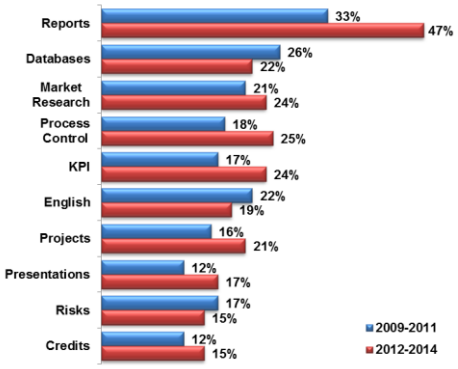


Fig. 8. Job postings distribution by required background and period

About the background required, it weighs heavily reporting tasks or report writing (24%). One in four job postings, contains the term "database" which makes clear that the SQL language has become very important in Lima. Not just someone who can get statistics or models is needed, organizations valued professionals whose can extract themselves from the data sources. Other tasks are in high demand as Process Control or Indicators Development.

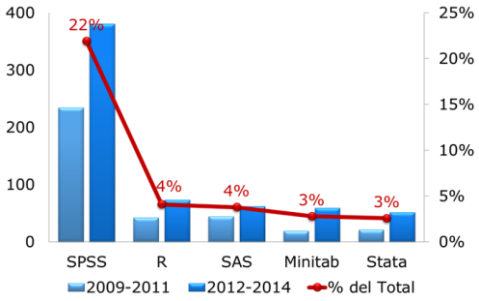


Fig. 9. Most required Statistical Software

The importance of SPSS in the area of Lima is also clear growth in recent years (almost doubling its appearance in the ads). Others such as R or SAS are still not much required; maybe because the cost of acquisition or the time required learning the software (SPSS is easier).

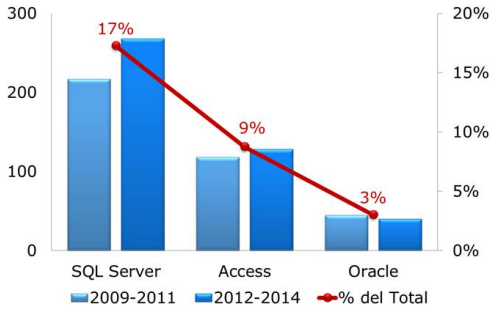


Fig. 10. Most required Database Management Software

Regarding the database software, SQL Server predominates over Access or Oracle.

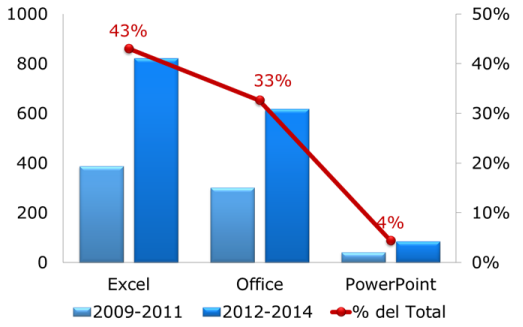


Fig. 11. Most required Office Software

Notice the importance of Excel (appears in four out of ten job postings) and its great increase in recent years.

Finally, it is important to determine the existence of specialization profiles, segments that meet specific characteristics and are different from others. For this, we use SAS Enterprise Miner to compare the results from four, five and nine segments, we decided to choose five segments because it showed better indicators of distance between clusters and better possibilities of interpretation. The distribution of each segment is shown in the next figure:

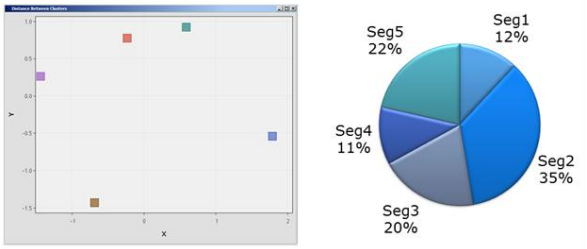


Fig. 12. Term-based Segmentation in SAS Enterprise Miner

After segmenting the messages in these groups, we decided to perform a characterization, that is, find the most common expressions in each cluster, in order to get a better idea of the composition of each segment:

	Seg1 Risk Management	Seg2 Reporting	Seg3 Business Intelligence	Seg4 Trainee	Seg5 Market Research
Knowledge.1	Risk Management	Data Analysis	Databases	Investigate	Investigate
Knowledge. 2	Stocks Management	Comercial	Marketing	Engineering	Techniques Cuantitativ
Profile 1	Analysts	Marketing	Campaign Management	Proactive	Exp. Surveys
Profile 2	Engineers	Data Analysts	Experience in B.I	Graduate	Exp. Marketing
Software 1	SQL	Excel	Excel	SPSS	SPSS
Software 2	SPSS	Office	SQL	Excel	Excel

Fig. 13. Segments Characterization

Through descriptive terms offered by the five clusters finally formed and considering the results of characterization through WordClouds. The following professional profiles were obtained:

Risk managers (Cluster 1): Professionals with experience in portfolio and risk management (both credits and investments), preferably analysts and engineers. They are sued for the financial and banking sector. They were also requested domain mainly SQL and SPSS.

Fig. 14. Word cloud of the Cluster 1

Analysts with reporting tasks (Cluster 2): Analysts with good statistical knowledge required for tasks of reporting and report writing. Mainly related to the areas of marketing and sales. The most required software is the Office suite, more specifically Excel.

Fig. 15. Word cloud of the Cluster 2

Business Intelligence Professionals (Cluster 3): Profiles that manage and analyze databases generally related to marketing and related areas (customers, sales, campaigns). They were also asked experience in campaign management and business intelligence. In software they are required Excel and SQL.

Fig. 16. Word cloud of the Cluster 3

Students or graduates in trainee programs (Cluster 4): Young graduates who are at the end of its cycle of studies (generally engineering) with knowledge of analysis tools and required to be proactive. They are required to dominate Excel and SPSS.

Fig. 17. Word cloud of the Cluster 4

Market researchers (Cluster 5): Professionals in the field of market research (both quantitative and qualitative analysis). They were also required experience in processing and analysis of surveys and marketing knowledge (for research applications). They are required Excel and SPSS too.

Fig. 18. Word cloud of the Cluster 5

These are the profiles we wanted to find, as we have seen, each implies that the professional should have sought some proper statistics to job in question features.

5 Conclusions

According to the results, we can conclude that Statisticians have relative success in Lima. In addition, we have obtained the following conclusions:

1. The main goal (to identify key competencies and requirements) has been successfully achieved. It was possible to detect the main (technical and personal) requirements that often companies require in their job requirements. And due to the temporary separation into two periods, we also found interesting differences about the change in the demand of these requirements.

2. The second one (identification of professional profiles), has also been achieved. We have identified five types of professionals, each group are different from the rest and we have characterized them accurately and in a very clear way.

3. The results obtained in this analysis, may be useful for three agents who are involved in the labor market: companies, potential workers (statisticians) and educational institutions:

- Business: Companies can improve their job postings, making easy the contact with the wanted profiles. In the other hand, they could obtain certain advantages in areas such as employee training, based on the specific profiles founded.

- Statisticians: This analysis would be helpful for them, in order to improve the CV writing, increasing their chances to obtain a good employment opportunity. They can also focus their training in the same direction as do the requirements of companies.

- Education: Universities, training centers and other institutions can adjust their academic offer, in order to meet the needs of the market.

References

[1] AMSTAT (2015). "Statistics is the fastest-growing undergraduate degree". [Consulted: February 3, 2015]. Available in: http://bit.ly/1uvCn4F

[2] Cajachahua, L. (2008). "Estadísticos de Perú". Blog de empleo y prácticas. [Consulted: February 15, 2015]. Available in: http://bit.ly/1FZVfuV

[3] Cox, A., and Corral, S. (2013). "Evolving Academic Library Specialties". Journal of the American Society for Information Science and Technology. 64 (8): 1526-1542.

[4] Domenica, F., Mastrangelo, M., and Sarlo, S. (2012). "Text Clustering Based on Centrality Measures: An Application on Job Advertisements". [Consulted: June 1, 2015]. Available in: http://bit.ly/1HO6uVv

[5] ElPais.com (2014). "Las carreras con mayor tasa de empleo". [Accessed: October 29, 2014]. Available in: http://bit.ly/1rSot5P

[6] ElPais.com (2015). "¿Cuáles son los estudios con menos paro? ¿Y los que más tienen?" [Consulted: May 7, 2015]. Available in: http://bit.ly/1Jt25K3

[7] Han, J., and Kamber, M. (2001). Data Mining: Concepts and Techniques. Morgan Kaufmann.

[8] IPEA (2014). Radar: Technology, produção and Foreign Trade (2013) 27 Institute of Applied Economic Research. Setoriais Diretoria of Studies and Policies, of Inovação, Regulação and Infrastructure. [Consulted: June 1, 2015]. Available in: http://bit.ly/1SZHL9j

[9] LinkedIn (2014). "The 25 Hottest People Skills That Got Hired in 2014". [Consulted: December 17, 2015]. Available in: http://linkd.in/1x0LQBT

[10] LinkedIn (2014). "Top 10 Job Titles That Did not Exist 5 Years Ago". [Consulted: June 1, 2015]. Available in: http://linkd.in/KtpUbI

[11] Merca20.com (2014). " Infografía: 10 populares empleos que no existían hace 5 años". [Consulted: June 1, 2015]. Available in: http://bit.ly/1abEw6c

[12] Parr Rud, O. (2001). "Data Mining Cookbook". John Wiley & Sons, New York, NY.

[13] RPP.com (2015). "Conoce cuáles serán los empleos más demandados en los próximos 10 años". [Consulted: March 4, 2015]. Available in: http://bit.ly/1EYJH7k

[14] Swan, A., and Brown, S. (2008). "The Skills, Role and Career Structure of Data Scientists and Curators: An Assessment on Current Practices and Future Needs". Report to the JISC.

[15] Thompson, Cheryl A., and Craig Willies. (2015). "Data Workforce Needs: Disambiguation of Roles Using Clustering and Topic Modeling". [Consulted: June 1, 2015]. Available in: http://bit.ly/1QaPDpu

[16] Witten, IH, Frank, E., and Hall, MA (2011). "Data mining: Practical machine learning tools and techniques". San Francisco: Morgan Kaufmann. 3rd edition.

[17] National Institutes for Health (2015). PubMed: US National Library of Medicine. [Consulted: June 1, 2015]. Available in: http://1.usa.gov/1brVEaa

Social Networks of Teachers in Twitter

Hernán Gil Ramírez
College of Education
Carrera 27 #10-02
Barrio Alamos
Pereira, Risaralda (Colombia)
ZIP code 660003
hegil@utp.edu.co

Rosa María Guilleumas García
College of Humanities and Fine Arts
Carrera 27 #10-02
Barrio Alamos
Pereira, Risaralda (Colombia)
ZIP code 660003
roguiga@utp.edu.co

Abstract

This research aimed at identifying the trends in the topics of interest of the tweets published by 43 expert professors in the field of ICT and education and the network of their followers and followed in Tweeter, as well as their relationship with the characteristics of that network. With this purpose, NodeXL was employed to import, directly and automatically, 185.517 tweets which gave origin to a network of connections of 49.229 nodes. Data analysis involved social network analysis, text extraction and text mining using NodeXL, Excel and T-Lab. The research hypothesis was that there is a direct correlation between the trends identified in the topics of interest and the characteristics of the network of connections that emerge from the imported tweets.Among the conclusions of the study we can highlight that most of the trends identified from the analyzed tweets were related to education and educational technologies that could enhance teaching and learning processes; the association between education and technologies found through the text mining procedure applied to the tweets; and finally that the analysis of lemmas seems to be more promising than that of hashtags for detecting trends in the tweets.

1 Introduction

Currently, social networks in digital spaces are an important part of the life of a good number of people and institutions. Nevertheless, their study poses important challenges for researchers, since the huge volume of data circulating through them implies -for collection, processing, and analysis-, the use of specialized software, powerful equipment, complex analysis methods, and qualified people, items that are not always available in the small and middle-size educational institutions.

Though many users exchange through Twitter what Ferriter (2010) calls "digital noise," this researcher claims that professionals in education have found ways to use Twitter to share resources and provide a quick support to colleagues with similar interests, turning this service into a valuable source of ideas to explore.

Twitter may be used for communication purposes, but also to share information and build, collectively, academic communities. This social network enables interaction with other people, access to their interests and identification of trends from the published messages.

2 Research background

This work takes as referents some previous research on Twitter and the generation, exchange and propagation of information; it also considers works about the influence of users on this digital space. Shneiderman (2011) explores the reasons for the success of social media like Facebook, Twitter, YouTube, Blogs, and the traditional discussion groups and he concludes that it is due to the fact that they allow people to participate actively in local and global communities; the role of Twitter as a communication resource and information exchange tool during a crisis is tackled in Herverin and Lisl (2010) research, and also in Chew and Eysenback's (2010) work.

Weng, Lim, Jiang and He (2010) focus on the issue of the identification of the influential users of Twitter; Bakshy, Hofman, Mason and Watts (2011) study the features and relative influence of Twitter's users. Regarding the propagation of information, our referents are Lerman and Ghosh (2010), as well as the research carried out by Gómez, Leskovec, and Krause (2010), where they state that the diffusion of information, and viral propagation are fundamental processes in the networks; we finally highlight the work done

by Wu, Hofman, Mason and Watts (2011), where they stress the importance of understanding the channels through which information flows, in order to comprehend how it is transmitted.

3 Theoretical considerations

Castells (2011) thinks that the Internet is revolutionizing communication thanks to its horizontality, feature which permits users to create their own communication network and to express whatever they want, from citizen to citizen, generating a capacity of massive communication, not mediated by the traditional mass communication media. This communication networks are the basis of the "network society," a concept which was popularized by this author, who describes it as the social structure that characterizes the society of the early 21st century, a social structure constructed around (but not determined by) digital communication networks. (Castells, 2009, p.24). It is in the space and the time of the network society where the studied group of teachers constructs communication networks using Twitter, making out of it more than just a simple technology, a tool for communication, encounter, and assistance.

Castells defines a network as a set of interconnected nodes. The nodes may have more or less relevance for the network as a whole, so those of higher importance are called "centers" in some versions of the network theory. At any rate, any component of a network (including the "centers") is a node, and its function and meaning depend on the network programs and on its interaction with other nodes in it. (2009, p.45) This author explains that the importance of the nodes in a network is higher or lower depending on how much important information they absorb and process efficiently, that is, it is determined by their capacity to contribute to the effectiveness of the network in the achievement of its programmed objectives (values and interests).

In this sense, we approach the study of the communication networks created by teachers from connections they establish in Twitter. In this case, each user, and each web domain, hashtag, lemma, constitutes a node which establishes connections in the network under study, where it is evidenced that there are nodes with higher relevance than others. This is precisely what contributes to the understanding of the dynamics of these networks: what nodes are more important in the network, which are their contributions, and in what way they make up the structures of these relationships.

Social networks, as posed by Lévy (2004), provide tools for human groups to join mental efforts so as to constitute intellects or collective imaginaries. This allows for connecting informatics to be part of a technical infrastructure of the collective brain of lively communities, which profit from social and cognitive individual potentialities for their mutual development. Lévy (2004) describes collective intelligence as "una inteligencia repartida en todas partes, valorizada constantemente, coordinada en tiempo real, que conduce a una movilización efectiva de las competencias…" y agrega que "…el fundamento y el objetivo de la inteligencia colectiva es el reconocimiento y el enriquecimiento mutuo de las personas (…)".

Concerning this point, we can sustain that networks like Twitter create the suitable space to integrate the intelligence of many people, located in different places around the world; an intelligence that is permanently updating, allowing people linked to the network to widen their horizons and possibilities to access information. Our intent in this research is, following Lévy`s pathway, to appraise the potential of Twitter as a space for interaction in the network of the teachers under study, and also to value the information they exchange and which can be accessed through this means, as a manifestation of collective intelligence.

4 Methodology

This research followed a quantitative approach with a trans-sectional, correlational, non-experimental design, which allowed for the establishment of the relation between the trends in the topics of interest detected and the structure of the network of connections that emerged from the tweets published by the selected group.

In order to select the group to be studied, we adapted the snowball sampling method. An initial group of seven (7) professors was intentionally identified and selected on the basis of their academic background related to the use of the ICTs in education, and their academic contributions via the Internet, in particular through Twitter. In a second phase, there was a follow-up of these seven professors' Twitter accounts, in order to identify other teachers who followed them or that they followed, and who, on the basis of their contributions in Twitter, their publications and academic output about the use of ICT in ed-

ucation, could be part of the studied group. This procedure was repeated once again until finally it was formed, in a not probabilistic way, a group of 43 teachers.

Of the selected group, 65% were University professors, 23% primary and secondary teachers and 12% belonged to other type of institutions (non-formal, virtual tutors and advisors). Concerning their nationalities, 84% were from Spain, 7% from Argentina, 5% from Colombia, 2% from Mexico and 2% from Venezuela.

Using NodeXL we imported from Twitter, 185.517 tweets published by the network of connections of the 43 selected teachers between February the 4th and June the 6th, 2014.

As data collection instruments, we used NodeXL templates (which include not only the tweets but also the information of the edges, as well as that of the nodes). From the imported data rose a network of connections made up 49229 nodes and 98.494 edges.

These nodes were located in 128 countries. 88.3% of them were concentrated in 10 countries, among them, Spain, Argentina, The United States, Colombia, and Mexico. About one third of the nodes registered in their profile professions related with education.

In order to identify the trends in the topics of interest in the published tweets and their relationship with the features of the network from which they emerged, we made a graphic representation of the network and calculated its metrics, using NodeXL. Likewise, we identified the trends in the topics of interest by analyzing the imported tweets to quantify the frequencies of appearance of the hashtags and by applying text mining to the content of the tweets. We also identified the trends in the web domains and established the correlation among the frequencies of the topics of interest detected as trends and the metrics of the network, using multivariate analysis, and Pearson's correlation coefficient. For data analysis we used the programs NodeXL, Excel, T-Lab and Statgraphics.

5 Analysis and data interpretation

For data analysis and interpretation**,** we examined the features of the network of connections of the 43 teachers selected. Besides, based on the tweets published by the mentioned network, we identified the trends in the topics of interest and studied their correlation with the values obtained in the two previous steps.

5.1 Features of the communication network

We used NodeXL to make the graph of the network of connections as well as to calculate its metrics.

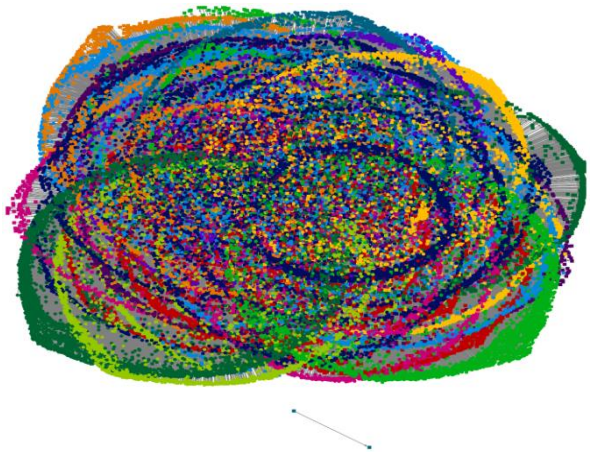


Figure 1. Communication network emerging from the imported tweets.

Taking a look at figure 1 with its 49.229 nodes and 98.494 edges, it is evident that, given their location, not all the nodes have the same importance in the network. A representative group of nodes, located in the center, are the most connected; a significant amount, the least connected, are displaced outwards, and a couple of them, though connected to each other, are disconnected from the network.

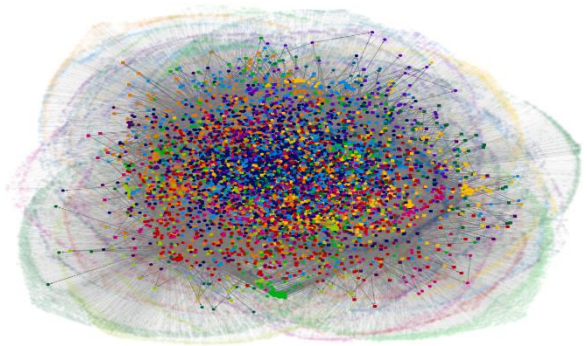


Figure 2. Communication network emerging from the imported tweets, filtered.

Figure 2 corresponds to the same network after the application of a filter based on the Betweenness Centrality index of the nodes and it shows only those with a value higher than 1 for that index. This produced a reduction of the network to 8.725 nodes (a 17.7% of the total). This process allowed us to note, more clearly, the set of nodes that occupied the center, while in the periphery, in opaque tones, there can be seen the remaining nodes, those out of the established filter.

Thus, we can see the configuration of a network, that as Castells sustains (2009:45), is made up of interconnected nodes; some, the so-called centers, of greater importance for the network, and others, less important, depending on their capacity to access information and process it efficiently; that is, on account of their capacity to contribute to the achievement of the objectives of the network itself.

The process of analysis implied, likewise, the calculation of the graph's metrics, as a basis for the quantitative measurement of the indices associated to the nodes and their edges. The graph is directed. The relation of reciprocity of the edges is of 0.27. The In-Degree ranges between 0 and 3.439, the Out-Degree between 0 and 1.789 and the Betweenness Centrality index between 0.0 and 354805308.32.

Of the 49.229 nodes analyzed, the 10 nodes with a higher In-Degree, Out-Degree, and Betweenness Centrality, belonged to the initial group of 43 teachers selected. This shows that, in addition to a relative high level of edges between the nodes of the network, the initial group of 43 teachers selected, from which the network of connections emerged, had a significant weight within the network, both for the amount of nodes connected to them as for the amount of nodes to which they were connected and therefore for their intermediation potential in the network. This is particularly important in a scenario where just a few nodes had high degrees of intermediation.

The 49.229 nodes of the network were organized in 24 groups of diverse sizes, according to the number of nodes in them. There was a high amount of edges inside each group, as well as among the different groups. For instance, group 1 had 8.220 nodes (16.7% of the total) and 174.397 edges. At the other end in size and edges were group 23 (with 525 nodes, 1.1% of the total and 586 connections) and group 24 (disconnected from the network, with just 2 nodes and a single edge between them).

Regarding the making up of the groups, we want to state that within a network of connections it is difficult to establish groups as well as their precise borders since the nodes can be involved in different relations and belong to more than one group.

In this research, the clusters were conformed with NodeXL, using the Clauset-Newman-Moore algorithm for clusters, that automatically identifies the groups from the network structure, placing the densely connected nodes in separated groups; that is, conforming each group with a set of nodes that are more connected to one another than what they are to other nodes.

On average, each of the 24 groups had 2.051 nodes, 2.939 inner edges and was connected to 21 of the 24 existing groups through 1.164 edges, what shows a highly connected network. In this respect, it is worth noting the existence of groups that were rather highly connected to other groups, as for example, group 1 with 5.678 edges, and group 2 with 3.076.

We believe that the communication that exists among the nodes, inside the conformed groups and among them, facilitates the access to information and its distribution inside the studied network, thanks to what Castells (2001) calls the process of horizontality, which allows all the nodes connected to the network to communicate massively, to share whatever they wish and thus build their communication networks, in this case through the use of Twitter.

As a summary, we can affirm that the network studied was decentralized, though not in the classic sense of the term since some nodes were connected to one or more central nodes, which in turn were often connected to several nodes, central or not, making the structure of this network more complex and robust, in such a way that if one of the central nodes were to disappear, this would not cause the disconnection of a great amount of nodes or the disappearance of the network.

The study of the tweets exchanged in the studied network showed that, within it, the identified trends (hashtags, lemmas and web domains) were the origin of other networks.

5.2 Identification of tendencies of the topics of interest to be published

The web domains referenced in the tweets, as well as the hashtags and slogans more used, led to the identification of the trends in the topics of interest to be published in the studied network.

Tendencies identified from the hashtags referenced in the tweets.

Of the 185.517 imported tweets, 31, 5% (58.349) included hashtags. The total of referenced hashtags was 88.798, out of which 29.590 were unique hashtags. We identified the hasthtags referenced in the tweets and calculated their frequency of appearance. The 10 hashtags with a higher referencing frequency (0.03% of the total) were used 6% of the times, while the remaining 29.590 (99.97%) appeared the 94% of the times.

The first place was for the hashtag *#educación*, followed by #ABPmooc_intef and #elearning, #tic, *#edtech*, *#educacion*, *#eduPLEmooc*, among others.

The ten hashtags with a higher frequency of use in the tweets could be grouped around three main topics: education (8 hashtags), politics (1 hashtag), and technology (1 hashtag). The predominance of the hashtags related to the topic of education could seem obvious in a network initially composed by teachers; however, we should remember that the 43 initially selected teachers were the seed of a network that was enlarged to include 49.299 nodes; this suggests that the 43 teachers followed and were followed either mainly by teachers, or by people interested in and concerned about education.

This piece of data may show some degree of homophily in the studied network of connections, since despite the fact that Twitter users are not forced to correspond to their followers (directed network) and most of the links are not corresponded, the users tend, however, to connect to others exhibiting interests and activities similar to their own (Kwan, Lee, Park, and Moon, 2010). This situation also matches Wu, Hofman, Mason and Watts's findings (2011), who highlight the significant homophily found in their research.

Network of tendencies identified from the 10 most referenced hashtags

The tendencies identified from the 10 most referenced hashtags enabled the conformation of a network of connections between the nodes referencing the hashtags (source node) and the hashtags which were being referenced (target node).

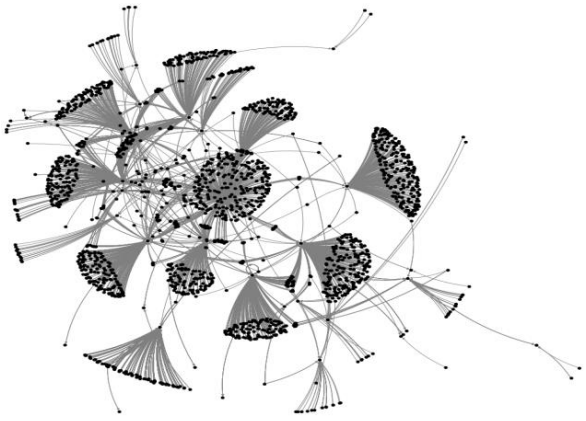


Figure 3. Network of connections of the 10 most referenced hashtags

Figure 3 illustrates how most of the connections were grouped around a specific hashtags. There are very few cases in which a node used more than one hashtag. However, as an example of this case, we can mention *#eduPLEmooc y #ABP-mooc_intef*, which set up some connections with the same users.

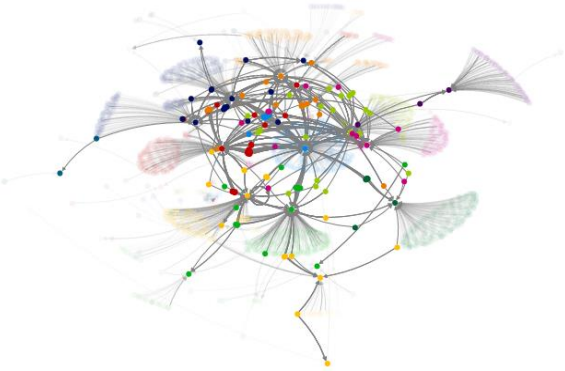


Figure 4. Network of connections of the 10 most referenced hashtags, filtered.

Figure 4 was the result of the application of a filter based on the Betweenness Centrality index of the nodes. It shows the 154 nodes (8.2% of the total) with a higher than the average value of this index. This process allowed the visualization of those nodes with greater force of intermediation in the network, located in the central part of the graphic. It also let us observe that most of them, about 91.8% have a low or no force of intermediation at all. These nodes, represented with opaque tones, were located in the periphery of the graphic according to the decreasing value of the index, a value that reached 0 for 1.533 nodes, that is, for the 81.3%.

As we can observe from these metrics, there was an important number of nodes which could be considered as "lurkers" since they do not contribute much to the network; they are mainly silent participants.

The In-Degree index in this network ranged between 0 and 335, the Out-Degree between 0 and a 6; and the Betweenness Centrality between 0 and 1.453.757,65. Although a hashtag can receive many entries (as in the case of #educación, with an In-Degree of 335, or #ElReyAbdica, with 237), these are generated by many nodes. We can then assert that the tendencies detected are actually a product of the individual contributions of an important number of network nodes, what evidences the materialization of Lévy's collective intelligence.

Within the network of connections of the 10 hashtags with a higher frequency of use in the

tweets, 21 groups were conformed. On average, each group connected only to 2 other groups, and there were even some groups that were not connected to any. It is remarkable that the groups with a larger number of nodes connected to a greater amount of groups. One example of this is Group 1, which having 271 nodes, was connected to 5 groups. In contrast, the groups with a lower number of nodes showed the tendency of not setting up connections to any group. This was the case of group 21, which having 2 nodes, did not connect to any group.

Trends identified in the lemmas of the tweets

In order to advance in the identification of the topics of interest in the tweets published by the network of connections of the group of selected teachers, we resorted to text mining. The analysis of the content of the tweets was done with T-Lab, using the automatic lemmatization (word grouping) and the selection of key words.

Starting from the 185.517 imported tweets, the corpus of analyzed data was made up of 175.122 elementary contexts (EC), 179.374 words, 162.072 themes, and 2.574.255 occurrences. The program automatically selected the 500 words with the higher level of occurrence in the corpus, out of which the non-meaningful terms were manually deleted later (articles, preposition, etc.) giving a remainder of 310 items. For text segmentation (elementary contexts), we used the paragraph, which in this case was equivalent to a tweet. For the selection of key words we employed the method of occurrences.

Nº	Lemmas	EC
1	Educación	4.024
2	Nuevo	2.543
3	Educativo	2.415
4	Social	2.404
5	Aprender	2303
6	Curso	2.238
7	Seguir	2.201
8	Blog	2.143
9	Stories	2.117
10	Vida	2.063

Table 1. Lemmas and Elementary Contexts (EC)

Lemmas associated with education, such as *educación, educativo, aprendizaje* or *curso* stood out in frequency of citation in the tweets as shown in Table 1. The lemma *educación* had already been identified as one of the 10 most referenced hashtags.

Analysis of co-occurrences/word associations

The co-occurrence is the number of times (frequency) that a lexical unity (LU) appears in the corpus or within the elementary contexts (EC), in this case in the tweets. The function *word association* was used to detect which words, in the elementary contexts, were the co-occurrences with the lemma *educación*.

The lemma *Education*, found in 4.024 of the 175.122 elementary contexts (EC) analyzed, was associated to a group of lemmas, considered as relatively close, among them *tic, technology*. Their relation with the lemma *educación* is confirmed by the higher values of the index of association presented in Table 2. *Tic, 0.166:* technology, 0.166. The closer the association between two lemmas, the higher the coefficient.

Table 2 presents data of the relationships between occurrences and co-occurrences of the lemma *educacion* in the elementary contexts.

LEMMA (B)	COEFF	E.C. (B)	E.C. (AB)
tic	0,166	1577	419
tecnología	0,166	1285	377
congreso	0,109	744	189
básico	0,107	396	135
innovación	0,1	634	160
ciencia	0,082	641	131
infantil	0,076	653	124
futuro	0,065	804	117
Blog	0,065	2143	191

Table 2. Lemma Educación (Theme A)[1]. Partial List

In addition, the lemma *tic* appeared in 1.577 elementary contexts, and the lemmas *educación* and *tic* were referenced together in 419 elementary contexts. As we can observe in Tables 1 and 2, there was evidence of the prevalence of lemmas associated with education, as well as of the close association between them, in the elementary contexts analyzed.

[1] Conventions: LEMMA A= Educación; LEMMA B = Lemmas associated with LEMMA (A); COEFF = Value of the index of association selected; E.C. (AB) = Total of EC in which the lemmas "A" and "B" are associated (co-occurrences).

Tendencies of the web domains identified in the tweets

Out of the 185.517 imported tweets, 59,4% included references to web domains. Using Excel, 113.361 domains were identified, out of which 18.448 were unique web domains. In order to detect the tendencies in the domains, we calculated their frequency of reference and located the 10 with the highest levels of reference. It is worth noting the great amount of references accumulated by these 10 domains, since making up just for a 0.05% of the amount of unique domains found in the tweets, they were referenced in the 25,4% of the occasions.

Among the 10 most cited web domains were blog sites (blogspot, 1st position), sites for video publishing (Youtube, 2nd position); social networks (Facebook, 3rd position; Instagram, 6th position; LinkedIn, 7th position; Foursquare, 9th position); online newspapers and journals (Paper.li, 5th position; eldiario.es, 10th position); content curation sites (Scoop.it, 4th position).

It should be highlighted that most of the referenced domains (4 out of 10) were social network applications. Likewise, we must point out the importance of the blogs for the studied network, since besides the tweets that included mentions to blogs of blogspot, there was also a considerable amount of domains making reference to other blogs, like in the case of blogs.elpais, blog.educalab, blog.tiching, blog.fernandotrujullo and blogthinkbig.

This listing of web domains in general and blogs in particular, permits the visualization of tendencies in the use of the web, and may help teachers approach the best possibilities to explore them and integrate them in their teaching practices.

Network of the tendencies of the web domains identified in the tweets.

The 10 domains more cited in the tweets allowed shaping a network of connections between these 10 web domains (target nodes) and the users referencing them (source node). This new network was made up of 10.900 nodes and 28.745 connections (7.319 unique connections and 21.426 duplicated connections).

To facilitate the analysis and interpretation of the graph, we applied a filter based on the Betweenness Centrality Index of the nodes, allowing the visualization of the nodes with a higher power of intermediation in the network, and therefore, with a greater significance in so far as the flow of information.

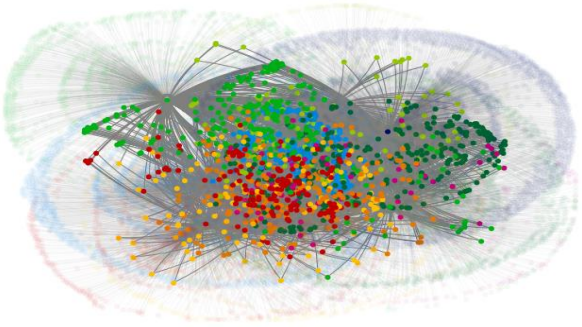


Figure 5. Network of connections of the 10 most referenced domains

Figure 5 shows the 1.397 nodes (about 12.8% of the total) with a higher than the average value. These were the small group of nodes located in the center of the graph. These nodes may be crucial in the flow of information, since they lied in the paths between other nodes in the network and therefore provided a link between them.

Toward the periphery, in opaque tones, we can see the remaining nodes, the ones left outside by the applied filter. The great majority of them had a low Betweenness Centrality index, that reached 0 for 9.295 nodes, that is, for the 85.3%. These values reflect a distribution of Pareto, in which a small number of nodes (about 13%) displayed the higher values of Betweenness Centrality, while a great number of nodes (87%) showed relatively low values in this index.

The In-Degree Index of this network had a minimum value of 0 and a maximum of 3.367; The Out-Degree presented a minimum of 0 and a maximum of 6; the Betweenness Centrality showed a minimum of 0 and a maximum of 56149284,89. These metrics evidenced a higher maximum value of In-Degree than of Out-Degree, what indicates that though a web domain may have been referenced many times (as the in the case of Youtube, with an In-Degree of 3.367), these references were done by many nodes. In other words, we can assert that the detected tendencies were actually a product of Lévy's collective intelligence, and not of reduced groups of nodes that fostered a particular interest.

Nine groups were configured inside the connection network of the 10 domains with the highest frequencies of appearance in the tweets. Group 1, despite being the most numerous, did not connect to any other groups, though other groups did connect to it. On average, each group established connections with five other groups; the average amount of nodes by group was 1.211 and that of the unique connections, 704.

We must highlight that the groups with a lower amount of nodes established connections with a greater amount of groups, to the point that groups 8 and 9 were connected to 8 of the 9 groups configured, while the groups with a greater amount of nodes –groups 1 and 2- were connected to less groups (0 and 1 group respectively). This could mean that a great number of the network nodes posted tweets referencing a particular web domain, while a minority of them, referenced in their tweets a greater variety of web domains.

Correlation between tendencies and metrics of the communication network.

In order to correlate the six (6) variables associated with the network of connections under study, we applied a multivariate analysis, relating pairs of variables of the metrics with the frequencies of the identified trends. The variables of the metrics were: In-Degree, Out-Degree, and Betweenness Centrality. The variables of the tendencies were: web domains (URL), hashtags, and lemmas.

	In-Degree	Out-Degree	Betweenness Centrality
URL	0,1627	**0,172**	0,146
Hashtag	0,0466	0,054	0,0454
Lemma	0,1961	**0,201**	**0,1833**

Table 3. Correlations

As shown in Table 3, in most of the relations between pairs of variables of the metrics and the tendencies of the topics of interest, we found a direct correlation, though weak.

The highest correlation was observed between lemmas and metrics, and the lowest between hashtags and metrics. In the first case, the highest correlation occurs between lemmas and out-degree, followed by lemmas and in-degree.

6 Conclusion

The methodological procedure used in this research allowed us to create a wide network of users interested in education starting from an initial group of 43 teachers.

Although the nodes of the initial group registered high values in the network metrics, their influence in the identified trends was low.

Most of the trends identified from the analyzed tweets were related to education and educational technologies that could enhance teaching and learning processes, as for instance, blogs, social networks as platforms for sharing documents and other resources, online journals and curation tools.

It stands out the association between education and technologies found through the text mining procedure applied to the tweets.

The importance of blogs as a trend was confirmed by its appearance among the web domains with the highest frequency of references in the tweets.

The direct correlation found particularly between the metrics of the network and the trends in the lemmas found in the analysis of the tweets, allows to conclude the importance of analyzing with particular attention the tweets published by users with a higher out-degree since they seemed to influence more the trends that arise from the studied network.

The analysis of lemmas seems to be more promising than that of hashtags for detecting trends in the tweets.

Since nearly 6 of each 10 tweets included a reference to a web domain, it would be interesting to be able to explore in a greater detail, what is what users are actually referencing through those web domains.

The results of this research and their usefulness for identifying trends in the topics of interest of educational professionals suggest we continue exploring the possibilities of social networks and the analysis of big data in the shaping academic communities.

References

Bakshy, E.; Hofman, J.M.; Mason, W.A.; Watts, D.J. (2011), Everyone's an Influencer: Quantifying Influence on Twitter, [en línea], disponible en: <http://research.yahoo.com/pub/3369>

Castells M. (2001), Internet y la sociedad red, [en línea], disponible en <http://tecnologiaedu.us.es/cuestionario/bibliovir/106.pdf>

Chew,C. and Eysenbach, G. (2010). Pandemics in the Age of Twitter: Content Analysis of Tweets during the 2009 H1N1 Outbreak., [en línea], disponible en <http://www.plosone.org/article/info:doi%2F10.1371%2Fjournal.pone.0014118>

Ferriter, W. M. (2010). Why teachers should try twitter. Educational Leadership, 67(5), 73. [en línea], disponible en

<http://ezproxy.utp.edu.co/docview/224840251?accountid=45809>

Gómez M., Leskovec J., Krause A., (2010), Inferring networks of diffusion and influence, [en línea], disponible en <http://dl.acm.org/citation.cfm?id=1835933>

Heverin, T. y Lisl, Z. (2010), Microblogging for Crisis Communication: Examination of Twitter Use in Response to a 2009 Violent Crisis in the Seattle-Tacoma, Washington Area, [en línea], disponible en <http://www.thomasheverin.com/uploads/4/6/5/8/4658640/heverin_iscram_2010.pdf>

Kwakn H., Lee C., Park H., and Moon S.,(2010), What is Twitter, a Social Network or a News Media?, [en línea], disponible en <http://an.kaist.ac.kr/~haewoon/papers/2010-www-twitter.pdf>

Lerman, K., and Ghosh, R.. (2010). Information contagion: an empirical study of the spread of news on digg and twitter social networks, *In Proceedings of 4th International Conference on Weblogs and Social Media (ICWSM)*. [en línea], disponible en <http://arxiv.org/PS_cache/arxiv/pdf/1003/1003.2664v1.pdf>

Lévy P (2004), Inteligencia colectiva. Por una antropología de ciberespacio (en línea), [en línea], disponible en <http://inteligenciacolectiva.bvsalud.org/public/documents/pdf/es/inteligenciaColectiva.pdf>

Shneiderman, B. (2011), Technology-Mediated Social Participation: The Next 25 Years of HCI Challenges, [en línea], disponible en <http://www.cs.umd.edu/localphp/hcil/tech-reports-search.php?number=2011-03>

Weng, J., Lim E., Jiang J., He Q., (2010), Twitter-Rank: finding topic-sensitive influential twitterers, [en línea], disponible en <http://dl.acm.org/citation.cfm?id=1718520>

Wu S., Hofman J.M, Mason J.,M., y Watts D. J, (2011), Who Says What to Whom on Twitter (¿Quién dice qué a quién en Twitter?), [en línea], disponible en http://research.yahoo.com/pub/3386

Esopo: sEnsors and SOcial POllution measurements

Vittoria Cozza
IIT CNR, Pisa, Italy
vittoria.cozza@iit.cnr.it

Ilaria Guagliardi
University of Calabria, Italy
ilaria.guagliardi@unical.it

Michelangelo Rubino
Expert System, Modena, Italy
michelangelo.rubino@email.com

Raffaele Cozza
Department of Computer Science
University of Milan, Italy
raffaele.cozza@studenti.unimi.it

Alessandra Martello
University of Calabria, Italy
a.martello@mat.unical.it

Marco Picelli
OverIT S.p.A.,
Fiume Veneto, Pordenone, Italy
marco.picelli@overit.it

Eustrat Zhupa
Uist *St. Paul the Apostle*,
Ohrid, Macedonia
eustrat.zhupa@uist.edu.mk

Abstract

In the following we present the idea of a smart sensor distributed platform where users collect pollution measurements by simply placing a small smart device out of their office or home window, a device that interacts with their smartphones. They provide time-geolocalized information that, trough an app, will be made available to the community that will have the chance to control the pollution level and eventually share it on the most popular social networks along with the related user's opinions and feedbacks. The big data coming from sensors and social networks will be analysed, in combination with local setting area data, in order to have a thorough view of the place the people live in and enhance our environmental conscience. The design of such a project, named Esopo, implies and requires technologies capable of providing data privacy, as it deals also with storing sensitive data, and efficiency, being the corresponding output prone to becoming unusable if not produced in real-time.

1 Introduction

The interest for the environmental issues is increasing continuously in many countries. The rapid development of urban areas has changed the physical, chemical, and biological composition of living environment (Guagliardi et al., 2012). As a consequence, millions of people living in and around urban areas are exposed to an unnatural and unhealthy environment. The increasing awareness that air, water, soil pollution induces a wide variety of adverse physiologic effects to humans, makes people more alert about environmental conditions. In particular, the effects of increasing amount of air pollutants, such as airborne particulate matter, nitrogen dioxide, sulfur dioxide, carbon monoxide on human health have been intensely studied in recent years, leading to unequivocal conclusions: high levels of pollutants are linked to increased rates of allergies in the less serious cases, to high risk of neoplasias in the worst ones. The problems connected to this subject pushed the World Health Organization (WHO, 2015) to define pollution guidelines and thresholds for each pollutants: e.g., the particulate matter annual mean is set to 10 mg/m3 (PM2.5) and 20 mg/m3 (PM10). This attention to environmental pollution can be supported by setting up a platform based on sensors detecting pollutants in the air. It has to be said that there are already systems used to control quality of air, yet the environmental sensors they use are limited in number and distributed in the main roads, consequently they cannot cover the whole area of interest.

We propose a system for collecting pollution data through portable sensors positioned in many points of a specific area and requiring no user intervention (passive detection) and for analysing this data. The monitoring platform should be composed of sensors that can detect enviromental measures such as Carbon Dioxide, Carbon Monoxide, Oxidizing and Reducing Gases and particle sensor. A group of people should be equipped with such sensors to detect pollution in a metropolitan area. Obviously, sensors need to be in contact with the air.

As regards the flow of data collected with sensors, measurements will be sent via bluetooth to people's personal devices and managed by the data logger mobile application, then will be transferred to remote servers and stored permanently. This requires the sensors to be connected to the platform through smartphones and tablets and, even more important, the data about the GPS position (latitude and longitude) to be sent as well as time data (timestamp). Pollution measurements, space and time are the three dimensions, detected anonymously, that will allow us to create a more detailed pollution map to monitor quarters, main and secondary streets of the cities. Pollution data about the current air conditions will be available in realtime to everyone by downloading the mobile application Air quality.

This is the second part of the project, what can be called as active detection. Indeed, every user can also share measurements on the most popular social networks, for instance by posting the level of PM10 in a particular area and adding a photo or a comment in Twitter as in Figure 1. This way,

Figure 1: Pollution measurement shared on Twitter.

everyone will be able to get information about the quality of air or to compare their perceptions with what is shown by the Air quality app. The project is therefore aimed at developing an application able to map data related to the environmental pollution and sharing it via social networks with the possibility of adding personal sensations.

Actually, many geographical areas are covered by pollution detectors, they are analysed according to pollutant thresholds that change on the basis of the law. Yet enviromental sensors have a sparse distribution on the territory. Such a system will provide the users with realtime information about the current air pollution levels within a denser map. As a plus, Esopo will analyse their opinions when they share air information on the most common social network, adding personal comments.

Geolocalized and time-tagged data along with user's opinions, will be used to generate a big pollution dataset.

The collected data will be accessed through multidimensional indexing structures (based on space, time and text). For deeper analyses it will be merged with other data regarding the area (data fusion) (Guo and Hassard, 2008; Guo et al., 2010).

2 Social and Smart Sensor networks

Nowadays, the interest for environmental issues is not restricted to few people, but it is growing, as well as the demand for pollution data in the Healthcare and the Security sectors. Actually, almost everyone owns a personal smart device, be it a smartphone or a tablet. The pollutants are usually concentrated in specific spots, e.g., the main city roads, as they are typically emitted by motor vehicles, but just few people know that their concentration can be different based on the time, usually more concentrated in the morning and less in the evening and night. We aim at designing a wireless sensor network to collect "affordable" information. When we use the adjective "affordable" we refer to open data, available for a free access. Citizens can monitor directly the quality of air combining the environmental sensors with their own mobile sensors. On the market, there is a number of sensor devices to be implemented in such kind of network and useful to control pollution 24/7. By law, the distribution of pollution control units must be revised periodically and, during this period, many things may change. Typically, just few control units can map all the pollutants. A smart distributed sensor network would cover a larger territory, also detecting more pollutants. This solution would integrate the control

units network with low costs for citizens and private companies. Once implemented, it will provide people with valuable information related to their life and this knowledge may eventually motivate them to advocate for a change.

We propose to build a sensor network collecting chemical, physical, and biological measurements, monitoring the environment with the minimum of user intervention: enhancing a number of user's smartphones with ad hoc sensors.

2.1 Sensors devices

We believe that a so called "social" wireless network of sensors should satisfy some general requirements, such as:

1. Sensors should be able to monitor the air quality around a user, especially nitrogen dioxide (NO2), sulphur dioxide (SO2), carbon monoxide (CO) and the so-called PM10.

2. The monitoring process should need very limited user intervention.

3. Sensors should be extensible, namely should be possible to connect, through an external interface, other peripherals, in order to extend the sensors capabilities for future purpose.

4. Sensors must connect in a wireless way to some portable smart devices, in order to gather data and send it to the back end infrastructure. A smartphone is believed to be the most suitable device for this task.

5. Sensors must be commercial off-the-shelf (cots).

Moreover, it is important to safeguard user privacy. To this aim, we define the following software requirements:

- All the possible API to access the sensors capabilities must be open source, so that it would be possibile to inspect them looking for threats to the privacy of the user.

- All applications developed must be released under open source licenses, so that each user - or central authority - could inspect them and be assured that there is no threat for the privacy of the user.

Given these requirements, we find out that there are just a few sensors suitable for our purpose (at the time of our study) namely: Air.Air sensors[1], CitiSense sensors[2], Sensordrone sensors[3] and M-Dust sensors, an innovative low-cost smart pm sensor[4]. Unfortunately, none of the previous options are able to satisfy all the requirements. For options 1) and 2), we were unable to find out whether these devices were really cots or just a proof of concept. Moreover, we were unable to understand exactly what kind of air parameter they are able to monitor and the API available to communicate with the sensors. Option 3) instead seems very interesting. Sensordrone (Sensordrone, 2015), in Figure 2, is a portable & wearable multisensor connectable to portable smart devices and can be turned into a multi-function environmental monitor: a carbon monoxide leak detector, a non-contact thermometer, a lux meter, a weather station.

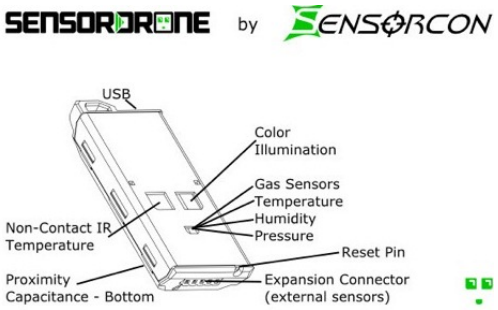


Figure 2: Sensordrone

From the information available on the website, it is able to communicate with Android and Apple devices through Bluethooth technology and it offers a well-defined, open source, API. Moreover, it is able to monitor most of the air quality parameters we need, but not the PM-10. On the contrary, M-Dust (Must, 2015; Soldo et al., 2012) can monitor PM10 and PM2.5 parameters, but it is not capable of directly communicating with a device such as a smartphone.

The idea then is to use the Sensordrone, extended with the particle sensor MDust for controlling PMs. This is possible since Sensordrone has also an expansion port were we can connect, as an expansion device, M-Dust. In interfacing Sensodrone with M-Dust, we must take into ac-

[1] https://www.kickstarter.com/projects/1886143677/airair-portable-air-quality-detector

[2] http://ucsdnews.ucsd.edu/pressrelease/small_portable_sensors_allow_users_to_monitor_exposure_to_pollution_on_thei

[3] http://sensorcon.com/sensordrone-1/

[4] http://www.particle-sensor.com/

count a technical inconvenient though, as the expansion port of Sensordrone accepts, as input, a signal ranging from 0 to 3 V, while M-Dust outputs a signal ranging from 0 to 4 V. The different input/output requirements, however, can be compensated using a voltage divider. However, at this stage, we are unable to exactly pinpoint the effect of the loss of resolution in doing this adaption.

2.2 Data flow

All the data measured with the environmental sensors, being it on demand or periodic, is saved in a local database on a personal Sensordrone device, that can be easily sent as a csv file to our data logger application, installed on the user smartphone. Each hour the data logger downloads a new csv file from Sensordrone and merges data acquired by the pollution sensor, the timestamp and other information coming from the smartphone, such as the GPS data tracking the position of the user.

Detected data is sent to a repository. Indeed the continuous data flow, composed of measurements, timestamps and user location, is sent and stored to the big data repository. At the same time, the user can visualize the current measurements, as well as the data of the past, by using an air quality application on his/her smart device. The repository will also collect text written by the users while they share and comment sensor data on the most popular social networks. The chosen repository will support georeferenced data, a hybrid multidimensional index to speed up soft real-time and offline analyses. The data can be retrieved with a popularity spatio-temporal-keyword search engine, as proposed in (Cozza et al., 2013). The system will also provide a polarity detection module to understand whether comments express a positive or negative sensation about the air and an opinion mining module to extract relevant information from unstructured social comments.

2.3 Privacy concerns

Constraints about profiling and trustworthiness will need to be matched. As regards profiling, different user profiles will be investigated to understand to what extent confidentiality and multiple view and map customisations can be satisfied, namely not only the information to show but also how to show it. Particular attention will be given on storing, processing and sharing that data referring to more than one subject (so called "multi-subject personal data") (Gnesi et al., 2014). We envisage a support architecture based on privacy policies, through which users can edit their privacy preferences, appropriately enforced at the time of the actual data processing (Casassa-Mont et al., 2015).

The second aspect to consider is the Data trustworthiness: an in-depth analysis of data trustworthiness is required to identify and test a model able to exploit geotagged data and to get the highest level of reliability. Sensor data cannot be linked to any user and its physical location.

Privacy concerns arise beyond data content and focus on context information such as the location of a sensor initiating data communication. The problem of data unintentionally shared when using and producing georeferenced information is formulated and discussed in (Friginal et al., 2013), (Cortez et al., 2015).Table 1 summarizes the data categories involved in Esopo: volunteered entered by the users, observed and inferred. Data

Volunteered data	**Observed data**	**Inferred data**
Pollution measurements	Online activity (time, location from GPS), Pollution level exposures	Habits and lifestyle
Interaction with airQuality app: sharing, comment, likes	Online activity (time, location from GPS)	Opinions on air quality, relationship: friends, followers, mentions

Table 1: Social and sensor data

can be inferred by information intelligence analysis: data fusion can enhance informativeness of data coming from sensors, Opinion Mining and Polarity Detection or any Social Network Analysis can be performed on these data shared on social networks through the air Quality app.

For a state-of-the-art survey of existing privacy-preserving techniques in WSNs readers should refer to (Li et al, 2009).

2.4 Applications

Due to the variety and the volume of data retrieved and analyzed we can offer several applications.

This big data can be "consumed" in real-time, as it is "produced", and the user can get the cur-

rent level of pollution. This way, we can inform users about the air pollution at a specific moment of the day and can get reports, statistics and charts to evaluate the pollution level in a particular area, over the last weeks or months. This will be possible through a modern user friendly interface application. Anytime the users have the chance to check on personal devices the information about pollution (consuming information) and are able to share measurements, they will send comments and images (producing new information) to the most popular social networks. It is already available an air quality mobile application for Sensordrone providing the end users with a view about the quality of air for a specific area and a second application that allows them to take measurements with the Sensordrone and to post them to Facebook, Twitter, Google+.

Big data can be "consumed" offline as well, this means that we can combine information about the environment: pollutants, humidity, oxidizing and reducing gases with social information and other information about an area and therefore produce statistics and predictive analyses. Furthermore, we think that if we carry out experiments and analyses in a metropolitan area of Italy, let's say Milan, where there are already air pollution stations, then it would be significant to combine this data with that coming from the wireless sensor network to enrich the information provided to end users and have a wider view of the environmental conditions in different locations of that metropolitan area, even those ones not monitored by Arpa (Arpa, 2015).

All the data acquired so far has a relevant added value for others applications too, as it is collected not only pollution data, but also data about the people daily movements for instance: the starting point of each journey and where it ended, the most common paths (let's call it mobility data); data about the busiest areas; data about public places where people connect to the internet (libraries, council houses, schools...). In a nutshell, a large amount of data that could be further analysed.

3 Related Work

In the main geographical areas, air quality data currently available from government agencies does not provide enough detailed measurements within particular neighbourhoods, then several projects have focused on increasing the spatial resolution of air pollution data using ubiquitous sensor networks. These works did raise the spatial granularity compared with data from fixed air pollution monitoring sites. In (Devarakonda et al., 2015) the authors present a vehicular-based mobile approach for measuring fine-grained air quality in real-time. They provide users with a small sensor that they should bring on their veichles or in public transportation to collect realtime information.

In (Hu et al., 2014) the authors combine air pollution and human energy expenditure data to give individuals real-time personal air pollution exposure estimates. They monitor pollution in an area and at the same time analyse users' life behaviours, specially they apply multiple data mining techniques to find out associations among activity modes, locations and the inhaled pollution.

The authors understand the relevance of automatically analysing how pollution level are perceived by people and combining air pollution exposure with personal health.

In (Leonardi et al., 2014) the authors propose SecondNose, an air quality mobile crowdsensing service, aimed at collecting environmental data to monitor some air pollution indicators to foster participants reflection on their overall exposure to pollutants. At the time of the work, SecondNose aggregates more than 30k data points daily from 80 citizens in Trento, northern Italy. Esopo has many features in common with SecondNose, in addition it encourages users to share pollution measurements on the social networks and, consequently, the combined analysis of sensor data and social data.

4 Conclusions

In this work we have described the project idea of a smart sensor network that stores environmental information from sensors and eventually collects social network comments about it. Sensor data can be shared through users that will have a real-time snapshot of the enviromental pollution in a defined area at a specific time and that may want to add their personal sensations. Sensors will share minute-by-minute air quality measurements that could provide a better understanding of risks related to potentially harmful exposure in the area and eventually identify patterns for any given day, week, month or year.

Furthemore, the analysis of sensor data com-

bined with people's sentiment on social networks related to it, permits a semantic analysis of collected measuremens through sensations perceived by people.

5 Acknowledgments

Work partly supported by the Registro.it project My Information Bubble (MIB). The authors thank Paolo Palana and Antonio Gabriele for sharing their technical knowledge on sensors and giving them support.

References

Arpa, Agenzia regionale per la protezione dell'ambiente. Homepage: http:/ita.arpalombardia.it/ITA/index.asp

Casassa-Mont, Marco and Matteucci, Ilaria and Petrocchi, Marinella and Sbodio, MarcoLuca: Towards safer information sharing in the cloud International Journal of Information Security, volume 14, number 4, pages: 319-334. issn 1615-5262, Springer Berlin Heidelberg.

M Nunez del Prado Cortez, J Friginal, Geo-Location Inference Attacks: From Modelling to Privacy Risk Assessment. Tenth European Dependable Computing Conference (EDCC), New Castle, UK., May 2014.

V. Cozza, A. Messina, D. Montesi, L. Arietta, M. Magnani: Spatio-temporal keyword queries in social networks. In Springer, editor, B. Catania, G. Guerrini, and J. Pokorn(Eds.): ADBIS 2013, volume 8133 of LNCS, pages 70-83, 2013.

Srinivas Devarakonda, Parveen Sevusu, Hongzhang Liu, Ruilin Liu, Liviu Iftode, and Badri Nath. 2013. Real-time air quality monitoring through mobile sensing in metropolitan areas. In Proceedings of the 2nd ACM SIGKDD International Workshop on Urban Computing (UrbComp '13).

J. Friginal, J. Guiochet, M.-O. Killijian: Towards a Privacy Risk Assessment Methodology for Location-Based Systems. Mobiquitous2013.

Stefania Gnesi and Ilaria Matteucci and Corrado Moiso and Paolo Mori and Marinella Petrocchi and Michele Vescovi: My Data, Your Data, Our Data: Managing Privacy Preferences in Multiple Subjects Personal Data Proceedings of Privacy Technologies and Policy - Second Annual Privacy Forum, APF 2014, Athens, Greece.

I. Guagliardi, D. Cicchella, R. De Rosa, 2012. A geostatistical approach to assess concentration and spatial distribution of heavy metals in urban soils. Water, Air & Soil Pollution 223: 5983-5998.

Yajie Ma, Yike Guo, Moustafa Ghanem: Distributed pattern recognition for air quality analysis in sensor network system. IADIS International Journal on Computer Science and Information Systems Vol. 5, No.1, 2010, pp. 87-100ISSN:1646-3692

Yike Guo, John Hassard: Air pollution Monitoring and Mining based on Sensor Grid in London Sensors (Basel). Jun 2008; 8(6): 3601-3623.

Ke Hu, Timothy Davison, Ashfaqur Rahman, and Vijay Sivaraman. 2014. Air Pollution Exposure Estimation and Finding Association with Human Activity using Wearable Sensor Network. In Proceedings of the MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis (MLSDA'14).

Chiara Leonardi , Andrea Cappellotto , Michele Caraviello , Bruno Lepri , Fabrizio Antonelli, SecondNose: an air quality mobile crowdsensing system, Proceedings of the 8th Nordic Conference on Human-Computer Interaction: Fun, Fast, Foundational, October 26-30, 2014, Helsinki, Finland

Na Li and Nan Zhang and Sajal K. Das and Bhavani Thuraisingham: Privacy preservation in wireless sensor networks: A state-of-the-art survey. Ad Hoc Networks 7 (2009), p. 15011514, 2009.

D. Soldo, A. Quarto, V. Di Lecce: M-DUST: an Innovative Low-Cost Smart PM Sensor. International Instrumentation and Measurement Technology Conference. Measurement Technology Conference: Industrial Track - Protecting the Environment. Graz (Austria) - April 14th, 2012, pp. 1823-1828.

Sensordrone project. Homepage: http://sensorcon.com/sensordrone/

MDust homepage: http://www.particle-sensor.com/

World Health Organization (WHO), 2015. Homepage: http://www.who.int/en/

Causation Generalization Through the Identification of Equivalent Nodes in Causal Sparse Graphs Constructed from Text using Node Similarity Strategies

Brett Drury Jorge Valverde-Rebaza Alneu de Andrade Lopes
Department of Computer Science
ICMC, University of São Paulo
C.P. 668, CEP 13560-970, São Carlos
{bdrury, jvalverr, alneu}@icmc.usp.br

Abstract

Causal Bayesian Graphs can be constructed from causal information in text. These graphs can be sparse because the cause or effect event can be expressed in various ways to represent the same information. This sparseness can corrupt inferences made on the graph. This paper proposes to reduce sparseness by merging: equivalent nodes and their edges. This paper presents a number of experiments that evaluates the applicability of node similarity techniques to detect equivalent nodes. The experiments found that techniques that rely upon combination of node contents and structural information are the most accurate strategies, specifically we have employed: 1. node name similarity and 2. combination of node name similarity and common neighbours (SMCN). In addition, the SMCN returns "better" equivalent nodes than the string matching strategy.

1 Introduction

Graphs can be constructed to represent a specific domain from which inferences can be made about a future event(s) based on observations (Newman, 2010). These graphs tend to be constructed: 1. manually from information elicited from experts in the field or 2. from other information sources (Horny, 2014). A manual construction process can be slow, and represent a partial slice of the domain. An alternative approach is to construct a domain specific graph from information in text. The advantage of this approach is that graphs can be constructed automatically, and therefore the construction process can be quick and the graph domain coverage can be more comprehensive than a graph constructed manually (Hensman, 2004; Jin and Srihari, 2007).

1.1 Node Merge Problem

A major disadvantage of constructing graphs from text is that the same assertions can be stated in various different ways. These variations may be in the words chosen and their order (Jin and Srihari, 2007). The consequence of varying language is that the graph generated from it can have many nodes, that have one edge, consequently accurate inference may be difficult due to the sparse structure of the graph (Tsang and Stevenson, 2010). An approach to minimize this characteristic of text built graphs is to merge similar nodes and their edges. This will improve the graph by: 1. decreasing the number of nodes, 2. increasing the average number of edges per node and 3. inferring new causes or effects for events which are not explicitly stated in the text the graph is constructed from. For example, "*... começa a reduzir preço do etanol*" and "*Preço do etanol começa a diminuir*" represent the same concept, but are written in a different order. In a graph constructed from text these two events would be two different nodes, but arguably these nodes should be merged because they represent the same event.

The merged node process is demonstrated in Figures 1 and 2. The figures demonstrate two candidates nodes for merging *B* and *B#*. The two candidates have very similar node names as well as common neighbours *C* and *A*. The merge process joins the two nodes into one node *B[B#]* which combines the neighbours of the previous two graphs. In a causal Bayesian Network where in-links are causes and out-links are effects, the proposed merge process would infer new causes and effects which are not explicitly stated in the construction text (Girju, 2003; Shpitser and Pearl, 2008).

1.2 Node Similarity

The proposal presented by this work is that identical nodes can be identified by Node Similarity

measures and these nodes are candidates for merging. It should be noted that the aim of this work is not to present general similarity measures, but to identify strategies which can accurately identify nodes that represent the same event.

This paper will present a series of experiments that evaluate a number of common node similarity measures as well as a number of novel variations of these techniques. This paper will conform to the following format: Related Work, Proposed Techniques, Evaluation and Future Work.

2 Related Work

The related work covers two main areas: causal graphs constructed from text and node similarity measure.

2.1 Causal Directed Graphs from Text

Causal directed graphs, are graphical models that represent the inference process between two variables: X and Y, through the use of two nodes and a directed link from: X to Y, whenever Y responds to changes in X when all other variables are being held constant (Shpitser and Pearl, 2008).

A common problem in this domain is the manner of the construction of the Bayesian Graph. Manual construction can be a labour intensive process that may not provide good coverage for a specific domain. An alternative is to construct graphs from information in text.

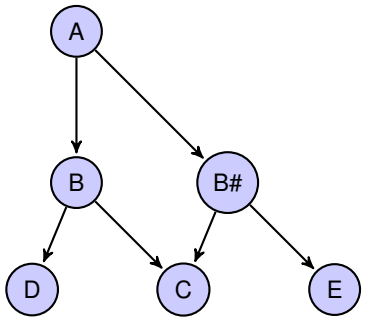


Figure 1: Candidate for Node Merging.

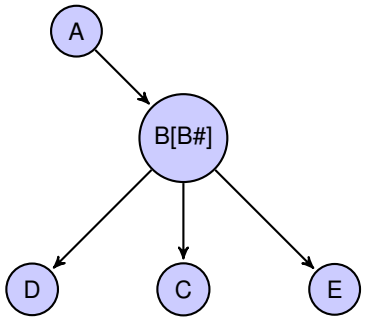


Figure 2: Graph after candidate node merged.

A number of attempts to construct Causal Bayesian Networks from text have been documented. An early attempt at constructing a Bayesian Graph from text was proposed by (Sanchez-Graillet and Poesio, 2004). They constructed a Causal Bayesian Graph from causal relations in text. They generalize about causal relations by identifying synonyms in similar event phrases. The synonyms are identified using external lexical resources that they admit did not provide full coverage. (Bojduj, 2009) used decision rules to extract causal relations to construct a Bayesian graph. It was not clear how causality was generalized and if the constructed graph was sparse. (Raghuram et al., 2011) produced a prototype called Auto-Bayesian that constructed Bayesian Graphss from causal relations in text. Finally, (Miranda Ackerman, 2012) produced a causal Bayesian Networks from causal topics in text. The topic approach provided a partial generalization about causation in the network.

2.2 Node Similarity Measures

The notion of similarity is documented in many domains, consequently similarity can be measured in a variety of ways. The notion of similarity is dependent upon the domain and the appropriate definition of similarity for that domain (Jeh and Widom, 2002).

In graphs, similarity between a pair of nodes indicates that these nodes share a common relation, consequently, similarity measures can be used to: 1. predict new relationships (Valverde-Rebaza and Lopes, 2013; Lü and Zhou, 2011), 2. detect communities (Valejo et al., 2014), 3. node classification (Valverde-Rebaza et al., 2014), and 4. improve the graph construction (Berton et al., 2015).

In graphs, where similarity among nodes is based solely on graph structure, similarity is referred to as structural similarity. Structural similarity measures can be grouped into measures that rely upon: 1. local or 2 .global information.

Global measures can obtain higher accuracy measures than local measures, but they are computational complex, and consequently are unfeasible for large-scale graphs. Local measures are generally faster, but obtain lower accuracy than global measures. Examples of common local measures are: 1. Common Neighbours, 2. Jaccard coeffi-

cient, 3. Adamic Adar, 4. Resource Allocation and 5. Preferential Attachment measures (Lü and Zhou, 2011; Valverde-Rebaza and Lopes, 2013). Standard global measures are: 1. SimRank, 3. Katz, and 3. Rooted PageRank (Lü and Zhou, 2011; Valverde-Rebaza and Lopes, 2013).

The following describes two common similarity measures: Common Neighbours and SimRank using a node pair: v_i and v_j that is assigned a score s_{v_i,v_j}. $\Gamma(v_i)$ denotes a set of neighbours of v_i.

The **Common Neighbours (CN)** technique assumes that v_i and v_j are similar if they share neighbours, therefore CN refers to the size of the set of all common neighbours of both v_i and v_j according to Eq. 1.

$$s^{CN}_{v_i,v_j} = |\Gamma(v_i) \cap \Gamma(v_j)| \tag{1}$$

The **SimRank (SR)** technique assumes two nodes are similar if they are joined to similar neighbours. The SimRank measure is defined as Eq. 2.

$$s^{SR}_{v_i,v_j} = \gamma \cdot \frac{\sum_{v_k \in \Gamma(v_i)} \sum_{v_m \in \Gamma(v_j)} s^{SR}_{v_k,v_m}}{|\Gamma(v_i)| \cdot |\Gamma(v_j)|} \tag{2}$$

Where the parameter $\gamma \in [0, 1]$ is the decay factor. Due to SimRank can also be interpreted in terms of a random walk process, that is, the expected value of $s^{SR}_{v_i,v_j}$ measures how soon two random walkers, respectively starting from nodes v_i and v_j, are expected to meet at a certain node.

3 Proposed Techniques

The aim of the proposed techniques is to generalize causal relationships in a causal graph constructed from text without recourse to lexical resources as per (Sanchez-Graillet and Poesio, 2004) by identifying equivalent nodes and merging them. We call this the Node Merge Problem [1].

This paper evaluates a number of node similarity techniques for their ability to identify merge candidates (nodes which have different names, but represent the same event). The base techniques are three common strategies: SimRank, Common Neighbours and Node Name Similarity (String matching) (Robles-Kelly and Hancock, 2004). These techniques are commonly used to identify similar nodes in pre-processing step in link prediction strategies (Lü and Zhou, 2011; Valverde-Rebaza and Lopes, 2013). The strategies which were developed for this paper were: Fuzzy SimRank and String matching with common neighbours.

3.1 Fuzzy SimRank

Fuzzy SimRank is an adaptation of SimRank. SimRank is a recursive algorithm which relies upon the structural similarity of nodes. In sparsely connected graphs the structure is poor because very few nodes are connected, and consequently SimRank can not make accurate comparisons between nodes (Jeh and Widom, 2002). Fuzzy SimRank assumes an implied structure through partial edge similarity. The SimRank algorithm computes similarity by making a direct comparison of neighbours of given nodes. A match is only recorded when the nodes are exactly the same. Graphs created from text may have many similar nodes which when compared will be scored the same as nodes that are not related. Fuzzy SimRank applied a value between 0 to 1 based upon the similarity of the node names, i.e. a score of 1 indicates that the node names are equal, and a score of 0 indicates that the nodes names have no common text. The values computed for the similarity between nodes are computed with common string matching algorithms. The string matching algorithms used in the Fuzzy SimRank algorithms for this paper were: Longest Common Sub-sequence, Levenstein Distance and Sorensen Distance (Rahm and Bernstein, 2001).

3.2 String matching with common neighbours

String matching with common neighbours (SMCN) is a technique that computes a similarity between two nodes using: node name similarity and common neighbours. The common neighbours measure was altered to compute two similarity measures: an in-link and out-link similarity because in-links and out-links represent cause and effect respectively, consequently in-links and out-links for equivalent nodes can not be the same. The SMCN is represented by:

$$< Sim(N_1O, N_2O) + Sim(N_1I, N_2I) + Sim(N_1N, N_2N) > \tag{3}$$

[1] The node merge problem was explained on page 1

where, N_1O is the out-links of Node 1 in a two Node comparison pair, N_2O is the out-links of Node 2 in a two Node comparison pair, N_1I is the in-links of Node 1 in a two Node comparison pair, N_2I is the in-links of Node 2 in a two Node comparison pair, N_1N is the Node name of Node 1 in a two Node comparison pair and N_2N is the Node name of Node 2 in a two Node comparison pair.

There were three versions of SMCN which varied the similarity measure (Sim) used for the nodes comparison. The first similarity measure was a Jacard distance, that relied upon exact matching of nodes to compute a similarity between neighbours of two nodes. The remaining variations computed similarity between neighbours by using a Longest Common Subsequence similarity measure (LCSM). The LCSM measure is approximated by comparing the node names of all the neighbours of one of the candidate node against all of the all the neighbours of other of the candidate node. An average is taken of all of the similarity scores. This measure is demonstrated in Algorithm 1. The algorithm iterates through all of the nodes and compares each node with all of the nodes in the graph. The node pairs that have a similarity above a pre-determined threshold are marked as candidates for merging. It should be noted that a node can be marked as a merge candidate for more than one node, consequently a merged node will represent at least 2 nodes and a maximum of $n - 1$ nodes where n represents the number of nodes in the graph.

Input: N1,N2, threshold
Output: Sim
`/* N1 = Neighbours of Node 1, N2 = Neighbours of Node 2 */`
`/* Sim = similarity, threshold = lower bound similarity score */`
localSim = ()
for $node1$ *in* $N1$ **do**
  for $node2$ *in* $N2$ **do**
    sim = simularity($node1$,$node2$)
    if *sim* > *threshold* **then**
      localSim.push(sim)
    end
  end
end
return (mean(localSim))

Algorithm 1: Fuzzy Node Matching

The variations of the fuzzy node matching used differing threshold values, these values were a range of $>= 0.0$ $<= 1.0$, where 1.0 is a perfect match. The threshold values used for the variations were: 0.9 and 0.0. These values were chosen to see if that 'neighbour near misses' (0.9) produced better results than measuring the similarity of all neighbours.

4 Evaluation

The evaluation was intended to demonstrate the ability of the proposed candidate techniques to identify equivalent nodes in a graph. These nodes would be candidates for merging. The candidate techniques were evaluated on a graph created from Brazilian - Portuguese news stories [2] . The graph was created from causal relations extracted from the Brazilian - Portuguese news corpus. The relations were extracted using Levin's causative pattern: $NP\ V\ NP$, where NP is a noun phrase and V is a causal verb (Levin, 1993). The verb used in these experiments was the verb " causar ". This verb was chosen because: 1. it is a simple causative verb and consequently it will not form part of the cause or effect and 2. it is unambiguous. Levin's pattern assumes that: the first NP is the cause and the second is the effect. The position of cause and effect NP can be reversed. The reversing of the cause and effect NPs in these experiments was based upon lexical indicators such as "por" or "de". An example of this phenomenon is demonstrated in the phrase: " falta de chuva por causar de seca ", the NP 'falta de chuva' is the effect rather than the cause because of the preposition "de" .

The graph was created by transforming the NPs into nodes. The nodes were connected using the causal verbs. For example, the phrase " falta de chuva por causar de seca " would be transformed in into the structure shown in Figure 3.

The final graph contained 4045 nodes and 2180 edges. It was expected that this graph would contain duplicate nodes because the corpus it was constructed from contained repeating themes over a long period of time.

The typical node similarity evaluation strategies such as *Top K holdout* were not appropriate for this problem because edges in this graph do not indicate similarity, but cause or effect. This was

[2]Graph available from `https://goo.gl/IPe8qB` in pickled NetworkX Digraph Format

confirmed in a brief experiment where all candidate similarity strategies failed to identify the missing neighbours. We therefore used a manual evaluation strategy. The evaluation were conducted by a single annotator. The three evaluations were: precision for top 'n' similarities for 'n' randomly selected nodes, precision for most statistically significant similarities and precision by similarity score.

4.1 Precision for 'n' similarities for 'n' randomly evaluation

This evaluation is adapted from the information retrieval literature (Manning et al., 2008). Thus, randomly are selected 10 nodes from the graph and ranked the most similar nodes by descending accuracy score from 1 to n. The evaluation verified whether two nodes represented equivalent events. Thus, was evaluated: a. 5 most similar nodes, b. 10 most similar nodes and c. 20 most similar nodes. An average of the results for all nodes was then calculated. The results are in Figure 4. Strategies which returned no documents or a score of 0 for intervals are excluded from the diagram for clarity. The results demonstrate that rank is not a good indicator for node equivalence as all strategies performed poorly. The SimRank variations scored 0 accuracy or did not return any results for all of the selected nodes. The local similarity measures fared little better. Although the evaluation was limited it is an indication that rank provides little information when identifying equivalent nodes.

4.2 Precision for most statistically significant similarities evaluation

In this subsection, we evaluate if statistical significance was an indicator of node equivalence. Statistical significance in this case was the number of standard deviations between an accuracy for a node pair and average accuracy for all node pairs.

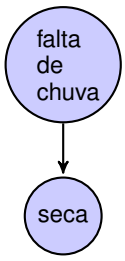


Figure 3: Sample Structure.

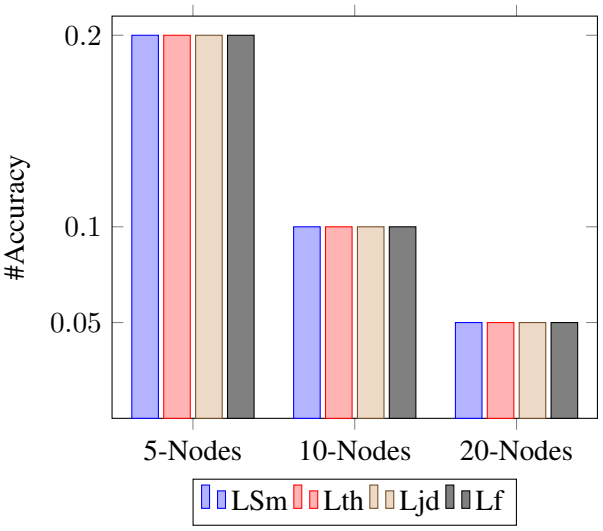


Figure 4: Accuracy for Randomly Selected Nodes, *where Lsm = Local similarity with string matching, Lth = Local similarity with string matching and common nodes with a minimum similarity threshold of 0.9, Ljd = Local similarity with string matching and common nodes with a minimum similarity threshold of 1.0 and Lf = Local similarity with string matching and common nodes with a minimum similarity threshold of >0.0*

This evaluation computed node similarities for every possible combination of nodes in the graph. The candidate node pair similarities were ranked by node (as per previous evaluation). A standard deviation is computed from the non zero node similarities. The number of standard deviations is computed between: 1. the most similar node pair for a given node and 2. the second most similar node pair. All the node pairs are then ranked by the number of standard deviations. Thus, was evaluated the: a. 5 most statistically significant, b. 10 most statistically significant and c. 20 most statistically significant, similar candidate pairs. The results are in Figure 5. Techniques which scored 0 for all of the sample intervals were excluded from the diagram for clarity.

The techniques provided improved candidate pair equivalences. The SimRank variations which used Levenstein or Common Longest Sequence generated better node equivalence pairs than the basic SimRank. However, do not was observed a statistically significant among node pairs. The best results were gained by the string matching (Lsm) approach. The approach returned very similar node pairs where the difference between the node names were minor differences in words. An example is provided in Table 1. For example in the first example the only difference between the pairs is the word **nesta**.

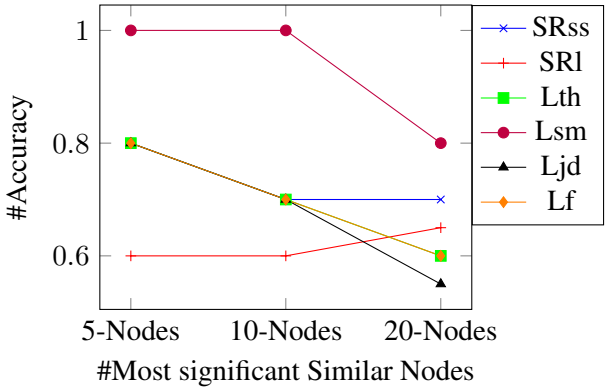


Figure 5: Accuracy for the statistically significant most similar nodes, *where, SRss = SimRank with Common Subsequence, SRl = SimRank with Levenstein Distance, Lsm = Local similarity with string matching, Lth = Local similarity with string matching and common nodes with a minimum similarity threshold of 0.9, Ljd = Local similarity with string matching and common nodes with a minimum similarity threshold of 1.0 and Lf = Local similarity with string matching and common nodes with a minimum similarity threshold of >0.0*

Node 1 Node Name	Node 2 Node Name
reconheceu **nesta** terça-feira pode faltar gasolina alguns postos	reconheceu terça-feira pode faltar gasolina alguns postos
traders importaram cerca toneladas produto desde outubro	**operadores** importaram cerca toneladas produto desde outubro
acusações envolvimento mensalão **esquema financiamento ilegal suposta compra deputados pelo**	acusações envolvimento mensalão

Table 1: Equivalent Node Pairs Examples

4.3 Accuracy by similarity score evaluation

The goal here is evaluate if the node similarity score was an indicator of node equivalence. The evaluation computed a similarity score for each node candidate pair. The evaluation created a range of $0.5 <= 1.0$ in steps of 0.1, i.e there were 5 sub-ranges in the overall range. The lower bound of the sub-range acts as minimum similarity and the upper bound acts a maximum similarity. For each of these sub-ranges candidate pairs were randomly chosen and evaluated for node equivalence. The results are demonstrated in Figure 6. Techniques that scored 0 for all intervals are not included. The results show that the SimRank variants perform poorly. The string matching (Lsm) did improve accuracy with very high similarities. At these high similarities the differences between node names was very small.

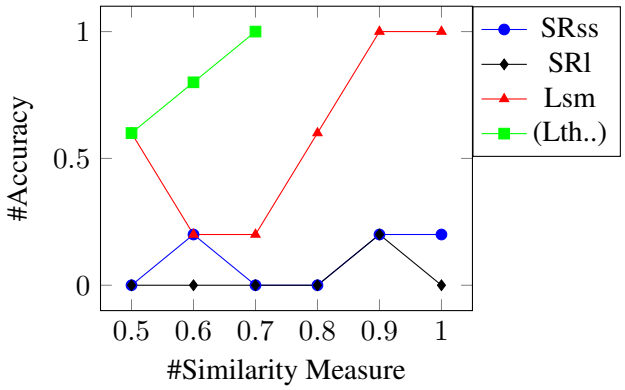


Figure 6: Accuracy by confidence intervals, where *SRss = SimRank with Common Subsequence, SRl = SimRank with Levenstein Distance, Lsm = Local similarity with string matching and Lth.. = L similarity with string matching and common nodes (techniques: Lth,Ljd, and Lf).*

The techniques that combined string matching with common neighbours performed well, gaining the best results at similarity level 0.7 after which no candidates pairs were returned. In contrast with the string matching results in the previous evaluation, the SMCN techniques returned "less" similar node names, but the events were equivalent. The common neighbours reinforced the notion of equivalence identified through string similarity. A comparison of high similarity examples from the string matching (SM) and SMCN techniques is shown in Table 2. It is quite clear from the comparison that the high similarity from the string matching returns node names where the differences are due to extraneous information, i.e the removal of the differences did not alter the meaning of the sentences. The SMCN differences were equivalents where removing the differences would change the meaning of the sentence.

5 Conclusion

The results demonstrate that local measures return the best results when compared to the various global (SimRank) techniques. In particular, the local measures that used: 1. node name similarity and 2. node name similarity with common neighbours (SMCN) produced the best results. It is arguable that the SMCN technique gained " better results " than the node name similarity technique. The node name similarity returned nodes that had similar node names that were differentiated by:

Node 1 Node Name	Node 2 Node Name	Technique
infecção pode destruir rapidamente tecido **causar** danos irreversíveis	infecção pode destruir rapidamente tecido **provocar** danos irreversíveis	SMCN
radiografia pulmões jornalista mostrou inflamações características doença	radiografia pulmões jornalista mostrou **infiltrações** inflamações características doença	SMCN
edmundo volta após sofrer **dívárias** punições disciplinares	edmundo volta após sofrer **diversas** punições disciplinare	SMCN
caso consigam manter vendas elevadas exterior	caso consigam manter vendas elevadas exterior **por**	SM
depredações piquetes durante greve geral	depredações piquetes durante greve geral **ontem**	SM
reconheceu terça-feira pode faltar gasolina alguns postos	reconheceu **nesta** terça-feira pode faltar gasolina alguns postos	SM

Table 2: Equivalent Node Pairs Examples (Confidence)

additional characters or words whereas the SMCN technique returned nodes that had lower node name similarity, but conveyed the same meaning. In addition the SMCN technique has the potential to be used in an iterative process because increasing the number edges may identify additional equivalent nodes. Furthermore, the SMCN technique avoids a common mistake made by the node name similarity technique, where two node names have a high superficial similarity, but convey the opposite meaning, for example 'momento oportuno' and 'momento inoportuno'. The SMCN similarity score would be low because these two nodes would have different edges. The use of partial node name (fuzzy) matching in the global and local measures did not improve the accuracy of the technique.

In general Node Similarity measures seem to be a viable strategy for identifying equivalent nodes in a "node merge" causation generalization strategy.

5.1 Future Work

The limitations of manual evaluations is that the amount of data that can be evaluated is restricted and the interpretation of results can be subjective, and open to errors. Consequently, the next step is to construct a larger graph and adapt one of traditional neighbour prediction evaluations, although at this stage it is not clear which one. In addition at the most accurate setting the *SMCN* strategy reduced the node count by *1%*, therefore we will be required to find settings that increase the number of nodes merged without sacrificing accuracy.

This work, we believe, has great potential in the generalization of causal statements in text and graph construction because it allows the inference of new causes and effects that are not stated explicitly in the construction text.

Acknowledgments

This work was partially supported by the São Paulo Research Foundation (FAPESP) grants: 2013/12191-5, 2011/22749-8 and 2011/20451-1.

References

L. Berton, J. Valverde-Rebaza, and A. Lopes. 2015. Link prediction in graph construction for supervised and semi-supervised learning. In *Proceedings of The 2015 International Joint Conference on Neural Networks*, IJCNN 2015, pages 1818–1825. IEEE.

B. N. Bojduj. 2009. Extraction of causal-association networks from unstructured text data. Master's thesis, California Polytechnic State University.

R. Girju. 2003. Automatic detection of causal relations for question answering. In *Proceedings of the ACL 2003 Workshop on Multilingual Summarization and Question Answering - Volume 12*, MultiSumQA '03, pages 76–83. Association for Computational Linguistics.

S. Hensman. 2004. Construction of conceptual graph representation of texts. In *Proceedings of the Student Research Workshop at HLT-NAACL 2004*, HLT-SRWS '04, pages 49–54. Association for Computational Linguistics.

M. Horny. 2014. Bayesian statistics. Technical report, Boston UNiversity.

G. Jeh and J. Widom. 2002. Simrank: A measure of structural-context similarity. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '02, pages 538–543. ACM.

W. Jin and R. K. Srihari. 2007. Graph-based text representation and knowledge discovery. In *Proceedings of the 2007 ACM Symposium on Applied Computing*, SAC '07, pages 807–811. ACM.

B. Levin. 1993. *English verb classes and alternations : a preliminary investigation*.

L. Lü and T. Zhou. 2011. Link prediction in complex networks: A survey. *Physica A*, 390(6):1150 – 1170.

C. D. Manning, P. Raghavan, and H. Schütze. 2008. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA.

E. J. Miranda Ackerman. 2012. Extracting a causal network of news topics. In Pilar Herrero, Hervé Panetto, Robert Meersman, and Tharam Dillon, editors, *On the Move to Meaningful Internet Systems: OTM 2012 Workshops*, volume 7567 of *Lecture Notes in Computer Science*, pages 33–42.

M. E. J. Newman. 2010. *Networks: an introduction*. Oxford University Press.

S. Raghuram, Y. Xia, J. Ge, M. Palakal, J. Jones, D. Pecenka, E. Tinsley, J. Bandos, and J. Geesaman. 2011. Autobayesian: developing bayesian networks based on text mining. In *Database Systems for Advanced Applications*, pages 450–453. Springer.

E. Rahm and P. A. Bernstein. 2001. A survey of approaches to automatic schema matching. *the VLDB Journal*, 10(4):334–350.

A. Robles-Kelly and E. R. Hancock. 2004. String edit distance, random walks and graph matching. *International Journal of Pattern Recognition and Artificial Intelligence*, 18(03):315–327.

O. Sanchez-Graillet and M. Poesio. 2004. Acquiring bayesian networks from text. In *LREC*. European Language Resources Association.

I. Shpitser and J. Pearl. 2008. Complete identification methods for the causal hierarchy. *J. Mach. Learn. Res.*, 9:1941–1979.

V. Tsang and S. Stevenson. 2010. A graph-theoretic framework for semantic distance. *Comput. Linguist.*, 36(1):31–69.

A. Valejo, J. Valverde-Rebaza, B. Drury, and A. Lopes. 2014. Multilevel refinement based on neighborhood similarity. In *Proceedings of the 18th International Database Engineering & Applications Symposium*, IDEAS'14, pages 67–76. ACM.

J. Valverde-Rebaza and A. Lopes. 2013. Exploiting behaviors of communities of Twitter users for link prediction. *SNAM*, 3(4):1063–1074.

J. Valverde-Rebaza, A. Soriano, L. Berton, M. C. F. de Oliveira, and A. Lopes. 2014. Music genre classification using traditional and relational approaches. In *Proceedings of 2014 Brazillian Conference on Intelligent Systems*, BRACIS 2014, pages 259–264. IEEE.

Text Mining Applied to SQL Queries: A Case Study for the SDSS SkyServer

Vitor Hirota Makiyama
CAP – INPE
São José dos Campos
São Paulo - Brazil
vitor.hirota@gmail.com

M. Jordan Raddick
Physics and Astronomy Dept.
The Johns Hopkins University
Baltimore, Maryland, USA
raddick@jhu.edu

Rafael D. C. Santos
LAC – INPE
São José dos Campos
São Paulo - Brazil
rafael.santos@inpe.br

Abstract

SkyServer, the portal for the Sloan Digital Sky Survey (SDSS) catalog, provides data access tools for astronomers and scientific education. One of the interfaces allows users to enter *ad hoc* SQL statements to query the catalog, and has logged over 280 million queries since 2001. This paper describes text mining techniques and preliminary results on mining the logs of the SQL queries submitted to SkyServer, along with what other applications we foresee for such procedure.

1 Introduction

With the increase in data collection and generation, datasets are growing at an exponential pace, making a real challenge to make available all the data being produced. As a solution, some large scientific datasets have been made available through publicly accessible RDBMSes (Relational Database Management Systems). In which scientists and interested users can query and analyze only the most relevant and up-to-date data for their needs.

The Sloan Digital Survey is one such case. It makes available the largest astronomy survey to date through SkyServer[1], its Internet portal that allows users and astronomers to query the database and even perform data mining tasks using SQL (Standard Query Language), the *de facto* standard to query relational databases. The portal, in operation since 2001, has proven to be extremely popular, with over 1.5 billion page hits and almost 280 million SQL queries submitted.

Since 2003, SkyServer has been logging every query submitted to the portal. It collects access information, such as timestamp, user ip address, the tool used to submit the query, and the target data release (DR1, DR2, etc); and query information, e.g. the SQL statement, query success or failure and error message, number of rows returned, elapsed time. This data can be used to generate summarized access statistics, like queries per month or data release query distribution over time, as presented by Raddick et al. (2014). But for a more in depth usage analysis, data has to be processed and transformed, like Zhang et al. (2012), which color codes SQL queries for visual analysis and also presents a visual sky map of popular searched areas.

To further analyze such queries, this paper aims to apply text mining techniques with the goal to define a procedure to parse, clean and tokenize statements into a weighted numerical representation, which can then be fed into regular machine learning algorithms for data mining.

We proceed with an exploratory analysis, where we project part of the historical queries into a low dimensional representation and correlate the results with sample templates defined in the SkyServer help pages, a list of predefined queries ranging from Basic SQL, showing simple SQL structures; to specific examples on how to find Stars, Galaxies or Quasars.

2 Text Mining and SQL Queries

Text mining, or Knowledge Discovery in Texts (KDT), is an extension to the traditional Knowledge Discovery in Databases (KDD), the nontrivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data (Fayyad et al., 1996), but targeting unstructured or semi-structured data instead of regular databases, such as emails, full-text documents and markup files (e.g., HTML and XML). It is a multidisciplinary field involving, among others, information retrieval and extraction, machine learning, natural language processing, database technology and visualization (Tan, 1999).

[1] http://skyserver.sdss3.org

SQL queries in this context can be seen as mini-documents. As a well defined language, we can leverage the structure provided by the language in order to fine-tune and optimize the preprocessing step of queries to suit the specific cases found. For instance, there is no need for stop words removal, and by analyzing the token type (table name, column, variable, expression, constant, etc) we can perform a different normalization or substitution.

3 Methodology

The methodology followed is the traditional KDD process, comprising the following phases: selection, preprocessing, transformation, data mining, and interpretation/evaluation, with each phase briefly discussed below.

3.1 Selection

For this paper, we used a normalized version of the raw data made available by Raddick et al. (2014) which analyzed a 10-year span of log data (12/2002 to 09/2012), amounting to almost 195 million records and 68 million unique queries.

As a proof-of-concept, we filtered the queries to those coming from the last version of the online SQL search tool (skyserver.sdss3.org), which only allows SELECT statements and has a timeout of 10 minutes. The assumption was to have a dataset with less variance and complexity. This filter also restricted queries with errors and no rows returned, resulting in a final dataset of 1.3 million queries.

3.2 Preprocessing and Transformation

The main objective of the preprocessing phase is to parse the text queries into a *bag-of-words* like representation, but instead of just the set of tokens present in each document, we also keep the count of each token in that statement.

As noted before, we can leverage the fact that SQL is a structured language, by using a proper parser and add a layer of metadata on top of each token. Knowing what kind of token we are processing, we can add specific actions for each token type.

Since SkyServer uses Microsoft SQL Server as its RDBMS, we extended the readily available .NET T-SQL parser library to build a custom one. Other than normalizing case sensitivity, the custom parser also removes constants (strings and numbers), database namespaces, and aliases; substitutes temporary table names, logical and conditional operators for keywords; and qualified each token with the SQL group, e.g. *select, from, where, groupby, orderby*. Substitutions and filters were performed with the intention to remove tokens that are trivial (such as database namespaces) or too specific (such as constants, table aliases, or arithmetic operations), and thus, would be of little contribution in discriminating or grouping each query within the dataset.

An example of the original statement and its normalized version is shown in Figure 1. Figure 2 shows the final feature vector.

```
SELECT p.objid, p.ra, p.dec,
       p.u, p.g, p.r, p.i, p.z,
       platex.plate, s.fiberid,
       s.elodiefeh
FROM   photoobj p,
       dbo.fgetnearbyobjeq(1.62917,
         27.6417, 30) n,
       specobj s, platex
WHERE  p.objid = n.objid
       AND p.objid = s.bestobjid
       AND s.plateid =
         platex.plateid
       AND class = 'star'
       AND p.r >= 14
       AND p.r <= 22.5
       AND p.g >= 15
       AND p.g <= 23
       AND platex.plate = 2803
```

(a) Raw SQL query.

```
select objid ra dec u g r i z
       plate fiberid elodiefeh
from   photoobj fgetnearbyobjeq
       specobj platex
where  objid objid logic objid
       bestobjid logic plateid
       plateid logic class logic
       r logic r logic g logic g
       logic plate
```

(b) Tokenized SQL.

Figure 1: Example of a SQL query and its normalized version. Whitespace is included for readability.

It is important to note that, since the parser is strict, it can only process syntax valid statements.

Lastly, we weight tokens according to its frequency, so the most common or unusual rare tokens are balanced to have more or less con-

select_objid	1
select_ra	1
select_dec	1
select_u	1
select_g	1
select_r	1
select_i	1
select_z	1
select_plate	1
select_fiberid	1
select_elodiefeh	1
from_photoobj	1
from_fgetnearbyobjeq	1
from_specobj	1
from_platex	1
where_objid	3
where_logic	8
where_bestobjid	1
where_plateid	2
where_class	1
where_r	1
where_g	2
where_plate	1

Figure 2: Feature vector.

tribution in its power of discrimination. One of the most popular weighting scheme is the TF*IDF (term frequency times inverse document frequency), which assigns the largest weight to terms that arise with high frequency in individual documents, but are at the same time, relatively rare in the collection as a whole (Salton et al., 1975).

3.3 Data Mining

On a general perspective from data analysis, clustering is the exploratory procedure that organizes a collection of patterns into natural groupings based on a given association measure (Jain et al., 1999). Intuitively, patterns within a cluster are much more alike between each other, while being as different as possible to patterns belonging to a different cluster.

In text mining, clustering can be used to summarize contents of a document collection (Larsen and Aone, 1999). So, with this idea in mind, what kind of summarization could be done over the historic SQL logs and how such summary would compare to the predefined templates? For that, we apply in this paper the Self-Organizing Map (SOM) algorithm.

3.3.1 Self-Organizing Maps

Kohonen's SOM (Kohonen, 2001) is a neural network algorithm that performs unsupervised learning. It implements an orderly mapping of high-dimensional data into a regular low-dimensional grid or matrix, reducing the original data dimension while preserving topological and metric relationships of the data (Kohonen, 1998).

The SOM consist of M units located on a regular grid. The grid is usually one- or two-dimensional, particularly when the objective is to use the SOM for data visualization. Each unit j has a prototype vector $m_j = [m_{j1}, ..., m_{jd}]$ in a location r_j, where d represent the dimension of a data item. The map adjusts to the data by adapting the values of its prototype vectors during the training phase. At each training step t a sample data vector $x_i = [x_{i1}, ..., x_{id}]$ is chosen and the distances between x_i and all the prototype vectors are calculated to obtain the best-matching unit (BMU). Units topologically close to the BMU are then updated, moving their values towards x_i.

Distance calculation between the data vectors and prototypes on the SOM can be calculated using the Euclidean, Cosine or other metrics. The neighborhood considered around the BMU can be circular, square, hexagonal (to determine its shape) and the distance between an unit and the BMU can be weighted by a gaussian or difference-of-gaussians function so units closest to the BMU will be updated with different weights used by units further from it. During training the weights used for updating the units and the size of the neighborhood can change according to several different possible rules.

The algorithm has two interesting characteristics that suggest its use for data visualization: quantization and projection. Quantization refers to the creation of a set of prototype vectors which reproduce the original data set as well as possible, while projection try to find low dimensional coordinates that tries to preserve the distribution from the original high-dimensional data. The SOM algorithm has proved to be especially good at maintain the topology of the original dataset, meaning that if two data samples are close to each other in the grid, they are likely to be close in the original high-dimensional space data (Vesanto, 2002).

These features and the possible variations and parameters of the Self-Organizing Map makes it an interesting tool for exploratory data analysis,

particularly for visualization (Morais et al., 2014; Vesanto, 2002). There are three main categories of SOM applications for data visualization: 1) methods that get an idea of the overall data shape and detect possible cluster structures; 2) methods that analyze the prototype vectors (as representatives of the whole dataset) and 3) methods for analysis of new data samples for classification and novelty detection purposes.

In this paper we use visualization methods related to the second and third categories: the U-Matrix and plotting of existing data samples (in our case, query prototypes or templates) over the U-Matrix. The Unified Distance Matrix (U-Matrix) is one of the most used representations of the trained SOM (Gorricha and Lobo, 2012). It is a visual representation of the SOM to reveal cluster structure of the data set. The approach colors a grid according to the distance from each vector prototype and its neighbors: dark colors are chosen to represent large distances while light colors correspond to proximity in the input space and thus represent clusters.

3.4 Data and Implementation

After preprocessing, the initial 1.3 million selected queries were compressed to 8,477 token sets with 2,103 features. As usual in a text mining context, this dataset is extremely sparse, with only 0.008% non-zero values.

Templates were preprocessed in the same manner as the token sets, also using the same idf weights and scaling factors. Since some templates have more than one version, the 45 selected entries expanded to 51, denoted with a suffix letter to indicate when it is a second or third alternative.

Huang (2008) shows that the Euclidean distance performs poorer than other distances in a text clustering context. Hence, for this paper, we chose the Cosine distance as the metric to find BMUs during the SOM training.

For this paper, we used a 30x30 SOM trained for 45 epochs.

3.5 Analysis

We used two plots for an initial visual analysis, the u-matrix, presented in Figure 3, in which numbers indicate the template id over their respective BMU, and a hitmap scatter plot, presented in Figure 4, in which the size of the circles indicates the number of token sets that elected that prototype its BMU.

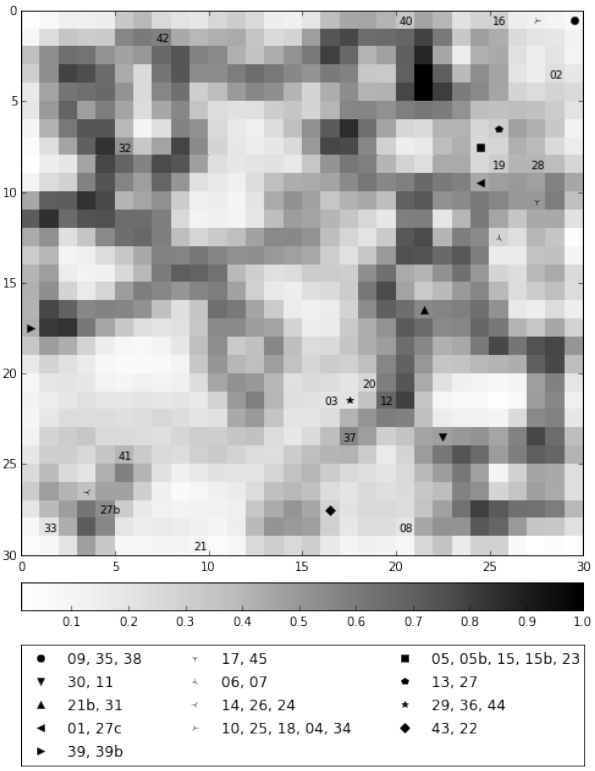


Figure 3: U-Matrix

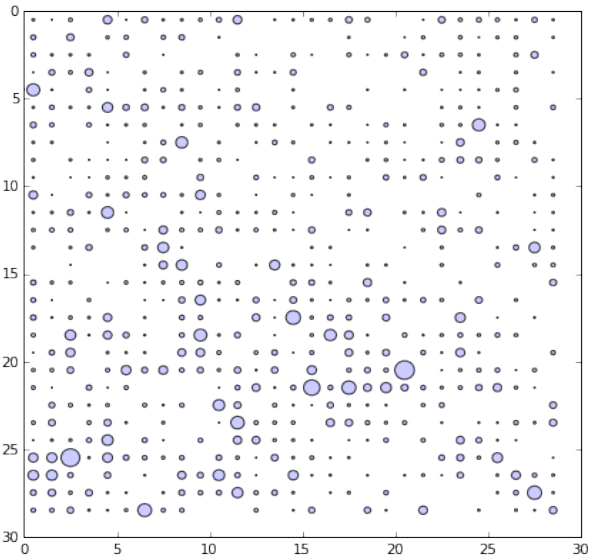


Figure 4: Hitmap

From the figures above, we can see that the trained SOM is able to well distribute the dataset over prototypes and some areas can be visually defined as clusters (regions of light colors circled by dark points).

In some cases, more than one template elected the same prototype as their BMU, as we can check from the legend. So after calculating a distance

matrix, we sorted the top 5 closest templates using the Cosine distance, to see how they compare with the trained SOM.

Below, for each pair, we present their Cosine distance using the Term Frequency representation, and the Euclidean distance between their SOM BMUs, along their name.

1. **Pair:** 15 and 15b
   Distances: TF: 0.0 and SOM: 0.0
   15: Splitting 64-bit values into two 32-bit values
   15b: Splitting 64-bit values into two 32-bit values

2. **Pair:** 21b and 31
   Distances: TF: 0.0 and SOM: 0.0
   21b: Finding objects by their spectral lines
   31: Using the sppLines table

3. **Pair:** 22 and 43
   Distances: TF: 0.0205 and SOM: 0.0
   22: Finding spectra by classification (object type)
   43: QSOs by spectroscopy

4. **Pair:** 39 and 39b
   Distances: TF: 0.1610 and SOM: 0.0
   39: Classifications from Galaxy Zoo
   39b: Classifications from Galaxy Zoo

5. **Pair:** 05 and 15
   Distances: TF: 0.1632 and SOM: 0.0
   05: Rectangular position search
   15: Splitting 64-bit values into two 32-bit values

The SQL queries presented that generated the templates listed here are in the Appendix A.

4 Conclusions and Future Work

As a work in progress, further analysis is definitely due, but from this very early results with the SOM, further work is justified by noticing that close pair of queries are being correctly mapped close to one another.

The Self-Organizing Map was selected as a visualization tool due to its quantization and projection properties. Other methods such as clustering could be used, but preliminary tests showed that the selection of algorithms and parameters is not trivial, and the results were not as useful for exploratory data analysis as the SOM and its visual representations.

Next steps include the evaluation of which queries were similar (but not equal) to a specific template, in order to identify queries that were derived from a template; the analysis of clusters of queries that do not have an associated template, which could uncover possible good candidates for new templates: popular queries that can be included in the list presented in the SkyServer as samples; and finally, the processing of the whole log of queries to build a more comprehensive dataset of the historical logs.

This structured representation can also be correlated with other features in the logs, as elapsed time or error results, allowing other applications of KDD, such as the running time or failure prediction.

Acknowledgments

Vitor Hirota Makiyama was supported by a grant from *Coordenação de Aperfeiçoamento de Pessoal de Nivel Superior* (CAPES).

References

Usama M. Fayyad, Gregory Piatetsky-Shapiro, Padhraic Smyth, and Ramasamy Uthurusamy. 1996. *Advances in Knowledge Discovery and Data Mining*. AAAI Press / The MIT Press.

Jorge Gorricha and Victor Lobo. 2012. Improvements on the visualization of clusters in geo-referenced data using Self-Organizing Maps. *Computers & Geosciences*, 43:177–186.

Anna Huang. 2008. Similarity Measures for Text Document Clustering. In *New Zealand Computer Science Research Student Conference*, pages 49–56.

Anil K. Jain, M. Narasimha Murty, and P. Joseph Flynn. 1999. Data clustering: a review. *ACM computing surveys (CSUR)*, 31(3):264–323.

Teuvo Kohonen. 1998. The self-organizing map. *Neurocomputing*, 21(1):1–6.

Teuvo Kohonen. 2001. *Self-organizing maps*, volume 30. Springer.

Bjornar Larsen and Chinatsu Aone. 1999. Fast and Effective Text Mining Using Linear-Time Document Clustering. In *Proceedings of the 5th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, volume 5, pages 16–22. ACM Press.

Alessandra M. M. Morais, Marcos G. Quiles, and Rafael D. C. Santos. 2014. Icon and Geometric Data Visualization with a Self-Organizing Map Grid. In *Computational Science and Its Applications – ICCSA 2014*, volume 8584 of *Lecture Notes in Computer Science*, pages 562–575. Springer International Publishing.

M. Jordan Raddick, Ani R. Thakar, Alexander S. Szalay, and Rafael D. C. Santos. 2014. Ten Years of SkyServer I: Tracking Web and SQL e-Science Usage. *Computing in Science & Engineering*, 16(4):22–31.

G. Salton, A. Wong, and C. S. Yang. 1975. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620, November.

Ah-Hwee Tan. 1999. Text Mining: The state of the art and the challenges. *Proceedings of the PAKDD 1999 Workshop on Knowledge Disocovery from Advanced Databases*, 8:65–70.

Juha Vesanto. 2002. *Data Exploration Process Based on the Self-Organizing Map*. Ph.D. thesis, Helsinki University of Technology.

G. Vettigli. 2015. MiniSom: minimalistic and NumPy based implementation of the Self Organizing Maps.

Jian Zhang, Chaomei Chen, Michael S. Vogeley, Danny Pan, Ani R. Thakar, and M. Jordan Raddick. 2012. SDSS Log Viewer: visual exploratory analysis of large-volume SQL log data. 8294:82940D.

Appendix A. SkyServer SQL Templates

Sample SQL templates available from SkyServer's help pages that are mentioned in this paper. The list below comprises of the identification number used in the exploratory analysis process, name and category, a brief explanation, and the SQL statement.

05: Rectangular position search (Basic SQL)

Rectangular search using straight coordinate constraints

```
select objid, ra, dec
from photoobj
where (ra between 179.5 and 182.3)
  and (dec between -1.0 and 1.8)
```

15: Splitting 64-bit values into two 32-bit values (SQL Jujitsu)

The flag fields in the SpecObjAll table are 64-bit but some analysis tools only accept 32-bit integers. Here is a way to split them up using bitmasks to extract the higher and lower 32 bits and dividing by a power of 2 to shift bits to the right (since there is no bit shift operator in SQL.)

```
select top 10 objid, ra, dec,
  flags, -- output the whole bigint
      as a check
  flags & 0x00000000ffffffff as
      flags_lo, -- get the lower 32
      bits with a mask shift the
      bigint to the right 32 bits,
      then use the same mask to sget
       upper 32 bits
  (flags/power(cast(2 as bigint),
      32)) & 0x00000000ffffffff as
      flags_hi
from photoobj
```

15B: Splitting 64-bit values into two 32-bit values (SQL Jujitsu)

The hexadecimal version of above query which can be used for debugging

```
select top 10 objid, ra, dec,
  cast(flags as binary(8)) as flags,
  cast(flags & 0x00000000ffffffff as
       binary(8)) as flags_lo,
  cast((flags/power(cast(2 as bigint
      ), 32)) & 0x00000000ffffffff
      as binary(8)) as flags_hi
from photoobj
```

21B: Finding objects by their spectral lines (General Astronomy)

This query selects red stars (spectral type K) with large CaII triplet eq widths with low errors on the CaII triplet equivalent widths.

```
select sl.plate, sl.mjd, sl.fiber,
  sl.caiikside, sl.caiikerr,
  sl.caiikmask, sp.fehadop,
  sp.fehadopunc, sp.fehadopn,
  sp.loggadopn, sp.loggadopunc,
  sp.loggadopn
from spplines as sl
  join sppparams as sp
    on sl.specobjid = sp.specobjid
where fehadop < -3.5
  and fehadopunc between 0.01 and
      0.5
  and fehadopn > 3
```

22: Finding spectra by classification (object type) (General Astronomy)

This sample query find all objects with spectra classified as stars.

```
select top 100 specobjid
from specobj
where class = 'star'
  and zwarning = 0
```

31: Using the sppLines table (Stars)

This sample query selects low metallicity stars ($[\mathrm{Fe/H}] < -3.5$) where more than three different measures of feh are ok and are averaged.

```
select sl.plate, sl.mjd, sl.fiber,
  sl.caiikside, sl.caiikerr,
  sl.caiikmask, sp.fehadop,
  sp.fehadopunc, sp. fehadopn,
  sp.loggadopn, sp.loggadopunc,
  sp.loggadopn
from spplines as sl
  join sppparams as sp
    on sl.specobjid = sp.specobjid
where fehadop < -3.5
  and fehadopunc between 0.01 and
      0.5
  and fehadopn > 3
```

39: Classifications from Galaxy Zoo (Galaxies)

Find the weighted probability that a given galaxy has each of the six morphological classifications.

```
select objid, nvote,
  p_el as elliptical,
  p_cw as spiralclock,
  p_acw as spiralanticlock,
  p_edge as edgeon,
  p_dk as dontknow,
  p_mg as merger
from zoonospec
where objid = 1237656495650570395
```

39B: Classifications from Galaxy Zoo (Galaxies)

Find 100 galaxies that have clean photometry at least 10 Galaxy Zoo volunteer votes and at least an 80% probability of being clockwise spirals.

```
select top 100 g.objid, zns.nvote,
  zns.p_el as elliptical,
  zns.p_cw as spiralclock,
  zns.p_acw as spiralanticlock,
  zns.p_edge as edgeon,
  zns.p_dk as dontknow,
  zns.p_mg as merger
from galaxy as g
  join zoonospec as zns
    on g.objid = zns.objid
where g.clean=1
  and zns.nvote >= 10
  and zns.p_cw > 0.8
```

43: QSOs by spectroscopy (Quasars)

The easiest way to find quasars is by finding objects whose spectra have been classified as quasars. This sample query searches the SpecObj table for the IDs and redshifts of objects with the class column equal to 'QSO'

```
select top 100 specobjid, z
from specobj
where class = 'qso'
  and zwarning = 0
```

Spreader Selection by Community to Maximize Information Diffusion in Social Networks

Didier A. Vega-Oliveros and Lilian Berton
Department of Computer Science
ICMC, University of São Paulo
C.P. 668, CEP 13560-970, São Carlos, SP, Brazil
davo,lberton@icmc.usp.br

Abstract

Rumors or information can spread quickly and reach many users in social networks. Models for understanding, preventing or increasing the diffusion of information are of greatest interest to companies, governments, scientists, etc. In this paper, we propose an approach for maximizing the information diffusion by selecting the most important (central) users from communities. We also analyze the selection of the most central vertices of the network and considered artificial and real social networks, such as *email*, *hamsterster*, *advogato* and *astrophysics*. Experimental results confirmed the improvement of the final fraction of informed individuals by applying the proposed approach.

1 Introduction

The modeling of propagation or diffusion processes in social networks has recently received more attention, since it allows to understand how a disease can be controlled or how information spread among individuals. These diffusion process are generally analyzed employing complex network theory (Barrat et al., 2008; Castellano et al., 2009). The area of complex networks seeks to study and understand the dynamics and behavior of complex systems, from the structure of the network to the internal dynamics or interactions.

Models that describe the evolution of rumors can be adapted to analyze the spread of spam on the Internet, advertising and marketing, political ideologies or technological news between individuals (Castellano et al., 2009). In such cases, the representation in complex networks enables the analysis of traditional models and the heterogeneous structure, which has a strong influence on the information diffusion process (Moreno et al., 2004; Barrat et al., 2008; Castellano et al., 2009). Therefore, some individuals can have a higher influence than others according to the network structure. Researchers have focused on identifying the most influential vertices (Kempe et al., 2003; Kitsak et al., 2010; Lawyer, 2012; Pei and Makse, 2013; Hébert-Dufresne et al., 2013) according to topological properties. It is expected this influencers convince the largest number of individuals in the network. However, the selection of more than one of them not necessarily maximizes the expected fraction of informed individuals, compared to an uniformly random selection approach.

In this paper, we propose an approach to maximize the information diffusion considering the community structure of the network. The community symbolizes a group of individuals with a greater tendency to have more internal than external connections to other groups. The reason is that vertices belonging to the same community are likely to be more similar to each other and share similar properties and affinity. We confirmed that selecting the most influential individual from each community as initial spreaders increases more the information diffusion than selecting the most influential individuals from the whole network.

As a motivation example, let us consider a company that wants to market a new product in the blogosphere. The company could select three very influential individuals of this social network (bloggers with thousands of access) to advertise its product. However, these influencers may be popular in the same group of people. On the other hand, if the strategy is to identify the three main communities on the network and select the most influential individuals of each community, the company would achieve a variety group of users and maximize the marketing diffusion.

The main contribution of this paper is the information diffusion approach based on selecting the most influential individuals from communities.

We employed an artificial scale-free and four real networks: *email*, *hamsterster*, *advogato* and *astrophysics*. We applied the SIR model for rumor propagation selecting the initial seeds from the whole network and from the communities. The impact that the Truncate (TP), Contact (CP) or Reactive (RP) processes have in the information diffusion was analyzed. The experimental results showed that the selection of individuals from the communities as initial spreaders, the final fraction of informed individuals is improved.

The remainder of the paper is organized as follows: Section 2 introduces some definitions and measures covered in this paper, the community detection algorithm applied and the propagation process in networks. Section 3 brings some related work. Section 4 presents the proposed approach for information diffusion based on communities. Section 5 exhibits the experimental results on an artificial scale-free and four real social networks. Finally, Section 6 discusses the final remarks.

2 Theoretical background

A network is a collection of items called nodes or vertices, which are joined together by connections called links or edges. Formally we define the network $G = (V, E, W)$, where $V = \{v_1, v_2, \dots v_n\}$ is the set of N vertices, $E = \{e_1, e_2, \dots e_m\}$ is the set of M edges and $W = \{w_1, w_2, \dots w_m\}$ are the weights of the edges that determine the strength of the interaction between the respective vertices, in the case of weighted networks. In mathematical terms, an undirected and unweighted network can be represented by an adjacency matrix A, in which two connected vertices i and j are in the matrix $a_{ij} = a_{ji} = 1$, otherwise, they are equal to 0.

A path is a consecutive sequence that starts at vertex i and ends in j, so that vertices are visited more than once. The distance or length of the path is defined as the number of edges contained in the sequence. The shortest distance between two vertices is known as the shortest path or geodesic path $g_{i,j}$. A component is the largest sub-set of vertices from the network in which exist at least one path between each pair of vertices, but never connect to another component. A connected network has only one component. When the networks have more than one component, we considered the largest of them.

The degree or connectivity of vertex i, called as k_i, is the number of edges or connections that vertex i has. In the case of directed networks is the sum of the degrees of input (edges that reach the vertex) and output (edges that leave the vertex). The average degree $\langle k \rangle$ is the average of all k_i of the network. The vertices that have a very high degree in the network are called hubs.

The degree distribution of a network $P(k)$ is the probability of randomly select a vertex with degree k. Social networks present scale-free degree distribution (Barabási, 2007; Newman, 2010), with $P(k) \sim k^{-\gamma}$, in which most of the individuals have low degree, near to the average, and only a few of them have very high degree (hubs). The level of disorder or heterogeneity in vertices connections is obtained with the entropy of degree distribution. We employed the normalized version of the Shannon entropy, i.e.

$$\tilde{H} = -\frac{\sum_{k=0}^{\infty} P(k)\log(P(k))}{\log(N)}\,, \qquad (1)$$

with $0 \leq \tilde{H} \leq 1$. The maximum value for the entropy occurs when $P(k)$ shows a uniform distribution and the lowest possible value happens when all vertices have the same degree. The entropy of a network is related to the robustness and their level of resilience.

The robustness is also related to the correlation degree of the network. A network is assortative, or positive correlated, if vertices tend to connect with vertices with similar degree. A Network is dissassortative, or negative correlated, if vertices with low degree tend to connect with higher connected vertices (hubs). When networks do not present any of above patterns, they are called as non-assortative. For the calculation of the network correlation we employed the Pearson coefficient, formulated with adjacency matrix as

$$\rho = \frac{(1/M)\sum_{j>i} k_i k_j a_{ij} - \left[(1/M)\sum_{j>i}\frac{(k_i+k_j)a_{ij}}{2}\right]^2}{(1/M)\sum_{j>i}\frac{\left({k_i}^2+{k_j}^2\right)a_{ij}}{2} - \left[(1/M)\sum_{j>i}\frac{(k_i+k_j)a_{ij}}{2}\right]^2} \qquad (2)$$

where M is the total number of edges in the network. If $r > 0$, then the network is assortative. If $r < 0$ the network is dissassortative. If $r = 0$, then there is no correlation between the degree of vertices.

2.1 Centrality measures

In complex and social networks have been proposed measures to describe the importance or centrality of vertices (Costa et al., 2007) according to topological and dynamical properties. The centralities adopted in this work are briefly described as follow.

Degree centrality (*DG*) is related with the number of connections or popularity of a vertex (Costa et al., 2007) and in terms of the adjacency matrix is expressed as

$$k_i = \sum_{i \in N} a_{ij} \, . \tag{3}$$

Betweenness centrality (*BE*) quantifies the number of shortest paths that pass through a vertex j between all pair of different vertices (i, k) (Freeman, 1977). It expresses how much a vertex B_j works as bridge or is a trusted vertex in the transmission of information, i.e.

$$B_j = \sum_{i,k \in V, i \neq k} \frac{\sigma_{ik}(j)}{\sigma_{ik}} \, , \tag{4}$$

where σ_{ik} is the total number of different shortest path between i and k, and $\sigma_{ik}(j)$ is the number of times j appears in those paths.

PageRank centrality (*PR*) expresses the importance of a vertex according to the probability that other vertices have to arrive at it, after a large number of steps (Brin and Page, 1998). The idea is to simulate the surfing on the net. The user can follow the links available at the current page or jump to other by typing a new URL. In social terms, it can be approached like the more cited or renowned individuals. The formalization of the PageRank centrality is

$$\vec{\pi}^t = \vec{\pi}^{t-1} \mathbb{G} \, , \tag{5}$$

where $\vec{\pi}^t$ are the PageRank values for each vertex in the t^{th} step of navigation and $\mathbb{G}$ is known as the Google matrix. When $t = 0$ we have by default $\vec{\pi}^0 = \{1, \ldots, 1\}$. The jumps are represented by a probability α and we adopted the same value as defined in the original version (Brin and Page, 1998).

2.2 Community detection

Communities are sets of densely interconnected vertices and sparsely connected with the rest of the network (Newman, 2010). Vertices that belong to the same community, in general, share common properties and perform similar roles. Therefore, the division of a network into communities helps to understand their topological structure (structural and functional properties) and its dynamic processes, obtaining relevant information and features to the network domain.

We can evaluate a partition based on the scores obtained from a quality measure. The goal is to evaluate expected features in a good community division. One of the most popular quality measures is the modularity Q (Newman, 2004). It compares the current density of intra-community and inter-community edges relative to a random network with similar characteristics. It is based on the fact that random networks have no community structure.

Given a network with c communities, the Q modularity is calculated by a symmetric matrix $N \times N$, in which elements along the main diagonal e_{ii} represent connections into the same community and elements e_{ij} represent connections between different communities i and j. Equation 6 shows the formulation of Q.

$$Q = \sum_i \left[e_{ij} - \left(\sum_j e_{ij} \right)^2 \right] \tag{6}$$

If a specific division provides less edges between communities than would be expected by random connections, the value of Q would be 0. When the network has isolated communities the value of Q would be 1. This measure is employed by several techniques to identify communities in networks systems, especially in divisive and agglomerative methods (Guimera et al., 2003; Newman, 2004; Newman, 2006).

Newman (Newman, 2004) proposes an agglomerative method that is an optimized greedy algorithm, called *fastgreedy*. The approach starts with a copy of a real network of N vertices with no connections, producing at first N communities. At each iteration, two communities c_i and c_j, which have connections in real network, are chosen in order to obtain the greatest improvement of Q (Equation 7). A pruning is performed in the search space considering only the edges that exist between communities. Therefore, execution time is reduced when considering the new Q function (Equation 7).

$$\Delta Q_{ij} = 2 \left(e_{ij} - \frac{\sum_j e_{ij} \sum_i e_{ij}}{2M} \right) \tag{7}$$

The result can also be represented as a dendrogram. Cuts at different levels of the dendrogram produce divisions with greater or lesser number of communities, and the best cut yields the largest value of Q. The algorithm at each step has $O(M + N)$. Since there are at most $N - 1$ join operations required to build a complete dendrogram, their overall complexity is $O((M + N)N)$, or $O(N^2)$, for sparse graph. Consequently, by adopting this method is more treatable the analysis of communities in larger networks.

2.3 Propagation process on networks

In classical rumor diffusion models the ignorant or inactive individuals (S) are those who remain unaware of the information, the spreaders (I) are those who disseminate the information and the stifler (R) are those who know the information but lose the interest in spreading it. All vertices have the same probability β for transmit the information to their neighbors and μ for stopping to be active.

The Maki-Thompson (MT) (Maki and Thompson, 1973) model is a spreader-centric approach employed for describing the propagation of ideas and rumors on networks. In the MT process whenever an active spreader i contacts a vertex j that is inactive, the latter will become active with a fixed probability β. Otherwise, in the case that j knows about the rumor, it means j is an active spreader or a stifler, the vertex i turns into a stifler with probability μ. The behavior when the spreader stops to propagate can be understood because the information is too much known (contacting spreaders) or without novelty (contacting stifler).

Three possible choices related with the spreader behavior during the diffusion have been reported (Borge-Holthoefer et al., 2012; Meloni et al., 2012). They are the Reactive process (RP), Truncated process (TP) and Contact Process (CP). However, a clear analysis about the impact of spreaders behavior in the propagation process has not been tackled yet. Moreover, there is not a consensus about what to employ in rumor or information diffusion and it may cause a misinterpretation of results. The three main characteristic behaviors reported to spreaders are described as follow.

- Reactive Process (RP): In each iteration, the spreaders try to pass the rumor among all their ignorant neighbors. After that, it evaluates whether it will become stifler in the next iteration or not, considering the contact with all their spreader and stifler neighbors.
- Truncated Process (TP): It consists of truncate or interrupt the contagion in the precise time. In each iteration and for each spreader, it is randomly selected one neighbor at time, and setting up the states of the contact as corresponds. The selection continues until the spreader visit all its neighbors or it becomes stifler, whichever occurs first.
- Contact Process (CP): In each time step and for each spreader, it is chosen at random a single neighbor. Then, it is resolved the transition states according to the rule that corresponds. After that, continues with the next spreader of the network of the same time step.

Different theoretical models have been proposed for modeling the rumor dynamics on networks (Moreno et al., 2004; Barrat et al., 2008; Castellano et al., 2009; Borge-Holthoefer et al., 2012). These analytical models make assumptions about the network structure, such as the degree correlation or distribution, compartments or class of vertices with same probabilities, homogeneous/heterogeneous mixing or mean field theory. Notwithstanding all of them claim that their numerical solutions agree with the MC simulations, so we adopt this approach as an exploratory research.

3 Related work

Many approaches have been developed in order to understand the propagation of ideas or information through social networks (Castellano et al., 2009). Specially, characterize the individuals that are most influential in the propagation process has attracted the attention of researchers (Richardson and Domingos, 2002; Kempe et al., 2003; Kitsak et al., 2010; Pei and Makse, 2013).

The conventional approach for describing the most influential vertices is performing a microscopic analysis on the network. Vertices are classified considering their topological properties, sorted and ranked in order to generalize their ability to propagate (Kitsak et al., 2010; Hébert-Dufresne et al., 2013; Pei and Makse, 2013). However, to find the set of initial vertices that maximize the propagation capacity, the selection of the most influential spreader may produce an overlap of influence in the population (Kitsak et al., 2010; Pei and Makse, 2013).

In terms of topological properties, there not exists a consensus about what is the more accurate measure that describes the most influential vertices. Some researches claim that hubs are more representative to influence others vertices (Pastor-Satorras and Vespignani, 2001; Albert and Barabási, 2002). Vertices with higher degree are more efficient to maximize the propagation because, in general, hubs not tend to connect with each other and thus can achieve a greater number of vertices (Kitsak et al., 2010). In the case of communities, the degree proportion of a vertex i is defined as the number of edges that i has in each community. This degree proportion was found as a good descriptor of influence for communities (Lawyer, 2012).

On the other hand, the most influential vertices are described as those with the largest Betweenness centrality (Hébert-Dufresne et al., 2013), because they intermediate the communication between groups of vertices, which increase their influence. According to the authors, Betweenness centrality is a better descriptor of the most influential spreader in communities.

The PageRank is also considered a better measure to describe the most influential vertices (Cataldi et al., 2010). The reason is that it employs the random walk concept over the network to be calculated and vertices with higher values mean higher probability to be visited.

Finally, Kempe et al. (2003) propose a greedy algorithm to obtain η initial spreaders that maximizes the diffusion influence. The authors adopt a discrete optimization approach and prove that the optimization problem is NP-hard. It was implemented considering the independent and weighted cascade model that have only two states, which are different to the SIR model. The method evaluates one vertex at time to be added in the set of selected seeds. The new vertex is accepted if it is what most increment the diffusion. However, this approach has a very higher computational cost problem, although new researches try to optimize the performance (Chen et al., 2009).

4 Information diffusion by communities

Let us consider a constant population of N vertices in all time steps. Each vertex can be only in one state, that is $I_i(t) = 1$ *iff* $i \in I$, otherwise $I_i(t) = 0$, and $S_i(t)+I_i(t)+R_i(i) = 1$. The macroscopic fraction of ignorant ($\psi(t)$), spreaders ($\phi(t)$) and stifler ($\varphi(t)$) over time is calculated as $\psi(t) = \frac{1}{N}\sum_{i=1}^{N} S_i(t)$, that is similar to the other states and always fulfill $\psi(t) + \phi(t) + \varphi(t) = 1$. We assume that infection and recovering do not occur during the same discrete time window or step.

4.1 Setup

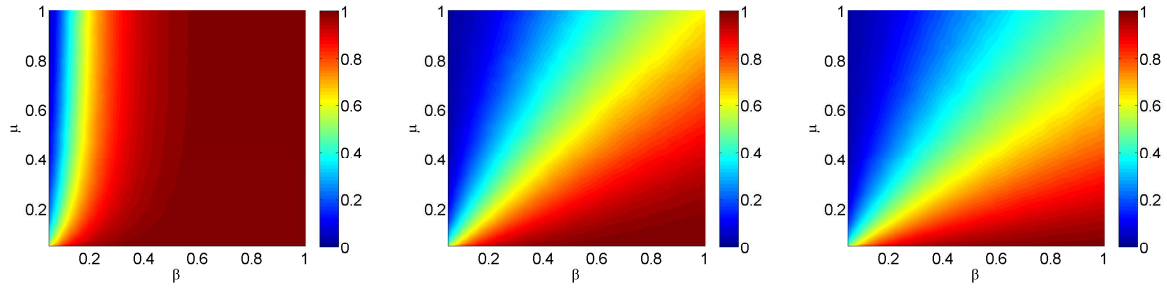
The initial setup for the propagation is $\psi(0) = 1 - \eta/N$, $\phi(0) = \eta/N$ and $\varphi(0) = 0$, where η represents the seeds or number of initial spreaders. Each simulation begins with a selection of η vertices. At each time step, all spreaders uniformly select and try to infect its neighbors with probability β, or stop the diffusion with probability μ according to the spreader behavior adopted. Successful change of state (to be spreader or to be stifler) are effective at the next iteration. The simulations run until the end of the process is reached, when $\phi(\infty) = 0$.

4.2 Community selection approach

We propose to select the initial spreaders from the community division of the network. The multiple seeds are the most central vertices of each community. The community division may be calculated by some divisive or agglomerative method (Section 2.2) and here the *fastgreedy* algorithm was employed. The method is detailed as follow:

First, given a required number of η initial spreaders, we find the η main communities of the network by the *fastgreedy* algorithm. Then, each community is isolated, which produces η components. The isolation process consists in maintaining the intra-community edges and erasing the inter-community connections. For each isolated community, a specific centrality measure is calculated to all vertices. Since vertices with higher centrality are considered more suitable to influence on the network, we select the most important vertex from each community. Therefore, these vertices influence more in their own community and the overlap of influence in the population is minimized. At the end, η seeds are selected and they have the best centrality value of its community. We take the original full network, the η seeds, the parameters and execute the corresponding simulations.

For the centrality measure, the point is to find what centrality better identifies the influential spreaders, by communities and in the whole network, that maximizes the information diffusion.



(a) MT-RP (b) MT-TP (c) MT-CP

Figure 1: MT (Maki and Thompson, 1973) propagation in an artificial scale-free network with $N = 1000$, $\langle k \rangle = 8$ and $\eta = 1\%$ of initial seeds selected. The color bar shows the final fraction of informed individuals. The behavioral approaches for spreader analyzed are: (a) Reactive process (RP); (b) Truncated process (TP); and (c) Contact process (CP)

Here, the degree (*DG*), PageRank (*PR*) and Betweenness (*BE*) centralities were considered.

5 Experimental results

In this section we analyzed the information diffusion in an artificial scale-free and four real social networks. We evaluated the impact spreaders behavior have in the diffusion on the networks. Then, the results about the selection of initial spreaders by communities, best-ranked vertices of the network and random seeds were explored.

5.1 Spreader behavior analysis

We analyzed the three behavioral approaches for the spreaders and present the impact they produce on the propagation process. We considered the MT model with an artificial scale-free network of size $N = 1000$ and $\langle k \rangle = 8$. In order to understand the overall spectral effect with the parameters, the simulations were evaluating a range of β and μ in $(0, 1]$. Therefore, the differences between the approaches are evidenced. For each tuple of values (β, μ), it was selected 100 times at random $\eta = 1\%$ of initial spreader (seeds) and each time was an average over 50 executions.

The impact of the behavioral approach in the final fraction of informed individuals is shown in Figure 1. We observed that the CP approach is less redundant in the number of contacts made by spreader, producing lower fractions of informed individuals, in comparison to the other behaviors. Still, because the single contact made by iteration, the CP behavior is more similar to a propagation through the "word-of-mouth" situation.

The RP approach obtained more than 80% of informed individuals with values of $\beta \geq 0.3$, no matter values of μ. Therefore, the RP approach favors a viral diffusion on the network with lower values of β and it happens independently of which are the initial seeds. For this reason, RP is a more suitable approach to simulate broadcasting propagation.

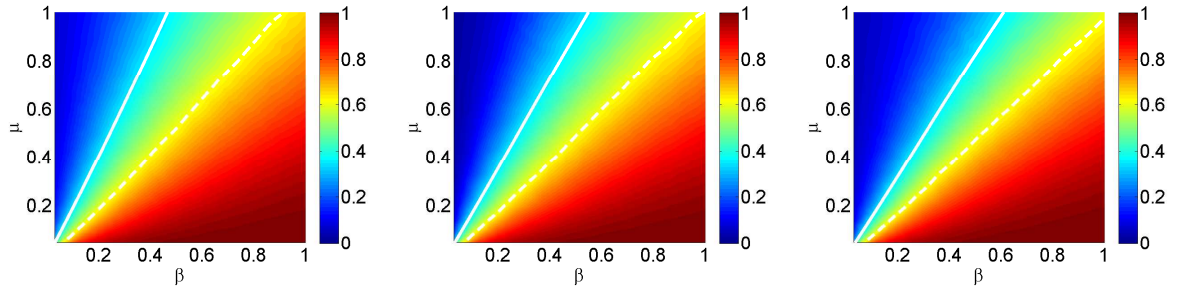
On the other hand, the TP behavior is more related to the contact network scenario, where the position and topological characteristics of seeds may have influence in the diffusion. Moreover, TP presents more balanced results, near 60% when $\beta \approx \mu$, and contacts are not as restricted as CP behavior. For this reason, we adopted hereafter the MT-TP approach as the propagation process for the analysis.

5.2 Multiple initial spreader analysis

The experiments were performed with three possibilities for choosing the initial spreader: (*i*) by randomly selecting η individuals as initial seeds in the network; (*ii*) by selecting the best- ranked η individuals with highest value of a specific centrality of the network; and finally, (*iii*) by detecting η communities on the network and for each isolated community selecting the individual with highest value of a specific centrality measure. The centrality measures selected were degree (*DG*), Betweenness (*BE*) and Pagerank (*PR*).

5.2.1 Real social networks

We adopted the *email* (Guimera et al., 2003), *advogato* (Kunegis, 2014a), *astrophysics* (Newman, 2001) and *hamsterster* (Kunegis, 2013; Kunegis, 2014b). All of them were assumed as undirected and unweighted networks and also it was considered the largest component for the simulations. The structural properties of the networks are summarized in Table 1, with the respective number of vertices N, the average degree $\langle k \rangle$, shortest paths



(a) MT-TP Community (b) MT-TP Random (c) MT-TP Best

Figure 2: MT-TP propagation in an artificial scale-free network with $N = 1000$ and $\langle k \rangle = 8$. The final fraction of informed individuals are shown in the color bar. The selection of $\eta = 4\%$ of seeds was made by: (a) η communities taking the individual with best PR centrality of each one; (b) uniform random selection of individuals; and (c) the η individuals with the best PR centrality of the network. Solid white lines to the left in the contour plots show the β and μ combinations that achieved 35% of informed individuals. Dashed white lines show the combinations that achieved 60% of informed individuals.

Table 1: Topological properties and results of community detection of the networks: last column, the best modularity Q and community division by fastgreedy algorithm

Network	N	$\langle k \rangle$	$\langle g \rangle$	$\tilde{H}$	ρ	FastGreedy	
						Q	Nc
email	1133	9.62	3.60	0.45	0.01	0.49	16
hamsterster	2000	16.1	3.58	0.48	0.02	0.46	57
advogato	5054	15.6	3.27	0.40	-0.09	0.34	49
astrophysics	14845	16.1	4.79	0.38	0.23	0.63	1172

average $\langle g \rangle$, normalized entropy $\tilde{H}$, pearson correlation ρ. Also, the best modularity Q value and division number of communities NC of the networks produced by the FastGreedy algorithm are reported.

email represents a social network of information exchanged by emails between members of the *Rovira i Virgili* University, Tarragona, with largest hub degree equal to 71.

hamsterster is an undirected and unweighted network based on the website data HAMSTERSTER.COM. The edges represent a relationship of family or friend among users. The largest hub has degree equal to 273.

advogato is an online community platform for developers of free software launched in 1999. Vertices are users of advogato, the directed edges represent trust relationships. The largest hub has degree equal to 807.

Finally, *astrophysics* is a collaborative network between scientists on previous studies of astrophysics reported in arXiv during January 1, 1995 until December 31, 1999. The network is weighted and directed and originally it has 16707 vertices. The largest hub of the main component has 360 connections.

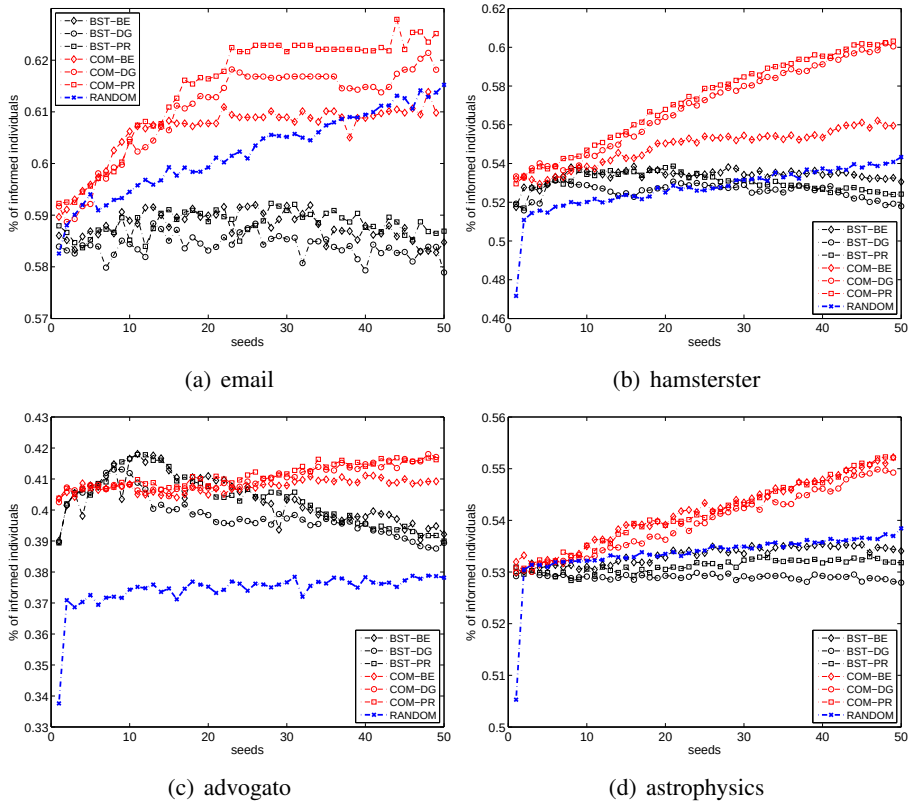
5.2.2 Information diffusion results

The final fraction of informed individuals ($\varphi(\infty)$) was averaged over 100 executions for each combination of initial seeds and parameters. This average represents the propagation capacity achieved by the selected seeds.

We evaluated the relation between the parameters and the selection of the initial spreaders in an artificial network. In this experiment the *PR* was defined as the centrality measure employed to find the seeds in the communities and the whole network. A value of $\eta = 4\%$ of initial spreaders was adopted for a scale-free network of size $N = 1000$, $\langle k \rangle = 8$, $\langle g \rangle = 3.19$, $\tilde{H} = 0.33$ and $\rho = -0.04$.

The propagation capacity $\varphi(\infty)$ was affected according to the initial seeds (Figure 2). The solid and dashed white curves represent the combination of β and μ parameters that obtained 35% and 60% of informed individuals respectively. We observed that these curves show a well defined linear pattern, which means any proportion of $\lambda = \beta/\mu$ will obtain equivalent $\varphi(\infty)$ results.

The selection of seeds by communities (Figure 2(a)) improved the diffusion on the network in comparison with the Random seeds (Figure 2(b)) and Best-ranked vertices (Figure 2(c)). This result is corroborated by the increase of the white lines



(a) email (b) hamsterster

(c) advogato (d) astrophysics

Figure 3: Propagation capacity of MT-TP model in the four real networks given the selection of seeds by communities (red points), best-ranked (black points), and randomly (blue points).

Table 2: Average of propagation capacities for the full range of $\eta \in [1, 50]$, for each network achieved by seeds: (second big column) selecting the most important individuals from communities; (third big column) selecting the best-ranked individuals of the network; and (last column) randomly selecting the initial seeds. The adopted measures were Betweenness (*BE*), degree (*DG*) and PageRank (*PR*) centralities

Network	Community			Best ranked			Random
	BE	*DG*	*PR*	*BE*	*DG*	*PR*	selection
email	0.6065	0.6105	**0.6150**	0.5880	0.5840	**0.5884**	0.6023
hamsterster	0.5485	0.5693	**0.5728**	**0.5306**	0.5226	0.5271	0.5273
advogato	0.4077	0.4102	**0.4112**	0.3993	0.3958	**0.4007**	0.3805
astrophysics	**0.5417**	0.5398	0.5415	**0.5321**	0.5278	0.5301	0.5337

slope. However, a little decrease in the lines slope is evidenced in the *MT-TP Best* case with respect to the *MT-TP Random* case.

Consequently, we sought to analyze the impact of η and centrality measures in the selection of seeds in the diffusion process. We varied the number of communities and seeds from 2 to 50 and fixed $\beta = 0.3$ and $\mu = 0.2$ for all simulations. The real social networks described and the MT-TP propagation model were considered in the analysis (Figure 3). The random selection of initial spreader (blue points, *RANDOM*) or best-ranked vertices (black points, *BST-***) of *DG*, *BE* or *PR* centrality, produced a constant propagation capacity ($\varphi(\infty)$). In some case, random selection of seeds reached a higher propagation capacity than the selection of best-ranked vertices. For a larger number of initial spreaders, $\varphi(\infty)$ tend to fall when the best-ranked vertices are selected.

On the other hand, when the community detection was performed and individuals with highest values of *DG*, *BE* or *PR* in each community (red points, *COM-***) were selected, the propagation capacity was improved and achieved the best results. Therefore, more individuals were informed in the network by the community selection, with the same propagation constraint (number of seeds).

In terms of the topological measures, we observed that vertices with highest PageRank cen-

trality in the communities (*COM-PR*) obtained in average the best propagation results (Table 2). Even in the selection of the best-ranked vertices, the PageRank was notable. Another important point is that often, the uniformly random selection of initial spreader could be a better option than select the most central vertices (best-ranked) of the network. This is contrary what is currently expected and adopted in marketing campaigns, for instances. For all networks and for all size of seeds, we evidenced that starts the diffusion from the best-ranked vertices produces lower influence, or final fraction of informed individuals, than purely select vertices at random; in some cases, the best-ranked selection achieved the worst results. However, the selection of initial spreaders by communities showed, independently of the centrality measure, higher results.

6 Final remarks

In this work, we proposed a method for maximizing the information diffusion on networks. First, we analyzed the impact of the spreader behavior in the propagation and confirmed that the Truncate Process (TP) is more suitable to simulate information diffusion on networks. We applied community detection and targeted the most influential vertices from these communities as initial seeds. Experimental results on an artificial scale-free and four real social networks confirmed the increase in the final fraction of informed individuals. Moreover, it was found that the PageRank centrality in communities was a better choice in terms of efficiency and influence maximization.

A brief overview about complex network measures, community detection and information propagation was introduced. We present our proposal to select initial spreaders by communities. There is still an open problem related to an exact definition of what is considered a community and what would be the ideal division. Nevertheless, we varied the number of communities from 2 to 50 and in general (for every community division) our proposal achieved better results versus propagation without considering the community structure.

In future work, other measures for selecting influential individuals on networks could be explored, in addition to *DG*, *BE* and *PR* applied here. Also, other models of propagation and network topologies could be tested, as well as novel strategies taking into account community information.

7 Acknowledgments

This research was partially supported by National Council for Scientific and Technological Development (CNPq) grant: 140688/2013-7 and São Paulo Research Foundation (FAPESP) grant: 2011/21880-3.

References

Réka Albert and Albert-László Barabási. 2002. Statistical mechanics of complex networks. *Rev. Mod. Phys.*, 74(1):47–97, jan.

A.-L. Barabási. 2007. The architecture of complexity: From network structure to human dynamics. *IEEE Control Systems Magazine*, 27(4):33–42.

Alain Barrat, MarseilleMarc Barthélemy, and Alessandro Vespignani. 2008. *Dynamical Processes on Complex Networks*. Cambridge University Press.

Javier Borge-Holthoefer, Sandro Meloni, Bruno Gonçalves, and Yamir Moreno. 2012. Emergence of Influential Spreaders in Modified Rumor Models. *Journal of Statistical Physics*, 151(1-2):383–393, September.

S. Brin and L. Page. 1998. The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems*, V:107–117.

Claudio Castellano, Santo Fortunato, and Vittorio Loreto. 2009. Statistical Physics of Social Dynamics. *Reviews of Modern Physics*, 81(2):591–646, may.

Mario Cataldi, Luigi Di Caro, and Claudio Schifanella. 2010. Emerging topic detection on Twitter based on temporal and social terms evaluation. In *Proceedings of the Tenth International Workshop on Multimedia Data Mining - MDMKDD '10*, pages 1–10, New York, New York, USA, jul. ACM Press.

Wei Chen, Yajun Wang, and Siyu Yang. 2009. Efficient influence maximization in social networks. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '09*, page 199, New York, New York, USA, jun. ACM Press.

L. D. F. Costa, F. A. Rodrigues, G Travieso, and P. R. Villas Boas. 2007. Characterization of complex networks: A survey of measurements. *Advances in Physics*, 56:167–242.

L C Freeman. 1977. A set of measures of centrality based on betweenness. *Sociometry*, 40:35–41.

R Guimera, L Danon, A Diaz-Guilera, F Giralt, and A Arenas. 2003. Self-similar community structure in a network of human interactions. *Physical Review E*, 68:2003.

Laurent Hébert-Dufresne, Antoine Allard, Jean-Gabriel Young, and Louis J Dubé. 2013. Global efficiency of local immunization on complex networks. *Scientific reports*, 3:2171, January.

David Kempe, Jon M. Kleinberg, and Éva Tardos. 2003. Maximizing the Spread of Influence Through a Social Network. In Lise Getoor, Ted E. Senator, Pedro Domingos, and Christos Faloutsos, editors, *KDD*, pages 137–146. ACM.

Maksim Kitsak, Lazaros K. Gallos, Shlomo Havlin, Fredrik Liljeros, Lev Muchnik, H. Eugene Stanley, and Hernán A. Makse. 2010. Identification of influential spreaders in complex networks. *Nature Physics*, 6(11):888–893, August.

Jrme Kunegis. 2013. KONECT – The Koblenz Network Collection. In *Proc. Int. Web Observatory Workshop*, pages 1343–1350.

Jrme Kunegis. 2014a. Advogato network dataset – KONECT, October.

Jrme Kunegis. 2014b. Hamsterster full network dataset – KONECT, jan.

Glenn Lawyer. 2012. Measuring node spreading power by expected cluster degree. page 4, September.

D. P. Maki and M Thompson. 1973. *Mathematical Models and Applications, with Emphasis on the Social, Life, and Management Sciences*. Prentice-Hall.

Sandro Meloni, Alex Arenas, Sergio Gmez, Javier Borge-Holthoefer, and Yamir Moreno. 2012. Modeling epidemic spreading in complex networks: Concurrency and traffic. In My T. Thai and Panos M. Pardalos, editors, *Handbook of Optimization in Complex Networks*, Springer Optimization and Its Applications, pages 435–462. Springer US.

Yamir Moreno, Maziar Nekovee, and Amalio F. Pacheco. 2004. Dynamics of rumor spreading in complex networks. *Physical Review E*, 69(6):066130, jun.

M. E. J. Newman. 2001. The structure of scientific collaboration networks. In *Natl. Acad. Sci. USA*, number 98, pages 404 – 409.

M E J Newman. 2004. Fast algorithm for detecting community structure in networks. *Physical Review E*, 69(3):66133.

M E J Newman. 2006. Finding community structure in networks using the eigenvectors of matrices. *Physical Review E*, 74(3):36104.

Mark Newman. 2010. *Networks: An Introduction*. Oxford University Press, Inc., New York, NY, USA.

Romualdo Pastor-Satorras and Alessandro Vespignani. 2001. Epidemic Spreading in Scale-Free Networks. *Physical Review Letters*, 86(14):3200–3203, April.

Sen Pei and Hernán A Makse. 2013. Spreading dynamics in complex networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2013(12):P12002, December.

Matthew Richardson and Pedro Domingos. 2002. Mining knowledge-sharing sites for viral marketing. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '02*, page 61, New York, New York, USA, jul. ACM Press.

Behavior of Symptoms on Twitter

Dennis Salcedo
El Bosque University/Bogotá-Colombia
dsalcedop@unbosque.edu.co

Alejandro León
El Bosque University/Bogotá-Colombia
alejandroleon@unbosque.edu.co

Abstract

With the amount of data available on social networks, new methodologies for the analysis of information are needed. Some methods allow the users to combine different types of data in order to extract relevant information.

In this context, the present paper shows the application of a model via a platform in order to group together information generated by Twitter users, thus facilitating the detection of trends and data related to particular symptoms. In order to implement the model, an analyzing tool that uses the Levenshtein distance was developed, to determine exactly what is required to convert a text into the following texts: 'gripa'-"flu", "dolor de cabeza"-"headache", 'dolor de estomago'-"stomachache", 'fiebre'-"fever" and 'tos'-"cough" in the area of Bogotá. Among the information collected, identifiable patterns emerged for each one of the texts.

1 Introduction

Social networks are important because of their user's opinions on diverse topics (Martos E 2010 and Soumen C 2003). Gathering, processing and analyzing those opinions is an important factor for making decisions, therefore, the study or analysis of mass opinion through social networks is an issue that has emerged as a key methodology in modern sciences (Linto C 2006) – such as psychology (Daniel T. Gilbert, Susan T. Fizke, Gardner L 1998) and economy (Ana S 2003), among many others – because it has an impact on the content generated by users (Robin B, Jonathan G, Andreas H, Robert J 2001).

Therefore, a characteristic Levenshtein distance analyzer, linking with diagrams of relationship and feeling to see how the information is behaving was needed. The result provided a close approach to the people who tweeted with a negative attitude to the symptoms.

The importance of conducting an analysis of information and structure for symptoms is important for the study of data mining and big data. This means, it can be determined how many users posting a tweet are actually sick.

2 Information and Levenshtein Distance

The information is collected using a python script in which the Twitter and json libraries are used. In order to gather tweets associated to a city, they need to be linked to a city code by using the platform of coordinates, GeoPlanet (Willi S 2010).

This way, it is possible to find all the tweets in the city of Bogotá that contains the associated symptoms. A basic algorithm used removes information such as special characters and blank spaces in the thread. Additionally, it gets a portion of the thread to perform this analysis on the desired patterns; the resulting thread is built with a maximum of 4 words.

The Levenshtein Distance shows the number of operations that you need in a thread to finish another one (Vladimir I 1965). It was used because of the simplicity of the algorithm but not for his efficiency. It is noted that there are similarities in the number of operations performed to obtain the desired pattern (Fig. 1).

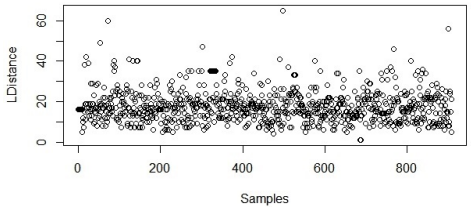


Figure 1: Levensthein Distance applied to the corpus that contains flu.

Therefore relevant information showing the pattern associated with the symptom to improve the filter using the analyzer is taken.

3 Experimentation

Clustering techniques are applied to the gathered information to determine how the information is behaving in two main aspects relationship of concepts and analysis of feeling. The relationship of concepts is established in order to show the relationship of the symptom in the information, by totaling the number of repetitions of the concepts (Fig. 2). The analysis of feeling is applied with dictionaries of positive and negative concepts, giving it a score in the remaining of the same one. Note that there is a pattern in the graph clustering associated with this symptom.

The symptom is regarded as high priority in the corpus (Jurgita M 2002), Therefore, the tree diagram or a relation of concepts are performed for observing the most relevant key points within the information.

During the experiment, a part of the information was used to primarily watch the junction of concepts you can see if the concepts have a positive or negative order in relation to the symptom. The diagram shows the relationship of the concepts:

Figure 2: Combination of concepts 'Gripa'-'Alejate'-"Move away" to describe the relationship between concepts. (Martin W and Fernanda B. Viégas 2008)

The relationship of concepts clearly shows that the symptom is associated with negative concepts; this is done in order to perform an analysis of feeling to the information.

The analysis of feeling performs a score to more traits of the corpus and notes that there are more negative comments than positive ones; this means that there is a high possibility that the person who posts a tweet is sick.

4 Conclusions

Performing an analysis of mass information requires a large amount of data processing and cleaning. The content and the intensity of the information are key factors in determining the maximum percentage of consistency.

The Levenshtein distance and the sentiment analysis can be considered an approach to extract relevant information about symptoms.

Future work will concentrate on obtaining more details on how to improve extraction and cleaning of the corpus to obtain accurate and clear results. The analysis of information with techniques such as artificial intelligence, machine learning and data mining helps to study the information in detail as well as in the decision making for that data set.

We need more work to indicate through the time how these symptoms spread in the city and what are the areas with more reported cases.

The common terms for the symptoms work but we must find a way to relate medical classifications to them.

References

Martos E Carrión. 2010. *Analyzes of new forms of communication via virtual communities or social networks*,Valencia, Spain,(1):2

Soumen C. 2003. *Mining the Web*. San Francisco, California.

Linto C Freeman. 2006. *The Development of Social Network Analysis*. Vancouver BC, Canada.

Daniel T. Gilbert, Susan T. Fizke, Gardner L. 1998. *The handbook of social psychology*, (2):226-227.

Ana S Garcia. 2003. *Social networks as a tool for structural analysis input-output*. University of Oviedo, Department of Economics.

Robin B, Jonathan G, Andreas H, Robert J. 2001. *Recommendation in the Social Web*.

Willi S 2010. *Free geo data solutions compared: GeoNames.org vs. Yahoo! GeoPlanet*.

Vladimir I Levenshtein 1965. *Binary codes capable of correcting deletions, incursions, and reversals*. Doklady Soviet Physics 10 (8): 707-710

Jurgita M 2002. *STUDIES ABOUT LANGUAGES No:3*.

Martin W and Fernanda B. 2008. *The Word Tree, an Interactive Visual Concordance*. Visualization and Computer Graphics, IEEE Transactions on (Volume:14 , Issue: 6)

Scalability Potential of BWA DNA Mapping Algorithm on Apache Spark

Zaid Al-Ars **Hamid Mushtaq**
Computer Engineering Lab, Delft University of Technology
2628 CD Delft, The Netherlands
E-mail: z.al-ars@tudelft.nl

Abstract

This paper analyzes the scalability potential of embarrassingly parallel genomics applications using the Apache Spark big data framework and compares their performance with native implementations as well as with Apache Hadoop scalability. The paper uses the BWA DNA mapping algorithm as an example due to its good scalability characteristics and due to the large data files it uses as input. Results show that simultaneous multithreading improves the performance of BWA for all systems, increasing performance by up to 87% for Spark on Power7 with 80 threads as compared to 16 threads (# of physical cores). In addition, Hadoop has slightly better performance of up to 17% for low system utilization, while Spark has up to 27% better performance for high system utilization. Furthermore, Spark is able to sustain high performance when the system is over-utilized, while the performance decreases for Hadoop as well as the native implementation.

1 Introduction

With the fast increase in the sizes of genomics datasets and the growing throughput of DNA sequencing machines, there is an urgent need to develop scalable, high-performance computational solutions to address these challenges. A number of different approaches are being investigated as possible solutions, ranging from highly connected, customized server-based solutions (Kelly15) to Hadoop-based big data infrastructures (Decap15). Predominantly, however, classical computer cluster-based solutions are the most widely used computational approach, either used locally or in the cloud (Stein10).

Each of these solutions has advantages and disadvantages as it relates to the scalability potential on large computer infrastructures. Customized solutions are expensive to design, but have the advantage of being highly optimized for the specific analysis pipeline being performed. Classical cluster-based solutions are more generic making them less costly, but also less effective. Genomics analysis problems, however, have the potential of offering a huge amounts of parallelism by segmenting the large input datasets used in the analysis. This creates the opportunity of using recent big data solutions to address these analysis pipelines.

In this paper, we investigate the scalability potential of using Apache Spark to genomics problems and compare its performance to both the native scalable implementation developed for classical clusters, as well as to Hadoop-based big data solutions (Zaharia12). This analysis is performed using a popular DNA mapping algorithm called the Burrow-Wheeler Aligner (BWA-MEM), which is known for its speed and high scalability potential on classical clusters (Li13).

This paper is organized as follows. Section 2 discusses typical genomics analysis pipelines and the different stages of the analysis. Section 3 presents the BWA-MEM algorithm, evaluates its scalability potential and discusses some of its computational limitations. Section 4 presents and compares the scalability potential of native implementations, Hadoop and Spark on the used test computer systems. Section 5 evaluates the effectiveness of these different solutions for genomics analysis by evaluating the scalability of BWA-MEM. Section 6 ends with the conclusions.

2 Genomics pipelines

In this section, we discuss the basic steps of a so-called *reference-based* DNA analysis pipeline, used for analyzing the mutations in a DNA dataset using a known reference genome.

DNA sequencing

The first step in any genome analysis pipeline starts by acquiring the DNA data using sequencing machines. This is done in a massively parallel fashion with millions of short DNA pieces (called *short reads*) being sequenced at the same time. These reads are stored in large files of sizes ranging from tens to hundreds of gigabytes. One standard file format used today to store these reads is called the FASTQ file format (Jones12).

Read mapping

The second step is used to assemble the short reads into a full genome, by mapping the short reads to a reference genome. This is one of the first computational steps in any genomics analysis pipeline, needed to reconstruct the genome. At the same time, it is one of the most computationally intensive steps, requiring a lot of

CPU time. The output represents a mapping of the possible locations of a specific read in the FASTQ file to a specific reference genome. BWA-MEM is one of the most widely used DNA mapping programs (Li13).

Variant calling

The third step is called variant calling, which uses algorithms to identify the variations of a mutated DNA as compared to a reference DNA. This analysis is becoming standard in the field of genomics diagnostics, where DNA analysis is used to advise clinical decision. The Genome Analysis Toolkit (GATK) and SAMtools are two widely used software tools for variant calling (Pabinger13).

3 BWA mapping algorithm

BWA-MEM is one of the most widely used DNA mapping algorithms that ensures both high throughput and scalability of the large datasets used in genomics. This section discusses the BWA-MEM algorithm which we use as an example for parallel algorithms used in big data application domains.

BWA-MEM, as well as many other DNA mapping algorithms, is based on the observation that two DNA sequences of the same organism are likely to contain short highly matched subsequences. Therefore, they can follow a strategy which consists of two steps: 1) seeding and 2) extension. The seeding step is to first locate the regions within the reference genome where a subsequence of the short read is highly matched. This subsequence is known as a seed, which is an exact match to a subsequence in the reference genome. After seeding, the remaining read is aligned to the reference genome around the seed in the extension step using the Smith-Waterman algorithm (Houtgast15).

In BWA-MEM, before starting the read alignment, an index of the reference genome is created. This is a one time step and hence not on the critical execution path. In our discussion, we assume that an index is already present. The different execution stages of BWA-MEM read alignment algorithm are described below. The first two stages belong to seeding.

SMEM generation

BWA-MEM first computes the so-called supermaximal exact matches (SMEMs). An SMEM is a subsequence of the read that is exactly matching in the reference DNA and cannot be further extended in either directions. Moreover, it must not be contained in another match.

Suffix array lookup

The suffix array lookup stage is responsible for locating the actual starting position of the SMEM in the reference genome. An SMEM with its known starting position(s) in the reference genome forms seed(s) in the reference.

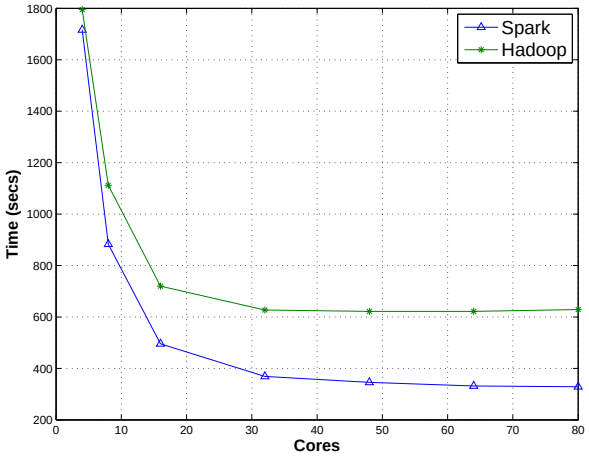


Figure 1: Execution time of WordCount on Power7

Seed extension

Seeds are subsequences of the read that are exactly matching in the reference genome. To align the whole read against the reference genome, these seeds are extended in both directions. This extension is performed using a dynamic programming algorithm based on Smith- Waterman.

Output

The read alignment information is written to a file in the SAM (sequence alignment/map) format (Li09).

4 Baseline performance analysis

The increasing sizes of big data files have called for a continued effort to develop systems capable of managing such data sizes and enabling the needed scalability to process them efficiently. Hadoop MapReduce is the current industry system of choice for big data systems, which uses the in-disk Hadoop distributed file system (HDFS). Apache Spark is emerging as a strong competitor in the field due to its in-memory resilient distributed datasets. This section compares these two systems using the WordCount benchmark to identify a baseline for the BWA-MEM implementation.

We tested the WordCount application on two different kinds of machines. The first one is an IBM PowerLinux 7R2 with two Power7 CPUs and 8 physical cores each. The Power7 cores are capable of executing 4 simultaneous threads. The second machine is an Intel Linux server with two Xeon CPUs and 10 physical cores each. The Xeon cores are capable of executing 2 simultaneous threads.

The results for the WordCount application are shown for the IBM Power7 and Intel Xeon in Figure 1 and 2, respectively. For the Hadoop version, the input files are read from the HDFS file system, while for Spark, the files are read from the local file system. We can see that on both machines, the Spark version is faster than the Hadoop version. Moreover, there is an approximately constant increase in the execution time of

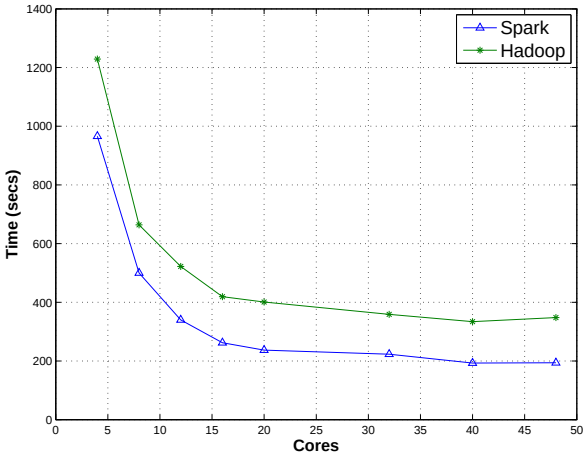


Figure 2: Execution time of WordCount on Xeon

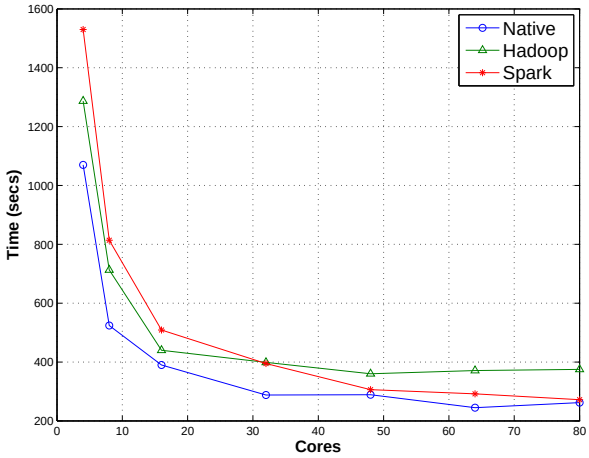


Figure 3: Execution time of BWA-MEM on Power7

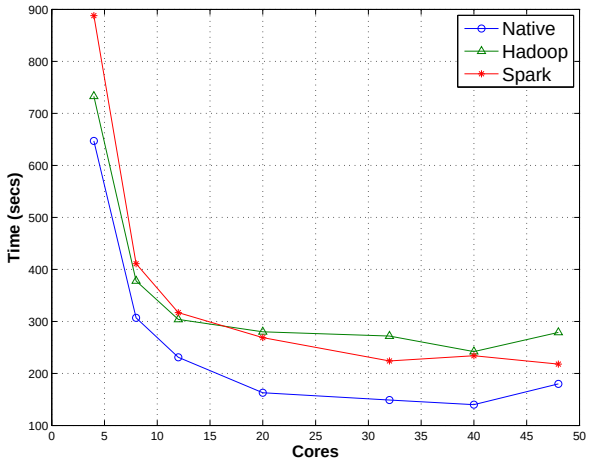


Figure 4: Execution time of BWA-MEM on Xeon

Hadoop as compared to that of Spark for the different thread count on both machines. This indicates a constant added overhead for Hadoop-based programs over Spark for different number of threads. This overhead is partly incurred by Hadoop having to access files from the HDFS system instead of the local file system.

The figures also show that the highest performance gains are achieved using scalability of the physical cores (up to 16 threads for the Power7 and 20 threads for the Xeon). In addition, high performance gains are achieved by running 2 threads per core, with the Power7 achieving better performance gains as compared to the Xeon. Further increase in the thread count on the Power7 only achieves marginal gains. One interesting remark is that on both machines, over-saturating the CPUs by issuing more threads than the machine is capable of causes Hadoop to loose performance slightly, while Spark is still capable of a (marginal) increase in performance.

5 Experimental results

In this section, we investigate the scalability potential of BWA-MEM on big data systems as an example of genomics data processing pipelines. We compare its native implementation, developed for classical clusters, to the performance of an Apache Spark as well as of a Hadoop-based big data implementation. The Hadoop version uses the Halvade scalable system with a MapReduce implementation (Decap15). In this evaluation, we use the same IBM PowerLinux 7R2 and Intel Xeon servers we used for the WordCount example above.

The results for the BWA mapping are shown for IBM Power7 and Intel Xeon in Figure 3 and 4, respectively. The results are shown for 3 different version of BWA: 1. native, 2. Hadoop, and 3. Spark. Both the Hadoop and Spark versions divide the input dataset of short reads into a number of smaller files referred to as *chunks*. For example, for these experiments, we had 32 input chunks. Halvade and Spark were run with 4 instances, which means that 4 simultaneous BWA tasks were run in parallel on different data chunks. The number of threads used in each experiment was varied by controlling the number of threads issued by each BWA instance (using the `-t` command line argument). For example, to use 12 threads, we used 4 instances with each instance having 3 threads, while to use 32 threads, we used 4 instances with 8 threads each.

It also has to be mentioned here that there are few differences in how files are read in Hadoop and the Spark versions. In the Hadoop version, the input zipped files are extracted on the fly by the Hadoop MapReduce framework and delivered to the mappers line by line, where each mapper is executing one instance of BWA. Each instance then processes the input data line by line. That data is read right away by the BWA instances using the stdin buffer. On the other hand, each instance in the Spark version reads a zipped input file and then decompresses it first. Afterwards, the decompressed file is forwarded as an input to a BWA instance. However, in both cases, the output is written to the local file system. It is also important to mention here that in the Power7 case, we wrote the output into a RAM

disk for all three BWA versions. This is because we wanted to know how good BWA scales with simultaneous threads without letting file I/O overshadowing the execution time.

On both the Power7 and Xeon machines, we can see that simultaneous multithreading improves the performance of BWA. This is because the BWA algorithm usually suffers from a large number of memory stalls, as a result of random accesses to the memory that renders the cache ineffective causing a high cache miss rate. These cache misses can be reduced by simultaneous multithreading, since some threads can run while others are stalled. The Power7 system is able to make significant performance gains using its capability to issue 4 simultaneous threads. Spark is able to increase in performance by up to 87% with 80 threads as compared to 16 threads (# of physical cores).

One interesting result in the figures is that while the Hadoop version is faster by up to 17% using lower number of threads, the Spark version gets faster by up to 27% at higher number of threads. This behavior can be explained by the way the input chunk files are handled. As mentioned before, the Hadoop version uses on-the-fly decompression of the zipped input chunk files, while the Spark version first decompresses a zipped input chunk file before using it. This causes an increased overhead that makes Spark run slower for lower number of threads. As the number of threads increases, Spark improves in performance and overtakes Hadoop in execution time.

Finally, the figures show that over saturating the thread capacity of the cores (i.e., issuing more threads than number of virtual cores available) causes the performance of the native version and the Hadoop version to degrade, while Spark is able to continue to improve in performance. This behavior is similar to the one observed in the WordCount example. Therefore, this could be caused by the internal implementation of the Spark and Hadoop systems themselves.

6 Conclusions

This paper analyzed the scalability potential of the widely used BWA-MEM DNA mapping algorithm. The algorithm is embarrassingly parallel and can be used as an example to identify the scalability potential of embarrassingly parallel genomics applications using the Spark and Hadoop big data frameworks. The paper compared the performance of 3 BWA implementations: 1. a native cluster-based version, 2. a Hadoop version, and 3. a Spark versions. Results show that simultaneous multithreading improves the performance of BWA for all systems, increasing performance by up to 87% for Spark on Power7 with 80 threads as compared to 16 threads (# of physical cores). The results also show that while the Hadoop version is faster by up to 17% using 4 threads, the Spark version gets faster by up to 27% at higher number of threads. Finally, the results also indicate that the Spark system is more capable of handling higher number of threads as it is able to continue to reduce its run time when over-saturating the thread capacity of the cores.

References

D. Decap, J. Reumers, C. Herzeel, P. Costanza and J. Fostier, "Halvade: scalable sequence analysis with MapReduce", *Bioinformatics*, btv179v2-btv179, 2015.

E.J. Houtgast, V.-M. Sima, K. Bertels and Z. Al-Ars, "An FPGA-Based Systolic Array to Accelerate the BWA-MEM Genomic Mapping Algorithm", *International Conference on Embedded Computer Systems: Architectures, Modeling and Simulation*, 2015.

D.C. Jones, W.L. Ruzzo, X. Peng and M.G. Katze, "Compression of next-generation sequencing reads aided by highly efficient de novo assembly", *Nucleic Acids Research*, 2012.

B.J. Kelly, J.R. Fitch, Y. Hu, D.J. Corsmeier, H. Zhong, A.N. Wetzel, R.D. Nordquist, D.L. Newsom and P. White,"Churchill: an ultra-fast, deterministic, highly scalable and balanced parallelization strategy for the discovery of human genetic variation in clinical and population-scale genomics", *Genome Biology*, vol. 16, no. 6, 2015.

H. Li, "Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM", arXiv:1303.3997 [q-bio.GN], 2013.

H. Li, B. Handsaker, A. Wysoker, T. Fennell, J. Ruan, N. Homer, G. Marth, G. Abecasis, R. Durbin, "The Sequence Alignment/Map format and SAMtools", *Bioinformatics*, vol. 25, no. 16, pp. 2078-2079, 2009.

S. Pabinger, A. Dander, M. Fischer, R. Snajder, M. Sperk, M. Efremova, B. Krabichler, M.R. Speicher, J. Zschocke, Z. Trajanoski, "A survey of tools for variant analysis of next-generation genome sequencing data", *Brief Bioinformatics*, bbs086v1-bbs086, 2013.

L.D. Stein, "The case for cloud computing in genome informatics", *Genome Biology*, vol. 11, no. 207, 2010.

M. Zaharia, M. Chowdhury, T. Das, A. Dave, J. Ma, M. McCauley, M.J. Franklin, S. Shenker, I. Stoica, "Resilient Distributed Datasets: A Fault-Tolerant Abstraction for In-Memory Cluster Computing", *NSDI*, April 2012.

Explain sentiments using Conditional Random Field and a Huge Lexical Network

Mike Donald TAPI NZALI
I3M, Univ. Montpellier
Montpellier, France
LIRMM, Univ. Montpellier
Montpellier, France
mdtapinz@univ-montp2.fr
tapinzali@lirmm.fr

Joël MAÏZI
Pierre POMPIDOR
Sandra BRINGAY
LIRMM, Univ. Montpellier
Montpellier, France

Christian LAVERGNE
I3M, Univ. Montpellier
Montpellier, France
Caroline MOLLEVI
Biostatistics Unit, ICM
Montpellier, France

Abstract

In this paper, we focus on a particular task which consists in explaining the source and the target of sentiments expressed in social networks. We propose a method for French, which overcomes a fine syntactic parsing and successfully integrate the Conditional Random Field (CRF) method and a smart exploration of a very large lexical network. Quantitative and qualitative experiments were performed on real dataset to validate this approach.

1 Introduction

In this article, we focus on a particular task of sentiment analysis which consists in explaining the target and the source of sentiments. For example, in the sentence "*We like the green initiative*", the sentiment is expressed by the verb "*like*", the target is "*green initiative*" and the source is "*We*". In (Bringay et al., 2014), we have proposed a method based on syntactic roles for English texts. Experiments have shown that our method is robust, even on the texts that are difficult to process : messages in health forums that contain misspelling and slang. Indeed, the method is not based on fine syntactic parsing. However, it is not possible to transpose this method directly to French because, to our knowledge, there is no resource available to explain semantic roles.

In this context, we propose a new approach based on machine learning methods and a very large lexical network, in French, issued from a contributory game *JeuxDeMots*[1]. The challenge is twofold: 1) Instead of using fine syntactic parsing, we use a statistical modeling method called Conditional Random Field (CRF), to extract candidates for targets and sources in large volumes of texts issued from poorly written social web messages 2) we also exploit the huge French lexical network *JeuxDeMots* (more than $300,000$ nodes and $7,000,000$ relations) to choose the best sources and targets among the candidates identified with CRF. This new method has been successfully exploited to analyse the sentiments expressed in the French tweets dealing with environment and climate change.

The paper is organized as follows. In section 2, we briefly present the state of the art. In section 3, we provide a description of our method. In section 4, we provide all the detail of the experiments carried out and the prime results. Finally in section 5, we conclude this work by providing the main perspectives associated with this work.

2 State of the Art

Since the early 2000s, sentiment analysis, also called "opinion mining", has experienced growing interest. Many methods have been developed to extract emotional states expressed or implied in texts. To identify sentiments, many resources exist (e.g. list of words, phrases, idioms), which were built mostly for English and polarity (e.g. *Linguistic Inquiry and Word Count* (Tausczik and Pennebaker, 2010)) or emotions (e.g. *NRC* lexicon (Mohammad and Turney, 2010a)). Some methods extend these vocabularies to specific domains (Neviarouskaya et al., 2011). Others are not restricted to the use of lexicons as (Strapparava and Mihalcea, 2008) who implement learning approaches.

Two categories of approaches are used to link sentiments and potential target and source. 1) Methods that essentially implement syntactical aspects, represented by combinations of rules (Mudinas et al., 2012) as the polarity inverters (*do not, just, very...*), conjunctions (*but, or*), etc. The effectiveness of these methods is strongly linked to language style that impacts on the syntactic rules

[1] http://www.jeuxdemots.org/jdm-accueil.php

to take into account and are not adapted to social web texts. 2) Methods that are based on different distance computations between words denoting sentiments and potential targets and source (as the proximity (Hu and Liu, 2004) or the position in the syntactic tree (Wu et al., 2009)). There are also many hybrid methods (Ding and Liu, 2007). In (Bringay et al., 2014), we proposed an efficient approach for English texts that requires a resource *FrameNet*[2] and the *SEMAFOR* parser [3] for explaining the semantic roles. To our best knowledge, such a resource does not exist in French. Consequently, we have proposed a method combining learning approach to find targets and sources candidates and a smart exploration of a large French lexical network to select the best one.

To choose the best candidate, we use the *Games with a purpose JeuxDeMots*, created in 2007 (Lafourcade and Joubert, 2012), to build a huge lexical network for french. For example, the game asks the player ideas associated with term *climatic change*. The player freely associate terms such as *bear*. Other players have already faced the same term. The player wins credits if the proposed term has already been proposed by another player. The more the proposal is specific, the more points he obtains. The lexical network generated with this game is a directed graph, with terms (nodes) and typed and weighted relations (edges) between the terms. There are more than 50 types of relationships. To weight the edges, *JeuxDeMots* is based on crowdsourcing. Each relation is weighted by a *strength of association*, denoted C_{jdm} representing the number of players who have associated two terms by the same relation. A first challenge is to explore the network to link terms in the sentences (sentiment and target/source) and explain these relations. The second challenge will be to exploit this very large network that includes more than 300000 terms and more than 7000000 relations.

3 Methods

The method is organised into 3 steps :

Step 1: Corpus. The corpus we used and annotations have been made in the Ucomp project[4]. These tweets deal with climate change. Table 1 and 2 present detailed statistics on the corpus.

Step 2: candidates generation with CRF. The CRF model was developed with domain independent surface and lexical features for the text tokens:

Class	Learning step		Test	
	#	%	#	%
Source	2448	31	1057	31
target	1875	24	804	24
Total	**7867**	**55**	**1861**	**55**

Table 1: Distribution of *source* and *target* in the corpus used

- The original token from the text (word form);
- Surface features: capitalization of the token (all in upper/lower case, combination of both), and punctuation mark in the token (PUNCT, NO_PUNCT);
- Lexical features: *n*-grams, number of consecutive repeats. Token frequency was computed based on the entire training corpus.
- Brown clustering: we used Percy Liang's implementation of Brown clustering (Brown et al., 1992), which is an HMM-based algorithm. In our work, we partition words into a base set of 100 clusters, and induces a hierarchy among those 100 clusters.
- Emotion lexicon: We built semi-automatically a new lexicon of French sentiments (Amine et al., 2014) by translating and expanding the English NRC lexicon (Mohammad and Turney, 2010b). This lexicon is free to download [5]. For each tokens, the corresponding feature takes the value "Yes" if the token appear in the lexicon and "No" otherwise. As a *source* and a *target* are usually surrounded by a sentiment token, we also consider the apparition of the sentiment in the neighborhood of the token (e.g. two tokens before or after the current token).

We experimented with standard tokenization (provided by TreeTagger) and custom tokenization (Tapi-Nzali et al., 2015) of French TreeTagger by

[2] https://FrameNet.icsi.berkeley.edu/fndrupal/home

[3] http://demo.ark.cs.cmu.edu/parse

[4] http://www.ucomp.eu/

[5] http://www.lirmm.fr/~abdaoui/FEEL.html

adding some segmentation rules (e.g. apostrophe : *l'image* is segmented into *l'* and *image*).

Step 3: Lexical network construction of each sentence. The purpose of this step is to extract a part of the lexical network *JeuxDeMots* representing the relationship between the meaning of the word and the candidates identified in Step 2. The intuition of the algorithm is the following one. We cross the lexical network from node to node. We stop when we no longer encounter new words or if we reach a maximum depth. Two other constraints are used to limit the expansion of the graph.

Constraint 1. To consider only the parts of the network related to our topic (environment and sentiment), we expand a node to another if the new one belong to these two predefined lexical fields chosen via Larousse thesaurus[6]. If there is no node in the lexical field, we expand to all neighboring nodes.

Constraint 2. We use the *association strength* C_{jdm} weighting the edges and consider only the relations frequently instantiated by players. A threshold is set by default.

Step 4: Identification of shortest paths. The objective of this step is to identify in the graph generated in step 3, the paths that must correspond to a compromise between the shortest paths, with a little depth, most reliable according to *strength of association* C_{jdm}. We have therefore redefined weights w_{rt} to foster some relationships like synonymy or significant semantic roles such as *patient* and *agent*. To identify the paths, we have adapted the shortest path algorithm and used the weights computed according to formulas 1 and 2. The weight w_i foster relationships that interest us with w_{rt} while taking into account the *strength of association* C_{jdm}. In equation 1, we verify a balance between w_{rt} and C_{jdm} terms. In equation 2, the term $(n-1)^2$ enables to penalize depth. We only consider paths which contain at least one *agent* or *patient* relationship. The path with the best score is proposed to the user to explain the link between the sentiment and the target or source.

$$w_1 = \frac{1}{1 - \frac{1}{C_{jdm}}} + w_{rt} \quad (1)$$

$$w_n = (max(w_{n-1}) * \frac{1}{1 - \frac{1}{C_{jdm}}} + w_{rt}) * (n-1)^2 \quad (2)$$

4 Experiments

Our experiments were carried out using 10-fold cross-validation. To do this, the training corpus was divided into 10 folds. To build our model, we need a training, development and test corpus. Cross-validation has been distributed as follows: The model is built on 8 folds, the optimization of the construction is performed on the ninth part (development) and the model evaluation performed on the last fold (test).

To perform our experiments, we use Wapiti[7](Lavergne et al., 2010). It is a very fast toolkit for segmenting and labelling sequences with discriminative models. For the iterative estimation of the model parameters, we used the algorithm RPROP (Riedmiller and Braun, 1992).

Table 3 presents the results obtained by different CRF models on training set by cross validation. The features of four bests configurations are :

- **Configuration 1** : Part Of Speech tagging + lemmatization + lowercase
- **Configuration 2** : Part Of Speech tagging + lemmatization + lowercase + brown clustering
- **Configuration 3** : All (Part Of Speech tagging + lemmatization + lowercase + brown clustering + emotion lexicon
- **Configuration 4** : Part Of Speech tagging + lemmatization + lowercase + emotion lexicon

	Training	**Test**	**All**
Tweets	3,001	1,783	4,784
Tokens	78,771	48,612	127,383
Source	1,131	604	1,735
Target	3,954	2,251	6,205

Table 2: Description of the corpus

The results of the evaluation are reported in terms of precision (the number of *source* and *target* correctly extracted over the total number of *source* and *target* extracted), recall (the number

[6] http://www.larousse.fr/dictionnaires/francais/thesaurus/77857

[7] http://wapiti.limsi.fr

Test	Model	Exact match			Partial match		
		P	R	F	P	R	F
Corpus used	Config 1	**0.40**	**0.27**	**0.32**	0.76	**0.52**	**0.62**
	Config 2	0.39	0.26	0.31	0.76	**0.52**	**0.62**
	Config 3	0.40	0.25	0.30	0.77	0.48	0.59
	Config 4	0.40	0.24	0.30	**0.78**	0.47	0.59

Table 3: Evaluation of *source* and *target* extraction in French tweet corpus.

Class	Exact match			Partial match		
	P	R	F	P	R	F
SOURCE	0.64	0.38	0.48	0.76	0.45	0.57
TARGET	0.34	0.20	0.25	0.79	0.47	0.59
All	0.40	0.24	0.30	0.78	0.47	0.59

Table 4: Results of best model on the test corpus

of *source* and *target* correctly extracted over the total number of *source* and *target* marked the corpus used) and F-measure (the harmonic average of precision and recall). We show two types of results. the first is the results achieved by *Exact match* and the second by *Partial match*. We consider that there is an ***Exact match*** when the tokens obtained with our model match exactly those of the standard test annotation and we consider a ***Partial match*** when the obtained token are included. For example, *governor* partially matches *The governor*. Overall, with *Exact match*, configuration 1 is the best performing configuration. Results show that, we performed a good results with *Partial match*. Compared to other configurations, configuration 4 gives the best results on precision (Precision 0.78), and configuration 1 and 2 give the same results and the best results on recall and F-measure (recall 0.52 and F-measure 0.62). Contrary, with the sentiment lexicon as feature, we increase precision, decrease recall and f-measure. Brown Clustering is good feature if we want to have a good precision.

If CRF is relevant for extracting target and source candidates, how can we link them to the sentiments also expressed in the sentences? In figure 1, a sentence is annotated after the exploration of the lexical network. Sentiment tokens are represented by red points. Target and Source obtained with CRF are colored (in blue and yellow). Arrows correspond to the paths identified in the network. The more the arrow is thick the more the path is valuated.

5 Conclusions and Future Work

A combination of CRF and huge lexical network exploration seems promising for explaining sentiments in social networks. By experimenting with the CRF model, we found that the results varied depending on the features. The best results are obtained with the features: *lemmatization, cluster ID and Part Of Speech tagging*.

The first advantage of this method is that we can detect multiple tokens (e.g. *parc eolien terrestre*, *La France*). Another advantage of this method is that it is efficient even if the sentence contains misspelling. For example, the system identify *modèle de dévelopement durable* (sustainable development model) even if the word *dévelopement* is misspelled. Finally, the main advantage of this contribution is not to restrict sentiment, source and target identification to the case in which sentiment word is present. Indeed, in most cases people express sentiments implicitly without using these sentiment words. An emotion cannot be limited to something a person feels about a fact and not the sentiment that a person expresses about this fact. Thus, it could be common to explicitly express sentiments about things, but it is more common to feel emotions without expressing them explicitly. Our method take into account this fact and try to identify source and target beyond the explicit cases.

The principal limitation of our method is the length of the sentences in the considered corpora (size of the tweets). In many sentences, there is no source or no target. Results are significantly reduced. Moreover, a quantitative study has to be performed on step 3 and 4 to evaluate the quality of the computed relations between sentiment and targets/sources.

Prospects associated with this work are numerous. First, in this work we focus only on the targets/sources expressed in sentences and we now have to focus on inter-sentence relationships at paragraph level. In future work, we are going to use the best model we obtained on health forum messages with longer sentences. We will also compare our method to identify relations between sentiments and source/target with the methods of the state of the art. We will also adapt CRF to extract directly relations. Finally, we will present to users the part of the network used to identify relations in order to help their interpretation

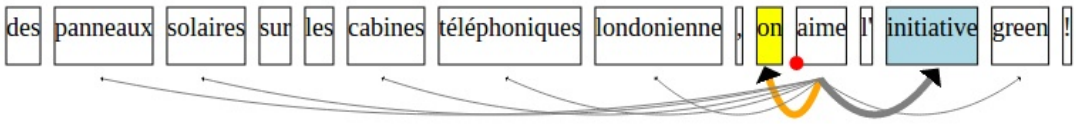


Figure 1: Relation example

References

Abdaoui Amine, Azé Jérôme, Sandra Bringay, and Pascal Poncelet. 2014. Feel : French extended emotional lexicon. volume ISLRN: 041-639-484-224-2. ELRA Catalogue of Language Resources.

Sandra Bringay, Eric Kergosien, Pierre Pompidor, and Pascal Poncelet. 2014. Identifying the targets of the emotions expressed in health forums. In *CICLing (2)*, pages 85–97.

Peter F Brown, Peter V Desouza, Robert L Mercer, Vincent J Della Pietra, and Jenifer C Lai. 1992. Class-based n-gram models of natural language. *Computational linguistics*, 18(4):467–479.

Xiaowen Ding and Bing Liu. 2007. The utility of linguistic rules in opinion mining. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '07, pages 811–812, New York, NY, USA. ACM.

Minqing Hu and Bing Liu. 2004. Mining opinion features in customer reviews. In *Proceedings of the 19th National Conference on Artifical Intelligence*, AAAI'04, pages 755–760. AAAI Press.

Mathieu Lafourcade and Alain Joubert. 2012. Increasing Long Tail in Weighted Lexical Networks. In *Cognitive Aspects of the Lexicon (CogAlex-III), COLING*, page 16, France, December.

Thomas Lavergne, Olivier Cappé, and Fran¸cois Yvon. 2010. Practical very large scale crfs. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 504–513, Uppsala, Sweden.

Saif M. Mohammad and Peter D. Turney. 2010a. Emotions Evoked by Common Words and Phrases : Using Mechanical Turk to Create an Emotion Lexicon. In *Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*, pages 26–34, Stroudsburg, PA, USA. ACL.

Saif M. Mohammad and Peter D. Turney. 2010b. Emotions evoked by common words and phrases: Using mechanical turk to create an emotion lexicon. In *Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*, CAAGET '10, pages 26–34, Stroudsburg, PA, USA. Association for Computational Linguistics.

Andrius Mudinas, Dell Zhang, and Mark Levene. 2012. Combining lexicon and learning based approaches for concept-level sentiment analysis. In *Proceedings of the First International Workshop on Issues of Sentiment Discovery and Opinion Mining*, WISDOM '12, pages 5:1–5:8, New York, NY, USA. ACM.

Alena Neviarouskaya, Helmut Prendinger, and Mitsuru Ishizuka. 2011. Affect analysis model: Novel rule-based approach to affect sensing from text. volume 17, pages 95–135, New York, NY, USA, January. Cambridge University Press.

Martin Riedmiller and Heinrich Braun. 1992. Rprop-a fast adaptive learning algorithm. In *Proc. of ISCIS VII), Universitat*. Citeseer.

Carlo Strapparava and Rada Mihalcea. 2008. Learning to identify emotions in text. In *Proceedings of the 2008 ACM Symposium on Applied Computing*, SAC '08, pages 1556–1560, New York, NY, USA. ACM.

Mike Donald Tapi-Nzali, Aurélie Névéol, and Xavier Tannier. 2015. Analyse d'expressions temporelles dans les dossiers électroniques patients. In *Actes de la Conférence Traitement Automatique des Langues Naturelles (TALN 2015)*, Caen, France, June.

Yla R. Tausczik and James W. Pennebaker. 2010. The psychological meaning of words: Liwc and computerized text analysis methods. volume 29, pages 24–54.

Yuanbin Wu, Qi Zhang, Xuanjing Huang, and Lide Wu. 2009. Phrase dependency parsing for opinion mining. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3 - Volume 3*, pages 1533–1541, Stroudsburg, PA, USA. Association for Computational Linguistics.

A Security Price Data Cleaning Technique: Reynold's Decomposition Approach

Rachel V. Mok
Department of Mechanical Engineering
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139
rmok@mit.edu

Wai Yin Mok, Kit Yee Cheung
College of Business Administration
University of Alabama in Huntsville
Huntsville, Alabama, 35899
mokw@uah.edu, kityeemok@gmail.com

Abstract

We propose a security price data cleaning technique based on Reynold's decomposition that uses T_0 (the time period of integration) to determine the de-noise level of the price data. The goal of this study is to find the optimal T_0 that reveals an underlying price trend, possibly indicating the intrinsic value of the security. The DJIA (Dow Jones Industrial Average) Index and the thirty companies comprising the index are our fundamental interest. Preliminary results suggest that the graphs of α (a key percentage measure) versus T_0 of the thirty companies and the DJIA Index exhibit at least two properties: (1) α drops exponentially as T_0 increases when $T_0 \lessapprox$ order of magnitude of 100 days, and (2) α drops linearly as T_0 increases when $T_0 \gtrapprox$ order of magnitude of 100 days. For the DJIA Index itself, T_0 is less than order of magnitude of 100 days. The result of applying our technique to each component stock of the DJIA parallels the result of the technique applied to the DJIA Index itself.

1 Introduction

Understanding and analyzing financial data in order to forecast and make cost-effective decisions is challenging because of the complex and volatile nature of security prices. The most recent financial market meltdown in 2008-09 casted doubts on financial data analysis and forecasting. Inability to recognize or acknowledge financial distress signaled by pertinent financial data was a significant factor leading to these catastrophic economic results (Kaur, 2015). Thus, veracity of financial data takes priority in any data driven decision making. Like any big data infrastructure, veracity includes validation, noise level, deception, detection, relevance and ranking of data collected (Goes, 2014). Depending on how collected financial data are captured and processed in an analysis, generated assessments can vary greatly from real financial market performance. One has to look no farther than the recent settlement of $77 million between the SEC and Standard & Poor credit rating agency to see an example of how data analysis can be misleading (http://www.sec.gov/news/pressrelease/2015-10.html).

Several financial computation models that deal with cleaning financial data employ similar methodologies, such as candlestick strategies (Detollenaere and Mazza, 2014), multiple-stage algorithm for detecting outliers in ultra high-frequency financial market data (Verousis and ap Gwilym, 2010), financial data filtering (http://www.olsendata.com) and data-cleaning algorithm (Chung et al., 2004a; Chung et al., 2004b). Most data cleaning methodologies involve the detection, distribution and/or the removal of outliers (Shamsipour et al., 2014; Sun et al., 2013). However removing outliers in the dataset may have a statistical distortion effect on the dataset itself (Dasu and Loh, 2012).

To this end, we propose a data cleaning technique based on Reynold's decomposition in order to decompose the price data into a mean part and a fluctuating part. Fluctuations in stock prices are perpetual and irrational in time because the weak form of market efficiency and different types of market participants create a complex dynamic of behavioral finance (Verheyden et al., 2015). Nevertheless, our approach could minimize part of the effect of irrational price fluctuations by incorporating and averaging fluctuation points (i.e., outliers) within a moving time period of integration, T_0. In essence, the length of T_0 in the analysis determines the level of veracity, with the larger the T_0, the lesser the influence of the fluctuation points will be. We believe our data cleaning tech-

nique is particularly applicable to security prices due to the intense nature of security price changes in relatively short periods, and it allows the user to gauge different moving time periods of integration to produce a unique set of statistical data for targeted analysis.

2 Reynold's Decomposition

In the study of turbulence in fluid dynamics, each component of the velocity is characterized by fluctuations over time. One method to study the dynamics in this regime is to perform a Reynold's decomposition such that the mean part of the velocity is separated from the fluctuations. We propose that this technique could also be used to study financial data. In other words, we propose that the price as a function of time, $p(t)$, can be decomposed into the following:

$$p(t) = \bar{p}(t) + p'(t) \tag{1}$$

where $\bar{p}(t)$ is the mean portion and $p'(t)$ is the fluctuating portion of the price. We define $\bar{p}(t)$ to be a moving time-average that can be found by performing the following integral

$$\bar{p}(t) = \frac{1}{T_0} \int_{t-T_0/2}^{t+T_0/2} p(t')dt' \tag{2}$$

where T_0 is the time period of integration. T_0 must be a time period that is greater than the time period of the fluctuations, τ, and less than the time period of interest, T. T is dependent on each particular analysis; for example, T could be weeks, months, or years. Thus, $\tau < T_0 < T$. Furthermore, the time-averaged value of the fluctuating portion over the entire time period of interest is zero (Müller, 2006; Mills, 1999). As the time period of integration increases, $\bar{p}(t)$ is farther away from the actual $p(t)$ and the magnitude of $p'(t)$ increases. Thus, the goal of this research is to find the optimal time period of integration, T_0, that excludes the miscellaneous fluctuations and captures the essential trend of the price data.

3 Methods

In this study, we focus on the thirty companies comprising the Dow Jones Industrial Average (DJIA) as of May 13, 2015, and the DJIA Index because, being the second oldest financial index, the DJIA is the benchmark that tracks financial market performance as a whole. Thus, it represents a broad market, and its validity is intensely scrutinized and followed by at least 10 Wall Street analysts (Lee and Swaminathan, 1999; Moroney, 2012; Stillman, 1986). The ticker symbols for the thirty companies that were studied in this analysis are as follows: GS, IBM, MMM, BA, AAPL, UTX, UNH, HD, DIS, CVX, NKE, TRV, JNJ, MCD, CAT, XOM, PG, AXP, WMT, DD, V, JPM, MRK, VZ, MSFT, KO, PFE, INTC, CSCO, and GE. Because different companies can comprise the DJIA Index at any point in time, we only focus on the index as a whole when performing the analysis for the DJIA Index itself.

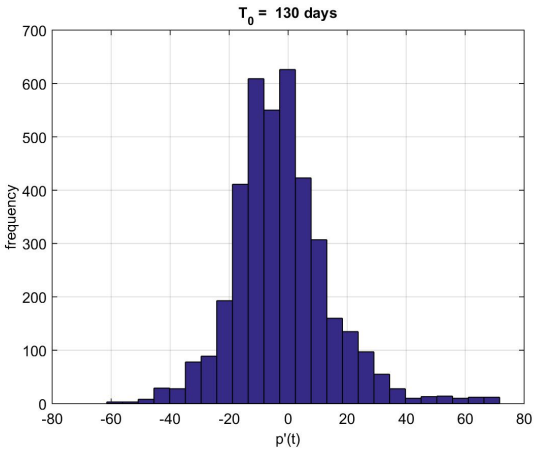


Figure 1: The histogram for GS with T_0 = 130 days.

Daily adjusted close stock price data for the thirty Dow Jones companies listed above from the time of inception of the company to May 13, 2015, are obtained from Yahoo! Finance. For the DJIA Index, the daily adjusted close stock price data from Jan. 29, 1985, to May 13, 2015, are also obtained from Yahoo! Finance. The adjusted close stock price is used because it accounts for stock dividends and splits. Only days in which the stock price is provided, i.e., business days, are considered in this study. Thus, the time from Friday to Monday is taken as only one (business) day.

We estimate the time period of fluctuations to be a day, $\tau \sim 1$ business day, and the time period of interest to be the total number of business days since the inception of the stock, $T \sim 260 \times n$ business days, where n represents the number of years since the inception of the stock. Further, we chose the following time periods of integration, T_0, for this study: 4 days, 10 days, 20 days, 30 days, 64 days, 130 days, 194 days, 260 days, 390

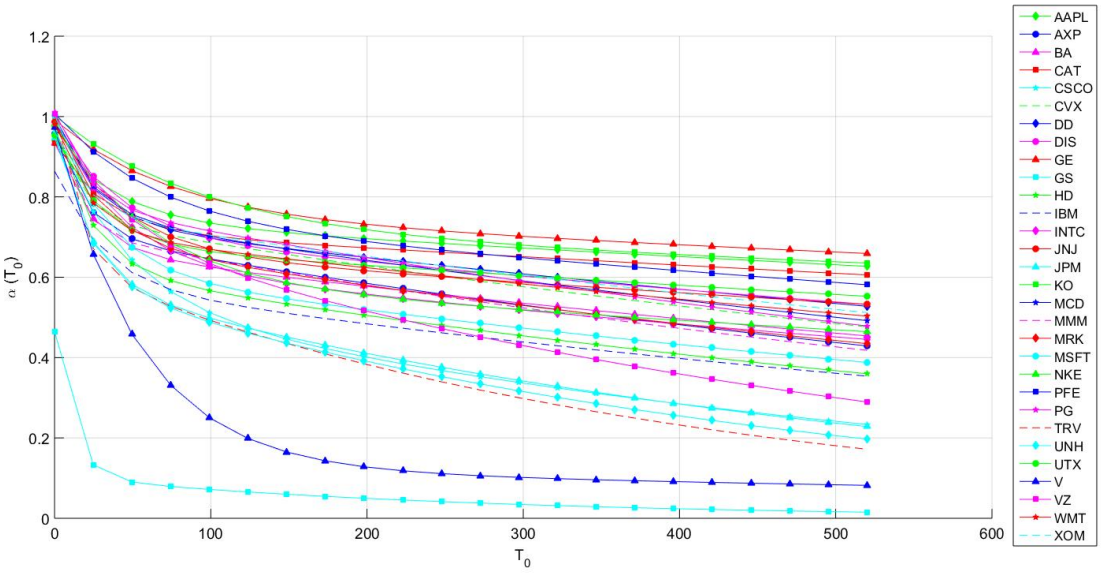


Figure 2: Fitted curve $\alpha(T_0) = a_1\ e^{b_1\ T_0} + c_1\ e^{d_1\ T_0}$ for the listed thirty stocks. $a_1, b_1, c_1,$ and d_1 are curve fitting parameters. The lowest goodness-of-fit measure R^2 among the thirty stocks is 0.9909.

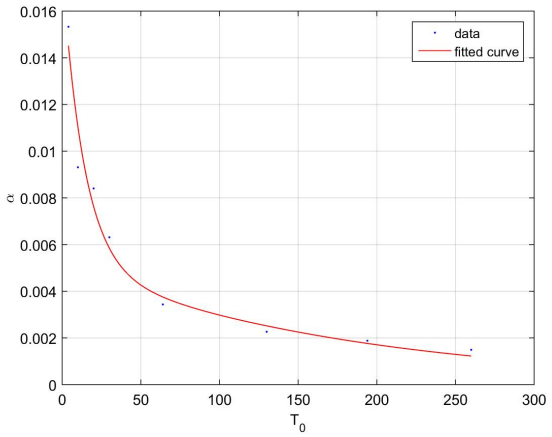


Figure 3: α vs. T_0 for the DJIA Index. The goodness-of-fit measure R^2 is 0.9707 for the fitted curve.

days, and 520 days, which roughly represent the following time periods: one week, two weeks, one month, one-and-a-half months, a quarter of a year, half of a year, three-quarters of a year, one year, one-and-a-half years, and two years, respectively.

$\bar{p}(t)$ is calculated by only considering the analysis time period from $T_0/2$ after the day of inception to $T_0/2$ before May 13, 2015, such that for each day $\bar{p}(t)$ is calculated, the full time period of integration is used. To exemplify, consider the case where $T = 1000$ days and $T_0 = 100$ days. Then the first 50 days (day 1 to day 50) are not included in the analysis, and neither are the last 50 days (day 951 to day 1000). For each day in the analysis time period, the integration stated in Eq. (2) is performed numerically to find $\bar{p}(t)$ for that day. $p'(t)$ is found by subtracting $\bar{p}(t)$ from $p(t)$, the actual price, for that day.

For each specific T_0, the statistics of $p'(t)$ are analyzed. Specifically, a histogram with 25 bins of $p'(t)$ is created for each T_0. As an example, Fig 1 shows a histogram for GS (The Goldman Sachs Group Inc). Note that like Fig 1, most of the histograms are centered around 0, which suggests that most of the fluctuations for the stocks are nearly zero. Therefore, the actual stock price is near or nearly equal to the local time-average for most of the time period analyzed. For most stocks, as T_0 increases, the maximum height achieved by the histogram decreases and the histogram tails become heavier. Thus, as T_0 increases, there are more observations away from the center of the distribution. This is observed because as the time period of integration increases, more points are considered in the average. Therefore, there is a greater likelihood that $\bar{p}(t)$ is different from the actual price.

To measure the fidelity of $\bar{p}(t)$ to $p(t)$, the number of data points of $p'(t)$ that are within 1 dollar from zero are counted and divided by the total number of data points in the analysis period. We will call this percentage measure α, and this measure should be as close as possible to 100% to reflect that $\bar{p}(t)$ is a good approximation of $p(t)$.

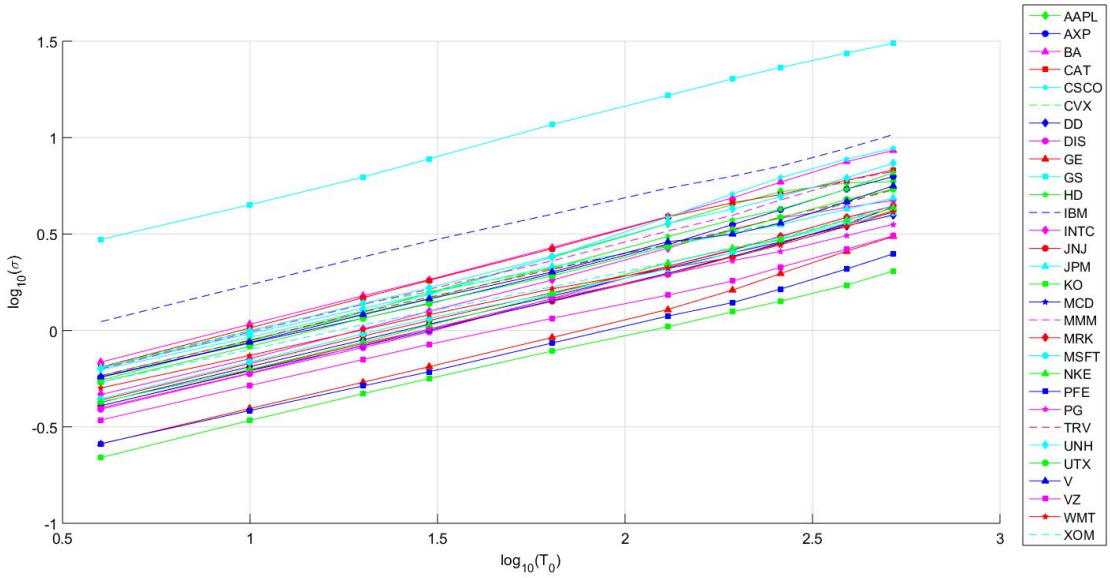


Figure 4: $\log_{10}(\sigma)$ vs. $\log_{10}(T_0)$ for the listed thirty stocks. Because each stock is a line in this plot, a power law relationship exists between σ and T_0.

As stated previously, if $p'(t)$ is near zero, that means that $\bar{p}(t)$ is close to $p(t)$ because $p(t) = \bar{p}(t) + p'(t)$. As T_0 increases, α decreases because the mean is farther away from the actual price when the integration period is larger. Using the MATLAB® curve fitting tool, it is found that for all of the thirty stocks the relationship between α and T_0 is best represented by the following equation

$$\alpha(T_0) = a_1\, e^{b_1\, T_0} + c_1\, e^{d_1\, T_0} \tag{3}$$

where $a_1, b_1, c_1,$ and d_1 are curve fitting parameters. In fact, the lowest goodness-of-fit measure R^2 among all thirty stocks is 0.9909. As an example, the curve fitting parameters for GS are $a_1 = 0.3613, b_1 = -0.09018, c_1 = 0.1034,$ and $d_1 = -0.003687$. The first derivative of this equation is

$$\frac{d\alpha}{dT_0} = a_1\, b_1\, e^{b_1\, T_0} + c_1\, d_1\, e^{d_1\, T_0} \tag{4}$$

and the second derivative is

$$\frac{d^2\alpha}{dT_0^2} = a_1\, (b_1)^2\, e^{b_1\, T_0} + c_1\, (d_1)^2\, e^{d_1\, T_0} \tag{5}$$

For most of the stocks, it was discovered that when T_0 is fewer than 100 days, the measure α drops exponentially as T_0 increases. However, the second derivative (Eq. (5)) becomes near zero in a range from 96 days to 387 days for the thirty stocks analyzed, with the most common being approximately 125 days. Thus, when T_0 is at least an order of magnitude of 100 days, α starts to decrease linearly for nearly all of the stocks analyzed. Fig 2 plots the curve fitted $\alpha(T_0)$ for all thirty analyzed stocks. As we can see, the general trend among the thirty stocks is that α drops exponentially when T_0 is fewer than 100 days, but α drops linearly when T_0 is greater than 100 days. An appealing fact is that the graph of α against T_0 for the DJIA Index, Fig 3, also exhibits similar trends in α, as shown in Fig 2. Note the different scales of the vertical axes of Fig 2 and Fig 3, which means that Fig 3 is much flatter than Fig 2.

Mathematically, we will define the point where the slope is constant by the following

$$\lim_{T_0 \to t_c} \frac{d^2\alpha}{dT_0^2} = 0 \tag{6}$$

where t_c is the time period of integration at which the second derivative of α approaches zero. Thus, for the thirty stocks analyzed, t_c is in the following range 96 days $< t_c <$ 387 days. Therefore, for time periods of integration larger than t_c, the change in α will be relatively small.

The standard deviation σ of the fluctuations $p'(t)$, defined as

$$\sigma = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(p'(t_i) - \bar{p'}(t))^2}, \tag{7}$$

is also analyzed where N is the total number of data points and $\bar{p'}(t)$ is the total time average of the

fluctuations. A large σ of the fluctuations reflects that $\bar{p}(t)$ is not equal to $p(t)$. To remove the miscellaneous fluctuations of the price, σ should be as large as possible. As indicated by the straight lines in the log-log plot in Fig 4, σ and T_0 are related by a power law where the slope of the line indicates the exponent.

Using the MATLAB® curve fitting tool, we fit the following equation for each of the thirty stocks:

$$\sigma = a_2 \, T_0^{b_2} + c_2 \tag{8}$$

where a_2, b_2, and c_2 are curve fitting parameters. As an example, $a_2 = 2.194, b_2 = 0.4317$, and $c_2 = -1.425$ for GS. The lowest goodness-of-fit measure R^2 among the thirty stocks is 0.9849. b_2, the exponent, varies from 0.35 to 0.69 for all thirty stocks. The average exponent is 0.5.

4 Results and Conclusions

This paper demonstrates preliminary results of an ongoing security price data cleaning research. We found that the graphs of α versus T_0 of the thirty companies and the DJIA Index exhibit at least two properties: (1) α drops exponentially as T_0 increases when $T_0 \lessapprox$ order of magnitude of 100 days, and (2) α drops linearly as T_0 increases when $T_0 \gtrapprox$ order of magnitude of 100 days. Thus, the optimal T_0 for the thirty companies studied is approximately 100 days. For the DJIA Index itself, the optimal T_0 appears to be less than 100 days. One of the possible explanations is that the DJIA Index might show the counter measure effect of fluctuation points among the thirty companies since the DJIA is a composite of the thirty companies that collectively provide a balance view of the market. As a result, T_0 might be even smaller for the second derivative to approach zero. We also found that σ and T_0 are related by a power law. As for future research, we plan to define mathematical metrics in our study of security price valuations and trading strategies.

References

Kee H. Chung, Chairat Chuwonganant, and D. Timothy McCormick. 2004a. Order preferencing and market quality on NASDAQ before and after decimalization. *Journal of Financial Economics*, 71(3):581–612.

Kee H. Chung, Bonnie F. Van Ness, and Robert A. Van Ness. 2004b. Trading costs and quote clustering on the NYSE and NASDAQ after decimalization. *Journal of Financial Research*, 27(3):309–328.

Tamraparni Dasu and Ji Meng Loh. 2012. Statistical distortion: Consequences of data cleaning. *Proceedings of the VLDB Endowment*, 5(11):1674–1683.

Benoit Detollenaere and Paolo Mazza. 2014. Do Japanese candlesticks help solve the trader's dilemma? *Journal of Banking and Finance*, 48:386–395, November.

Paulo B. Goes. 2014. Big data and IS research. *MIS Quarterly*, 38(3):iii–viii, September.

Inderjit Kaur. 2015. Early warning system of currency crisis : Insights from global financial crisis 2008. *IUP Journal of Applied Economics*, 14(1):69–83, January.

Charles M. C. Lee and Bhaskaran Swaminathan. 1999. Valuing the Dow: A bottom-up approach. *Financial Analysts Journal*, 55(5):4–23, September.

Anthony F. Mills. 1999. *Basic Heat and Mass Transfer*. Prentice Hall, second edition.

Richard Moroney. 2012. What we're thinking add it up: Dow has further upside. *Dow Theory Forecasts*, 68(9):2–3, February.

Peter Müller. 2006. *The Equations of Oceanic Motions*. Cambridge University Press.

Mansour Shamsipour, Farshad Farzadfar, Kimiya Gohari, Mahboubeh Parsaeian, Hassan Amini, Katayoun Rabiei, Mohammad Sadegh Hassanvand, Iman Navidi, Akbar Fotouhi, Kazem Naddafi, Nizal Sarrafzadegan, Anita Mansouri, Alireza Mesdaghinia, Bagher Larijani, and Masud Yunesian. 2014. A framework for exploration and cleaning of environmental data - Tehran air quality data experience. *Archives of Iranian Medicine*, 17(12):821–829, December.

Richard Joseph Stillman. 1986. *Dow Jones Industrial Average : history and role in an investment strategy*. Irwin Professional Pub.

W. Sun, B. Whelan, AB. McBratney, and B. Minasny. 2013. An integrated framework for software to provide yield data cleaning and estimation of an opportunity index for site-specific crop management. *Precision Agriculture*, 14(4):376–391, August.

Tim Verheyden, Lieven De Moor, and Filip Van den Bossche. 2015. Towards a new framework on efficient markets. *Research in International Business and Finance*, 34:294–308, May.

Thanos Verousis and Owain ap Gwilym. 2010. An improved algorithm for cleaning ultra high-frequency data. *Journal of Derivatives & Hedge Funds*, 15(4):323–340.

Automation of process to load database from OSM for the design of public routes

G. Bejarano, J. Astuvilca, P. Vega
Pattern Recognition and Applied Artificial Intelligence Group
Pontifical Catholic University of Peru
1801 Univesitaria Avenue, Lima, Peru
gissella.bejarano@pucp.edu.pe, {j.astuvilcaf, vega.pedro}@pucp.pe

Abstract

This paper presents an automatic process to load the transport information of Lima - Peru city as network and public transportation routes from *OpenStreetMap* (OSM) to a *PostgreSQL* database. Moreover, this work shows how OSM data is transformed in two graphs and in several actual bus routes. The work is a combination of SQL commands and python programs to transform OSM data and combine it with external information to specific structures in a final database. This information was necessary for a second goal that was the approach and study of the Transit Network Design Problem (TNDP). Since OSM is a free collaborative tool, it is subject to manual errors in the map that produce mistaken graphs and routes. These errors are corrected daily because the area information is updated frequently. The obtained results confirm that this process could be automated in a one-click step. Finally, we tried other methods to upload OSM data and none of them got an exact graph except for *osm2pgrouting*.

1 Introduction

Until 2014, the city of Lima, capital of Peru, had 403[1] formal routes, 1 Bus Rapid Transit[2] (BRT) line and 1 metro line. There are projects to build a new BRT corridor[3] and 5 more metro lines[4], in order to improve public transportation. Actually, some months ago, the number of formal bus routes was reduced to 322 urban routes, 77 beltway routes and 15 routes in unattended zones[5] because of the oversupply of routes that exist in the city.

The task of deciding which set of bus routes must continue in order to attend the public mobility demand and to optimize the cost of the user and the cost of the operator is a huge and full of possibilities task (Farahani et al., 2013). For this reason, a metaheuristic algorithm might be used to find a nearly global optimal solution (Farahani et al., 2013) to this problem defined by Fan and Machemehl (2004) as the Transit Network Design Problem (TNDP). Besides, according to Fan and Machemehl (2004), metaheuristics are more suitable to work with practical and real cases. Nevertheless, a real problem was to load a complete network as an input to the algorithm and form a graph by which a set of routes will pass. For this reason, the purpose of this work is to implement an automatic process that provides the necessary data input to an algorithm that solves the TNDP.

For our algorithm, two levels of graphs would be tested in order to find a solution progressively. The first graph represents the adjacent Traffic Zones (Agency, 2005), which we call minizones because they can be smaller boundaries in the future, while the second graph is the road and highways network level. First, sequences of minizones are created to generate general routes that satisfy soft and hard restrictions. Then, using these first routes, real bus routes are defined in a network of roads level. Once we have all the necessary information for the algorithm, the sequence and calls of execution are organized in a single executable command.

[1] http://lima.datosabiertos.pe/home/

[2] https://www.itdp.org/library/standards-and-guides/the-bus-rapid-transit-standard/what-is-brt/

[3] http://archivo.larepublica.pe/11-04-2015/municipalidad-de-lima-castaneda-anuncia-ampliacion-de-la-ruta-lima-norte-del-metropolitano

[4] http://www.aate.gob.pe/metro-de-lima/

[5] http://www.elperuano.com.pe/NormasElPeruano/2015/02/28/1205528-1.html

This work is organized as follows. Section 2 explains the information sources for the database created to keep the data needed and the solutions produced by the algorithm for the TNDP. Section 3 describes the extraction and transformation processes used to get the information mentioned and load it to a default database. Section 4 explains the organization of the files and instructions used in the single process and the order in which they are executed to automatically finish the process. Conclusions of this work are presented in Section 5 along with future recommendations.

2 Sources of Information

To solve a basic case of the TNDP, two sets of data are needed as we can see in Mautone and Uqhuart's work (2009). The first group is the network that generates one of the graphs required to evaluate solutions at a network road level, and information about the edges of this graph like the road owner of those edges and the road's type. The second group consists of the demand information, the number of travels made from an origin zone to a destiny zone and the adjacency graph among zones. Using the Pair Insertion algorithm (PIA), random routes that cover certain percentage of the total demand of the zones are generated. Besides those two sets of groups, Lima has an actual real solution and that is why we decided to test this one as an initial solution. The actual information about routes would be combined using a genetic algorithm in different forms in order to optimize the set of routes that solve the TNDP in both levels of graphs: zones and network.

The next two subsections describe the sources for all the groups of data mentioned.

2.1 Road Network

The two sources analyzed to generate the public transportation network are shown in Figure 1. The first source was the one given by the Ministry of Transportation and Communications of Peru. The problem with this source was that it was difficult to identify the edges and nodes of the network due to the format of the file: shapefile[6] (.shp), i.e. one set of edges was represented as a single edge and vice versa. In order to verify the correctness of the shapefile, it was loaded into OSM and in some cases, the roads were displayed crossing a block. The second analyzed source was OSM, which has user generated geographic content (Janssen and Crompvoets, 2012). After some transformation, OSM produces a valid, but not official, graph that represents the public transportation network. This network would be enough for our TNDP studies. Besides, this contribution on OSM could be used for future studies.

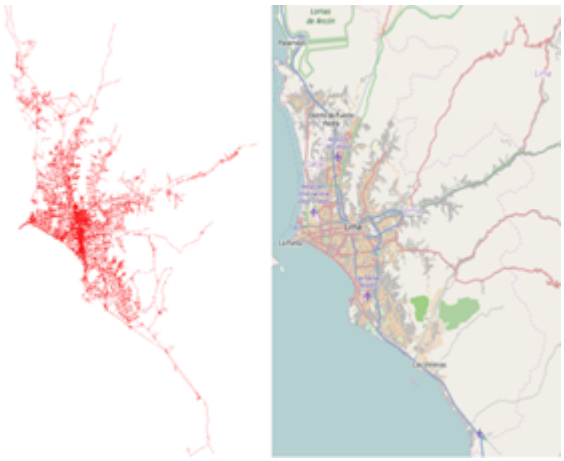


Figure 1: Difference among MTC Network (left) and Network from OSM (right)

2.2 Routes

In order to get the actual public transportation routes, two complementary sources were analyzed. The first source was the Metropolitan Lima Municipality, which has one SHP file with the whole set of routes. A process to get a SHP file per route was made through the *ogr2ogr*[7] command and a batch program. After the command produces a comma separated values file (.csv) which contains a *WKT*[8] (or *LineString*[9]) for each route, the batch program creates a SHP file from every record of the CSV file. Once a route is individually identified, it is drawn manually in the OSM interface and matched with all the ways (set of edges) by which it passes in order to maintain the geometric consistency with the road network. In this process, a lot of errors are generated and they would be solved thanks to daily reports that indicate their coordinates and details.

[6] Geospatial vector data format for Geographic Information System (GIS) Software.

[7] A command line utility from the "Geospatial Data Abstraction Library" (GDAL).

[8] Well-known text, markup language for representing vector geometry objects.

[9] Vector geometry object that represent a line.

2.3 Demand

Due to a study of demand and transportation in Lima made by the Japan International Cooperation Agency (2005), a set of 427 Traffic zones was established in the city of Lima. This information implied two types of data: polygons that represent the zones and the number of travels made among every pair of O-D zones (origin and destiny). The polygons are used to delimit a set of edges for which a route must pass by in order to satisfy the demand of that zone (like an origin or destiny).

3 Transformation process

This section describes the steps followed to extract data from OSM sources, upload it into a database and transform it into a final database called routes.

3.1 Extraction from the sources

In order to get the OSM data about the public transportation network and routes of Lima, the files containing the information of Peru and the boundary of Lima must be downloaded and used in a command line. There is a server named *Geofabrik*[10] which has data extracts of countries of all the continents and is updated daily (Zielstra and Zipf, 2010). However, as this server provides the information of the whole country, a boundary must be applied to get the information of only the city of Lima. This can be done through an OSM relation ID which can be obtained from the *MapIt*[11] website, through the insertion of a coordinate (latitude, longitude) of the objective city. After that, the OSM relation ID is introduced in a polygons generator page[12] and the .poly file is downloaded.

Osmosis is the command line application from OSM with which a file containing the information of the primary, secondary and tertiary highways of Lima is produced as well as the trunk, motorway, residential and their links. Figure 2 shows how the osmosis command takes as input the information of Peru (peru-latest.osm.pbf) and the boundary of Lima (lima-callao.poly) and produces the lima-callao.osm file.

3.2 Default and final databases

Once the lima-callao.osm file is produced, it is necessary to upload this information to a database to manage the information. For this task, a tool

```
osmosis --rb file=peru-latest.osm.pbf
--bounding-polygon file=lima-callao.poly
--way-key-value keyValueList="highway.primary,
highway.secondary, highway.tertiary, highway.trunk,
highway.motorway, highway.residential,
highway.motorway_link, highway.trunk_link,
highway.primary_link, highway.secondary_link,
highway.tertiary_link" --write-xml lima-callao.osm
```

Figure 2: Osmosis command to get .osm file

like *osm2pgrouting*[13] was used. It provides a process that converts OSM data into a topology and it is uploaded in database. First, we must create a database called *pgrouting-workshop*[14]. After that, the command shown in Figure 3 must be executed.

```
osm2pgrouting -file lima-callao.osm
-conf mapconfig.xml -dbname pgrouting-workshop
-user postgres -clean
```

Figure 3: Osm2pgrouting command to create pgrouting-workshop database

This command filters the road types mentioned in mapconfig.xml and create tables like *ways* (edges), *classes* (types of roads), *nodes*, *relations* (routes for example), *relation_ways* (which relates the edges of a route), *types*, *way_tag*, *way_vertices_pgr*. The full schema for this database is shown in Figure 4.

Certainly, there are other tools to import OSM data into a local database (*osm2pgsql*, *imposm* and *osmosis*). However, only *osm2pgrouting* builds an exact graph with edges and nodes, and Section 3 will show how that makes a difference.

In order to have the direct information of the sources and the information of the application or algorithm separated, a new database *routes* is created. Table 1 shows the source of every table in the new database (from default *pgrouting-workshop* database or from external data).

3.3 Transformation

In this section, a brief logic of the load of every table would be shown. See the Figure 5 for more details. every table would be shown.

The table *road_types* contains the different types of roads that are presented in Lima's OSM data

[10] http://download.geofabrik.de/
[11] http://global.mapit.mysociety.org/
[12] http://polygons.openstreetmap.fr/

[13] http://pgrouting.org/docs/tools/osm2pgrouting.html
[14] ftp://ftp.remotesensing.org/pgrouting/foss4g2010/workshop/docs/html/chapters/topology.html

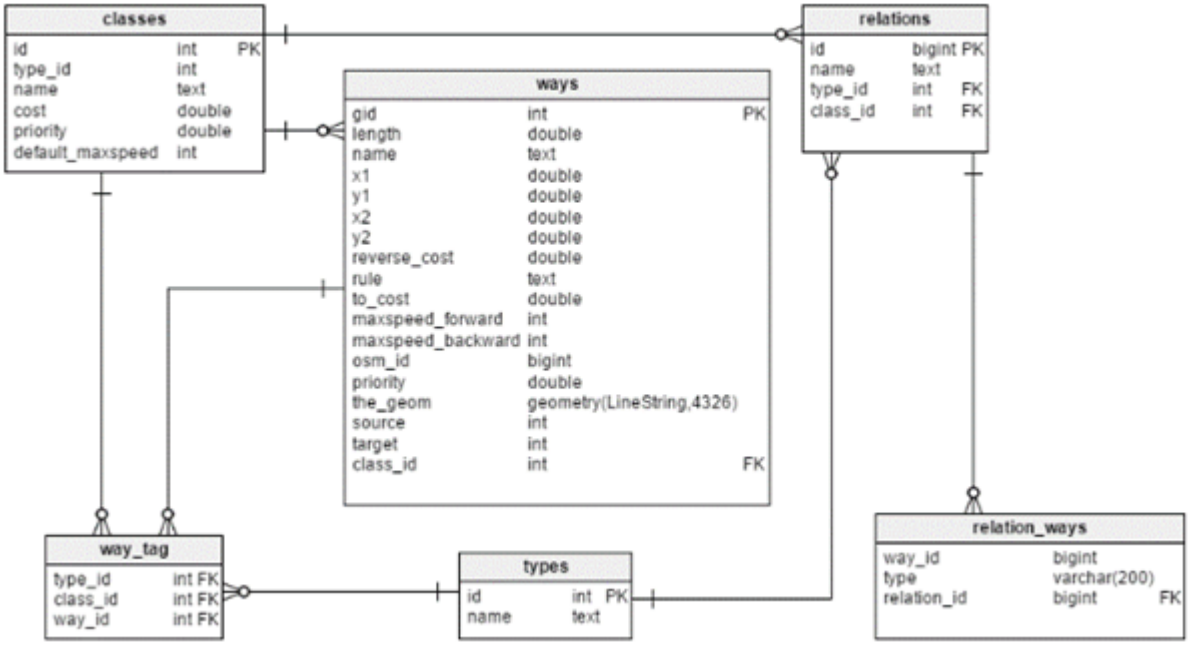


Figure 4: Pgrouting-workshop schema database

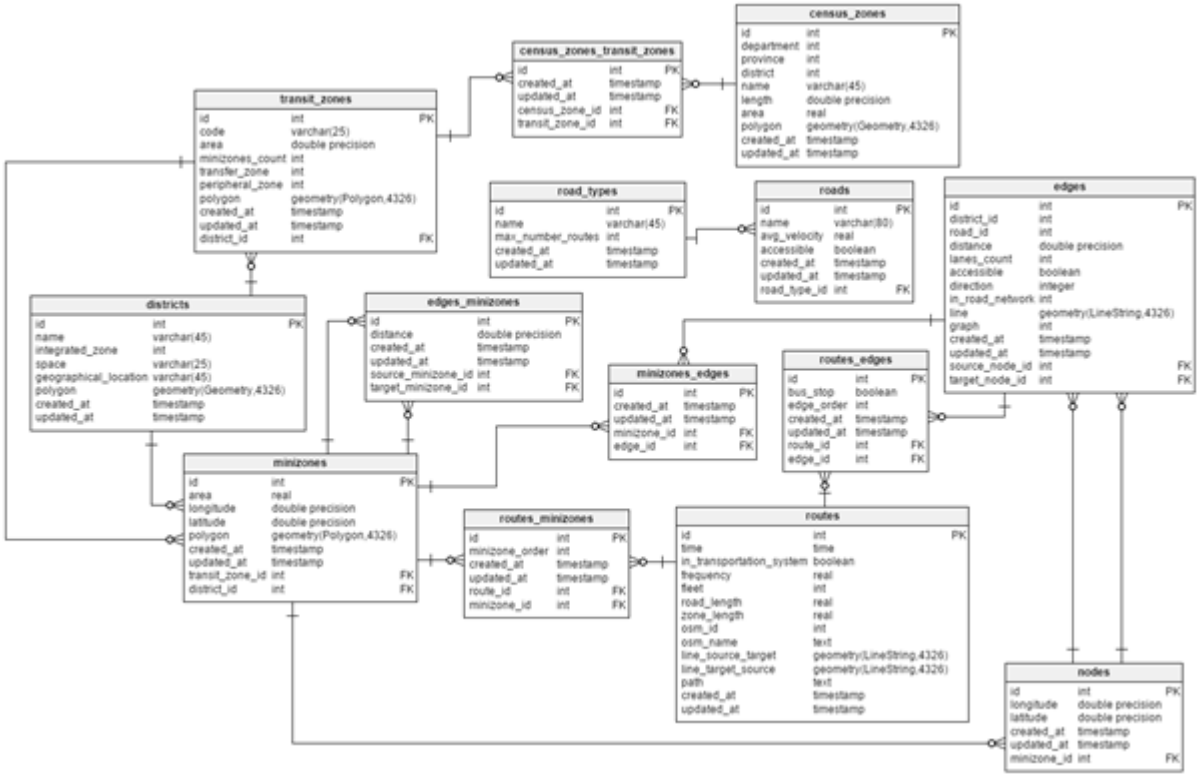


Figure 5: Routes schema database

Entity	Database	External File
RoadType	Classes (default)	
Road	Ways (default)	
Node	Ways (default)	
Edge	Ways (defaut)	
District		i4_districts.csv
Minizone		i8_census_zones.csv
Demand		demand_matrix.csv
Route		lima-callao.osm
RouteEdge	Ways (default)	list_final_routes.csv
RouteMinizone	routes, routes_edges, edges, transit_zones (routes)	

Table 1: Input and output of tables from routes database.

like primary, secondary and tertiary road among others. Besides, this table has an additional field (not from OSM) that indicates the maximum number of routes that would be used in the future when solving the TNDP.

The table *roads* is filled from the table ways of the *pgrouting-workshop* database, which contains the longitude and latitude of both nodes that form an edge (known as source and target). However, as a road is formed by several edges or ways, a distinct query by the name of the way is made to obtain unique roads.

The table *nodes* is filled by searching every way and saving each source or target as a unique key in a hash table. Besides that, a function from the *PostGIS* extension is applied to define which minizone every node belongs to. Additionaly, the *edges* table is similar to the table ways (on *pgrouting-workshop* database) but the *road_id* is brought from the previous step (*roads* table) to complete its load.

We use several steps to fill table *routes*. First, a file is generated from the lima-callao.osm file containing just the relations-routes from the users of the project. After that, just the routes which do not have any errors are listed in a CSV file and they are finally uploaded to the table.

The process to define which edges belong to certain route has been a continual feedback. First, a hash table of different sources and targets nodes from the table *ways* was created. Second, every node was linked to its respective previous and next node. Each node has its own *edge's gid*[15], relative to the current edge. After that, a search is made to identify which nodes (source or target) are used in the graph.

[15]Road link if of the table ways.

Finally, a whole loop is made to search along the hash table from the start node and get the edge's to which the actual node belongs to and its respective order. It is important to mention that there are some mistakes in this logic due to some errors or missing information in the direction of the ways (edges).

Based on the filling of the route's edges, the logic to fill the table *routes_minizones* is to analyze the source and target node of every edge that form the route. As every node belongs to a minizone, a list of every minizone that contains a route's nodes is made. Moreover, this list is ordered by the edge's order calculated in the previous step as an attribute of the table *routes_edges*. This list is grouped by the *minizone_id* to avoid the repetition of a minizone on different edges. This logic is applied using some functions of the *PostGIS* extension like *ST_Contains(polygon, point)* that decides whether a point is contained in a polygon or not and *ST_SetSRID(point, system)* that sets the 4326 system reference[16] of a point. This logic is better shown in Figure 6.

When nodes are on the limit of the polygon like the boundary of a demand zone, they could belong to more than one zone. However, this work did not analyzed which minizone is selected by the function *ST_Contains* from the *PostGIS* extension.

4 Results: Automatic process organization

Before getting a stable database in which you can execute an algorithm to the TNDP, several databases loads must be done in order to evaluate the accuracy of the routes drawn manually. That

[16]`http://suite.opengeo.org/opengeo-docs/glossary.html`

```
SELECT minizones.id, MIN(routes_edges.edge_order)
FROM routes_edges, edges, nodes source_nodes, nodes target_nodes, minizones
WHERE routes_edges routes_edges.edge_id = edges.id AND
      edges.source_node_id = source_nodes.id AND
      edges.target_node_id = target_nodes.id AND
      (ST_Contains(minizones.polygon,
        ST_SetSRID(
         ST_MakePoint(source_nodes.longitude, source_nodes.latitude),
         4326
        )
       )
      OR
       ST_Contains(minizones.polygon,
        ST_SetSRID(
         ST_MakePoint(target_nodes.longitude, target_nodes.latitude),
         4326
        )
       )
      )
GROUP BY minizones.id
ORDER BY 2
```

Figure 6: SQL script to fill table minizones

is the reason why an automatic process was set to run daily. In Figure 7, a sequence of programs, commands, input and output files are shown to explain the process of downloading the information from OSM, combine it with external information and load them to a final database.

```
python DownloadAndBoundary.py
./2_pgrouting/osm2pgrouting.sh
python TransformEntities.py
python 3_scripts/python/RouteReports.py
python UploadFinalDB.py
```

Figure 7: Content of executable final.sh

Some tools must be installed in the server and the local computer before executing this process. These are: *gdal*, *osmosis*, *PostGIS*, *pgrouting*[17] and *psycopg2*.

The automated process has a series of steps called from an executable file (final.sh) in Linux. It lasts 30 minutes approximately and the topology (graph) size is about 150000 edges and 100000 nodes; the number of routes is 300. The content and the structure of the commands and process called from final.sh are shown in Figure 8. Also, the process generates error reports about the current drawn routes, for each one evaluates how many edges exist in each node, so if more than two edges exist in one node, then it reports that the route has an error. This process was carried out daily until no error is found in the routes. It is important to mention that this process is recommendable when working in a local database where the password could be stored in files to allow the interaction of calculus inside and outside the database.

[17] http://pgrouting.org/

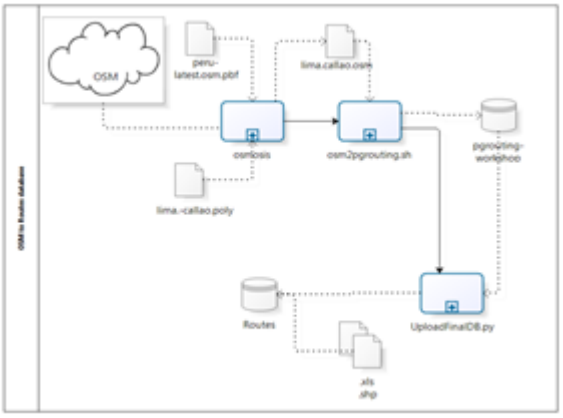


Figure 8: Diagram of process of downloading data, transforming it and uploading it to the database (Tool: Bizagi)

5 Conclusions and recommended work

This section presents the conclusions after implementing an automated process for uploading OSM data combined with external information. One of them was the confirmation of a tool that generates the topology of the map or graph rather than other tools that also upload the same data but in a different scheme.

5.1 Conclusions

In subsection 3.1, it was mentioned that there were other tools to upload OSM data. *Osm2pgsql* and *imposm*, after installed and executed, generate a table where the information of ways can be found. However, the field that represents the geometry does not generate the sequences of edges. Actually, *osm2pgsql* generates edges that are not necessary for the graph and make impossible to distinguish which ones are. We could have worked with edges that were not required but it would have been an unnecessary addition of data to a problem that is already complex. In addition to that, *imposm* generates the same edges composed of only the first and last node of the way. The manner these edges are stored in these schemes (*osm2pgsql* and *imposm*) hinders the recognition of edges in a way as seen in Figure 9 where just three edges (*osm2pgrouting*) should be generated from the selected way instead one (*imposm*) or seven (*osm2pgsql*).

On the other hand, there are a lot of routing applications that use OSM data to combine it with other type of information at some point (Amat et al., 2014)(Vetter, 2010). However, some of them

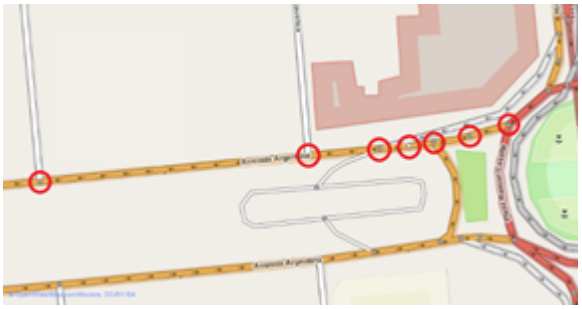


Figure 9: Shows the numbers of nodes counted in one way: Argentina avenue - Lima, Peru (Obtained from OSM and modified) (Tool: Bizagi)

are implemented just for certain cities, for other routing problems like private cars and bicycles or just do not work very well[18]. We found some commercial routing applications but obviously they do not public the process to combine their sources into a unique database.

5.2 Recommended work

As OSM is a free collaborative tool, more analysis is recommended in order to establish several logics that allow us to maintain the coherence of the data despite of new errors.

A full review of the possible scheme obtained and filled from *osmosis* should be finished in order to know if there is a faster and simpler way of loading the information. However, it seems that there is no direct form to identify the relations routes with the osmosis' scheme. This is because of the different order that the *route* tag has in a field to recognized that a relation is a public transportation route.

There is also another tool that seems to convert OSM data into a graph topology that must be analyzed: *OSM2PostGIS*[19]. However, it is still in an early development.

References

Japanese International Cooperation Agency. 2005. Plan maestro de transporte urbao para el área metropolitana de lima y callao en la república del perú fase 1–9. problemas y temas actuales del transporte urbano.

Guillermo Amat, Javier Fernandez, Álvaro Arranz, and Angel Ramos. 2014. Using open street maps data and tools for indoor mapping in a smart city scenario.

Wei Fan and Randy B Machemehl. 2004. Optimal transit route network design problem: Algorithms, implementations, and numerical results. Technical report.

Reza Zanjirani Farahani, Elnaz Miandoabchi, WY Szeto, and Hannaneh Rashidi. 2013. A review of urban transportation network design problems. *European Journal of Operational Research*, 229(2):281–302.

Katleen Janssen and Joep Crompvoets. 2012. *Geographic Data and the Law: Defining New Challenges*. Leuven University Press.

Antonio Mauttone and María E Urquhart. 2009. A route set construction algorithm for the transit network design problem. *Computers & Operations Research*, 36(8):2440–2449.

Christian Vetter. 2010. Fast and exact mobile navigation with openstreetmap data. *Master's thesis, Karlsruhe Institute of Technology*.

Dennis Zielstra and Alexander Zipf. 2010. A comparative study of proprietary geodata and volunteered geographic information for germany. In *13th AGILE international conference on geographic information science*, volume 2010.

[18] http://wiki.openstreetmap.org/wiki/Routing/online_routers#Route_service_comparison_matrix

[19] http://pgrouting.org/docs/tools/osm2PostGIS.html

Arquitectura de Big Data para la Predicción de la Portabilidad Numérica en Empresas de Telefonía Móvil

Alonso Raúl Melgarejo Galván
Facultad de Ingeniería de Sistemas
Universidad Nacional Mayor de San Marcos
Lima - Perú
alonsoraulmg@gmail.com

Katerine Rocio Clavo Navarro
Facultad de Ingeniería de Sistemas
Universidad Nacional Mayor de San Marcos
Lima - Perú
perclavo@gmail.com

Abstract

Actualmente en el Perú, las compañías de telefonía móvil se han visto afectadas por el problema de la portabilidad numérica, puesto que, desde julio del 2014, los clientes pueden cambiar de operadora móvil en sólo 24 horas. Las compañías buscan soluciones analizando los datos históricos de sus clientes para generar modelos de predicción e identificar qué clientes portarán, sin embargo, la actual forma en la que se realiza esta predicción es demasiado lenta. En el presente trabajo, mostramos una arquitectura de Big Data que resuelve los problemas de las arquitecturas "clásicas" y aprovecha los datos de las redes sociales para predecir en tiempo real qué clientes son propensos a portar a la competencia según sus opiniones. El procesamiento de los datos es realizado por Hadoop el cual implementa MapReduce y permite procesar grandes cantidades de datos en forma paralela. La arquitectura también utiliza otras herramientas de Hadoop como Mahout para generar nuestro modelo de predicción, y Hive para gestionar los datos con una sintaxis similar a SQL. Al realizar las pruebas y observar los resultados, logramos obtener un alto porcentaje de precisión (90.03% de aciertos), y procesar 10'000 comentarios en 14 segundos.

1 Introducción

En el Perú, la pérdida de clientes en la industria de la telefonía móvil es un problema que actualmente afecta a las grandes empresas de telecomunicaciones del país debido a la fuerte competencia que se ha generado en el mercado de servicios móviles de voz y datos, y que ha generado grandes ofertas comerciales y guerra de precios. OSIPTEL (2015) menciona que desde el ingreso de las operadoras móviles Bitel y Entel, la competencia se ha incrementado, siendo Entel la operadora que más clientes ha obtenido de Claro Perú y de Movistar.

Como vemos en la Figura 1, de acuerdo a las últimas cifras de OSIPTEL (2015), se muestra que en marzo, la portabilidad móvil creció en 46%, alcanzando un récord de 65,142 portaciones, el más alto desde julio del año pasado, fecha en la que se relanzó el mecanismo para realizar una portabilidad numérica y cambiar de operadora móvil en solo 24 horas manteniendo el mismo número de teléfono.

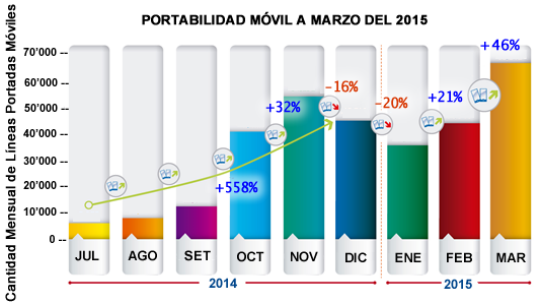


Figura 1: Portabilidad móvil a marzo del 2015

Como (Francisco et al., 2013) menciona, en el área de las telecomunicaciones, la fuga de clientes es un problema que cada vez es más necesario estudiar debido a la alta competitividad que se está desarrollando a nivel mundial. Se hace necesaria la aplicación de herramientas avanzadas que permitan predecir y describir de algún modo, qué clientes tienen mayor potencial de riesgo de cambiarse de compañía.

2 El problema de la portabilidad numérica: La fuga de clientes

La pérdida de clientes es un riesgo costoso que si no se maneja con cuidado podría llevar a la

compañía móvil de rodillas.

Según define (Francisco et al., 2013), dentro del sector de telecomunicaciones, el término churn o fuga de clientes es usado para describir el cese de servicios de la suscripción de un cliente, y se habla de churner o fugado para denominar a un cliente que ha dejado la compañía. Como (Clement et al., 2013) menciona, un cliente puede renunciar a la empresa e iniciar la terminación de su contrato de servicios (churn voluntario), o bien la empresa puede expulsarlo por fraude, falta de pago o por subutilización de los servicios suscritos (churn involuntario).

(Clement et al., 2013) también menciona que la fuga de clientes puede llegar a ser muy costosa para una compañía, ya que el cliente fuga hacia la competencia y por ende, no solo se pierde el ingreso no percibido, sino también el prestigio de la compañía expresado en la participación de mercado de la competencia.

2.1 Causas que ocasionan la fuga de clientes

Los factores que contribuyen al comportamiento de fuga en los servicios de telecomunicaciones son diversos.

(Francisco et al., 2013) menciona que, dentro de las principales razones por las que un cliente deja de adquirir los productos de una compañía se encuentran: la disconformidad con la calidad de la señal, cobertura, servicio al cliente, precios, cobros irregulares y la falta de políticas de retención con un mejor trato a los clientes, pero por otro lado Patricio (2014) también menciona que la red de contactos donde se desenvuelve el cliente es muy importante, pues el efecto "boca a boca" se ha convertido en un factor determinante en las decisiones de compra de un consumidor.

Anticiparse a esta problemática y lograr identificar qué lleva a un cliente a terminar su contrato, entrega diversos beneficios a la compañía como por ejemplo, la menor inversión en retener a un cliente (gracias a la recomendación de la red de contactos). Patricio (2014) afirma que adquirir un nuevo cliente cuesta seis veces más que retener a uno propio, y que los clientes que se mantienen más tiempo en la empresa generan mayores ingresos y son menos sensibles a las acciones de marketing de la competencia convirtiéndose en consumidores menos costosos de servir.

2.2 La influencia de la opinión dejada en la web

Hoy en día, los avances tecnológicos como el internet y las redes sociales permiten que el cliente tenga mayor acceso a la información y pueda comparar fácilmente la compañía que más le convenga.

Bajo el contexto de la competitividad que actualmente se vive en el mercado de telecomunicaciones, y sumándose a ello, el mayor acceso a la información, Patricio (2014) afirma que se genera un marco de flexibilidad y dinamismo respecto a la movilidad de los clientes de una compañía a otra.

Como menciona Tania (2011), la dinámica de participación social y la influencia que pueden tener las valoraciones dejadas por los consumidores en internet, han hecho que las empresas del mercado presten atención a la gestión de las opiniones que se dejan en la web sobre ellas. El community manager de una empresa debe ser rápido en la resolución de los conflictos que percibe en la web. Un conflicto que tarda un día en resolverse, probablemente se convertirá en un conflicto no resuelto, y en muchos casos propiciará una fuga de clientes hacia la competencia, afectando igualmente la reputación online de la empresa. El community manager debe tener criterio para destacar aquellos comentarios positivos, negativos o notables, que por alguna razón, merezcan la ejecución de alguna estrategia especial.

Patricio (2014) menciona que las redes sociales pueden influenciar distintos aspectos de una persona como por ejemplo contratar un servicio, comprar un producto o abandonar una compañía, mediante el efecto "boca a boca".

(Yiou et al., 2015) también afirma que las opiniones de los clientes tienen un impacto en los productos y los servicios, por eso es necesario capturar estas opiniones por medio de calls centers, correos, cuestionarios o webs para entender las necesidades de los clientes. Es por esto que el explotar la "voz del consumidor" debe ser considerado en la predicción de abandono de clientes.

2.3 La solución actual

Actualmente las empresas de telefonía móvil llegan a desarrollar modelos predictivos para identificar a los clientes que pueden llegar a portar hacia la competencia. La construcción de estos modelos puede variar, pero en general, se siguen los pasos mostrados en la Figura 2, como nos explica

(Clement et al., 2013) a continuación.

Primero se deben identificar las fuentes de datos con las que se construirá el modelo de predicción, la cuales corresponden a los datos internos de la empresa referentes al perfil del cliente y el tráfico de llamadas.

El perfil del cliente describe el grupo demográfico de los clientes y las características que tienen en común respecto al segmento, el plan contratado, el tipo de pago, el riesgo crediticio, el área demográfica y las penalidades por no pagar las cuentas. El tráfico de llamadas describe el tráfico generado y recibido por un cliente, el cual es caracterizado por las llamadas realizadas y recibidas desde líneas fijas o móviles, locales o internacionales, desde qué operador se realizó, la cantidad de SMS enviados y recibidos, y el tráfico de red en internet generado.

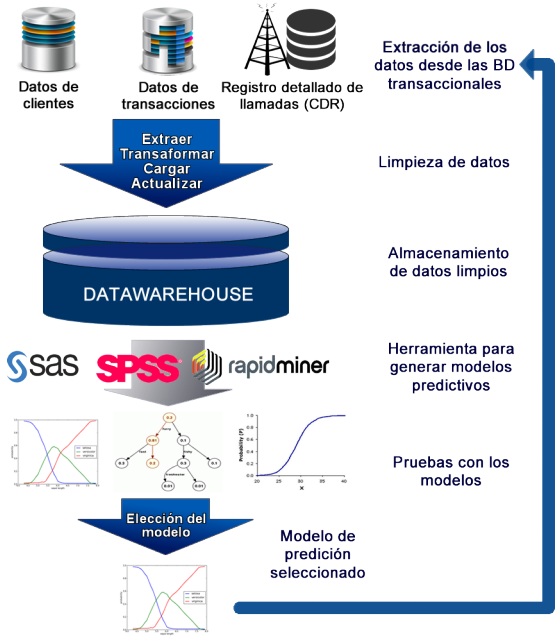


Figura 2: Pasos en el desarrollo de modelos de predicción

Luego de identificar las fuentes de datos con las que se trabajará, debe realizarse una extracción y limpieza de ellos. La limpieza de datos consiste en encontrar errores en los datos, duplicidad, incoherencias o valores incompletos e inconsistentes. Esta información es corregida reemplazando valores generados, o eliminada en el peor de los casos. Todos estos datos son almacenados en un Datawarehouse.

Una vez que los datos han sido limpiados y están listos para ser usados, se utilizan diferentes algoritmos de minería de datos existentes como Naive Bayes, árboles de decisión o redes bayesianas, para construir modelos de predicción. Cuando los modelos son generados, se realizan pruebas sobre ellos para encontrar aquel que tenga el mejor ratio de predicción, el cual será el modelo a usar para predecir la portabilidad.

Por supuesto, este no es el paso final, ya que el comportamiento de los clientes puede cambiar a lo largo del tiempo, por lo que es necesario actualizar el modelo volviendo a repetir todos los pasos.

Ahora que entendemos cómo se generan actualmente los modelos de predicción, veamos cuáles son los problemas que poseen.

2.4 Problemas de la solución actual

Dado el entorno actual en el cual las empresas de telefonía móvil se desenvuelven, la solución que actualmente utilizan para predecir la fuga de clientes presenta cuatro problemas:

P1. Portabilidad rápida, predicción lenta: Como menciona OSIPTEL (2015), la nueva ley de portabilidad hace que los clientes puedan realizar una portabilidad hacía la competencia en 24 horas, así que se necesita predecir esta portabilidad de manera rápida para evitarlo. La solución clásica requiere mantener un Datawarehouse con datos limpios y filtrados, lo cual consume demasiado tiempo.

P2. Confidencialidad de datos: La solución clásica trabaja principalmente con los datos internos de la empresa puesto que están estructurados y disponibles, sin embargo, debido a la confidencialidad y privacidad del negocio de las telecomunicaciones, es muy difícil para las operadoras encontrar fuentes de datos públicas que sean fidedignas y puedan usarse como un input adicional en la predicción de la portabilidad de sus clientes, como menciona (Clement et al., 2013).

P3. Datos no estandarizados e incoherentes: Estandarizar las características de las diversas fuentes de datos usadas para realizar un análisis es un reto ya que consume demasiado tiempo y esfuerzo. Es necesario implementar y mantener un Datawarehouse para realizar análisis de datos, lo cual incluye eliminar información irrelevante, duplicada o con valores nulos como menciona (Clement et al., 2013).

P4. Opinión cambiante: No se analiza la opinión cambiante que tienen los clientes respecto a un servicio, lo cual es un factor determinante

en sus decisiones como menciona Patricio (2014). Para poder analizar el cambio en las opiniones, deben incluirse fuentes de datos adicionales a los perfiles de clientes y el tráfico de llamadas.

Estos cuatro problemas ocasionan que la identificación de clientes portadores sea lenta e inexacta, pues se requiere realizar una limpieza de los datos o un análisis y ordenamiento de sus fuentes, lo cual demuestra que la implementación de una arquitectura tradicional para predecir la portabilidad numérica tiene carencias que deben resolverse con un enfoque diferente.

3 Un enfoque diferente: Big Data

La necesidad de procesar grandes volúmenes de datos en tiempo real es crucial para las empresas de hoy en día, por ejemplo, el procesamiento de volúmenes masivos de datos donde se esconde información valiosa respecto al comportamiento de compra de productos o servicios de clientes, y poder generar nuevos productos analizando dichos comportamientos. Esto último es particularmente cierto en el mercado de los negocios de las telecomunicaciones, donde el número de clientes normalmente llega a varios millones como menciona (Francisco et al., 2013).

En estos escenarios es donde entra Big Data, el cual es un campo emergente de tratamiento de datos que permite analizar grandes cantidades de datos y maximizar el valor del negocio dando un soporte en tiempo real a la información necesaria para la toma de decisiones, según define (M. Vasuki et al., 2014). Por otro lado, (Kankana et al., 2014) también nos dice que la invención de nuevas tecnologías ha llevado a la utilización de grandes cantidades de datos que van en aumento, además se ha creado la necesidad de conservar, procesar y extraer estos datos. De esto se encarga el término Big Data.

Para tener un mayor conocimiento sobre Big Data, en esta sección se verá el concepto de las 5 V de Big Data, el aprovechamiento de las nuevas fuentes de datos, el procesamiento paralelo masivo que ofrece MapReduce y la implementación más popular que tiene llamada Hadoop.

3.1 Las 5 V

Big Data se caracteriza tradicionalmente por el concepto de las 3 V: volumen, variedad y velocidad como menciona (Mario et al., 2013), pero (Abdullah et al., 2015) también menciona que actualmente, se adicionan 2 V más: variabilidad y veracidad, dando así un total de 5 V.

El **volumen**, según (Mario et al., 2013), es la dimensión más obvia al caracterizar grandes colecciones de datos creadas para diferentes usos y propósitos. El almacenamiento de Big Data supone el reto más inmediato, ya que su primera responsabilidad es la de preservar todos los datos generados en el ámbito de los sistemas transaccionales. La decisión de cómo se almacenan los datos tiene un impacto considerable en el rendimiento de los procesos de recuperación, procesamiento y análisis de Big Data.

La **velocidad**, según (Mario et al., 2013), caracteriza los flujos de datos desarrollados en entornos cada vez más distribuidos. Se pueden distinguir dos tipos de flujos: los flujos de nuevos datos y los flujos que contienen resultados generados por consultas. La velocidad describe lo rápido que se generan, demandan y entregan los datos en su entorno de explotación.

La **variedad**, según (Mario et al., 2013), se refiere a los diferentes grados de estructura o falta de ella que pueden encontrarse en una colección de datos. La colección puede integrar datos procedentes de múltiples fuentes, por ejemplo: redes de sensores, logs generados en servidores, redes sociales, datos de origen político, económico o científico, entre otros. Cada una de estas fuentes de datos tiene esquemas diferentes que son difícilmente integrables en un modelo único, por lo tanto, el manejo efectivo de la variedad pasa por encontrar un modelo lógico que facilite la integración de los datos, independientemente de su estructura.

La **veracidad**, según (Abdullah et al., 2015), hace referencia a la incertidumbre que hay en las fuentes de datos y su nivel de confiabilidad. La veracidad de una fuente de datos disminuye si en ella existen inconsistencias y datos incompletos. Es necesario trabajar con datos precisos, no falsos, no corruptos y que provengan de una fuente de datos fidedigna.

Por último, la **variabilidad**, según (Abdullah et al., 2015) se refiere al cambio que tienen los datos a lo largo del tiempo, tanto en su estructura como en su contenido.

También hay que considerar lo que recomienda (Mario et al., 2013): cualquier arquitectura diseñada para la gestión de Big Data debe afrontar las variables anteriores, sin embargo, la decisión

de cuál de ellas afrontar en primer lugar depende del entorno de explotación final de la arquitectura, por ejemplo, optimizar el almacenamiento de datos es un aspecto más crítico para una arquitectura destinada a un dispositivo móvil, que para una que será ejecutada en un servidor de alto rendimiento; la velocidad con la que se recuperan los datos es una prioridad para una arquitectura en tiempo real, pero no lo es tanto para una de procesamiento en batch. Por lo tanto, una arquitectura para Big Data debe priorizar las cinco dimensiones anteriores con el objetivo de cubrir de forma efectiva los requisitos con los que se diseña.

3.2 Nuevas fuentes de datos: Web y Redes Sociales

Los datos en forma de texto dentro de internet están creciendo cada vez más y es imposible analizar estos datos de forma manual debido a su ingente cantidad. Es aquí donde la necesidad de la automatización se hace evidente. (Sunil et al., 2014) dice que en la web los usuarios tienen la oportunidad de expresar sus opiniones personales sobre tópicos específicos dentro de blogs, foros, sitios de revisión de productos y redes sociales.

Como indica (M. Vasuki et al., 2014), este crecimiento explosivo de la información textual en la web ha traído un cambio radical en la vida humana. En la web la gente comparte sus opiniones y sentimientos, lo cual crea una gran colección de opiniones y puntos de vista en forma de texto que pueden ser analizados para conocer la eficacia de los productos y los servicios.

3.3 Procesamiento paralelo: MapReduce

Es un framework de programación creado por Google para computación distribuida que utiliza el método de "Divide y Vencerás" para analizar grandes conjuntos de datos complejos y que garantiza la escalabilidad lineal. Utiliza dos funciones para procesar los datos: la función Map y la función Reduce. El funcionamiento de estas dos funciones puede verse en la Figura 3.

Como explica (G. Bramhaiah et al., 2015), la función Map divide los datos de entrada en subpartes más pequeñas. Estas partes son distribuidas a lo largo de los servidores para que sean procesadas por separado. Luego la función Reduce recolecta las respuestas de todas las subpartes y las combina en una única salida. MapReduce divide el procesamiento de un algoritmo en etapas paralelizables que se ejecutan en muchos nodos,

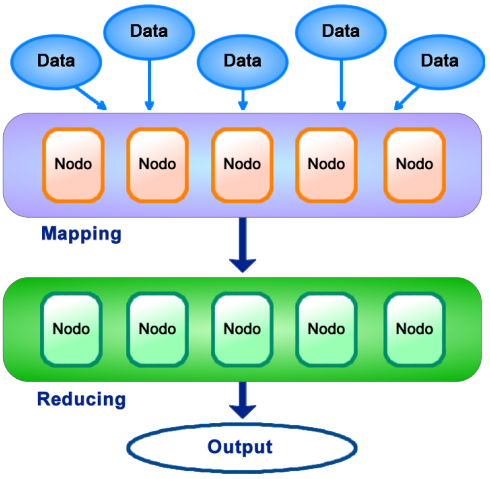


Figura 3: Procesamiento paralelo de datos con MapReduce

así como en etapas de agregación donde los datos obtenidos en la fase previa son procesados en un único nodo. De esta manera se facilita el almacenamiento y procesamiento del llamado Big Data. La herramienta más extendida según (Pablo et al., 2014) basada en el modelo MapReduce es Hadoop, la cual se explica a continuación.

3.4 La plataforma Hadoop

Según Dhruva (2014), Hadoop es un framework open-source para el almacenamiento y procesamiento de grandes cantidades de datos, sobre clústers que funcionen sobre hardware comodity.

(Debajyoti et al., 2014) agrega que una plataforma típica de Big Data basada en Hadoop incluye el sistema distribuido de archivos HDFS, el framework MapReduce de computación paralela, un sistema de alto nivel para la administración de datos como Pig o Hive, y Mahout como el módulo para el análisis de datos. Todo lo anterior es mostrado en la Figura 4.

(Kankana et al., 2014) explica que el HDFS es básicamente una estructura maestro/esclavo. Al maestro se le conoce como "Name Node", y a los esclavos como "Data Nodes". El trabajo principal del "Name Node" es almacenar metadatos de los datos, esto incluye la localización de los archivos que los contienen y también los diferentes atributos de los documentos. Los "Data nodes" se encargan de almacenar los datos en bloques en los diferentes nodos del clúster.

MapReduce, según (Debajyoti et al., 2014) es un módulo que Hadoop incorpora para el procesamiento paralelo de datos. Para que los progra-

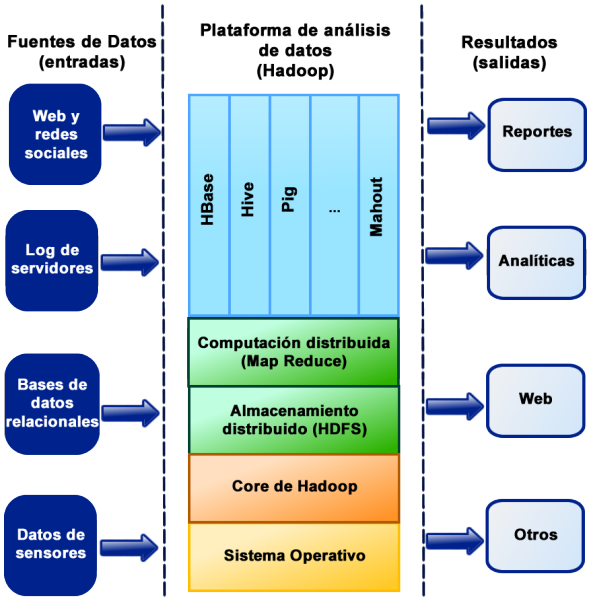


Figura 4: Plataforma Hadoop

madores puedan escribir programas sobre Hadoop deben de especificar las funciones Map y Reduce para que Hadoop las ejecute. Hadoop divide en muchos pequeños fragmentos la entrada la cual distribuye y procesa de forma paralela los nodos del clúster por medio de la función Map, luego por medio de Reduce combina los resultados. Cuando el procesamiento finaliza, el resultado puede residir en múltiples archivos.

Hive, según (Rakesh et al., 2014) es un módulo de Hadoop que soporta el manejo de archivos sobre HDFS por medio una sintaxis similar a SQL, el "Hive Query Language". Las consultas de Hive utilizan el módulo de MapReduce de Hadoop para ejecutarse de manera paralela y ocultar su complejidad al programador. Gracias a esto, es posible usar sencillas sentencias sobre los archivos ubicados en HDFS como "CREATE TABLE", "INSERT", "UPDATE", "SELECT", "DELETE", "JOIN", "GROUP BY", y otras sentencias válidas en el SQL clásico.

Mahout, según (Seyyed et al., 2014) ha sido diseñado para propósitos de minería de datos dentro de Hadoop, pues implementa los algoritmos de clustering y regresión más comunes por medio de MapReduce. Además Dhruva (2014) menciona que Mahout provee herramientas para encontrar de manera automática patrones en grandes volúmenes de datos y hace mucho más fácil y rápido el análisis dentro de Big Data.

4 El problema de la portabilidad numérica desde la perspectiva de Big Data

Después de haber explicado el problema de la portabilidad numérica y el enfoque de Big Data frente a las soluciones tradicionales, veremos cómo Big Data ofrece una solución para cada uno de los problemas encontrados en la portabilidad numérica. Analizaremos cómo cada problema está relacionado con una V de Big Data.

La Figura 5 muestra el cuadro en el que se relacionan a las 5 V de Big Data con cada uno de los problemas encontrados en la subsección 2.4. La explicación del por qué hacemos esta relación entre los problemas y las V es dada a continuación:

	Volumen	Velocidad	Variedad	Variabilidad	Veracidad
P1		X			
P2			X		X
P3	X		X		
P4				X	

Figura 5: Relación entre los problemas de la solución actual y las 5 V

El **primer problema** referido a la rapidez con la que un cliente puede fugar hoy en día a la competencia está relacionado a la V de "velocidad", puesto que se necesita identificar a estos clientes de inmediato para evitar que vayan a la competencia, y como solución, Big Data procesa y entrega la predicción en tiempo real.

El **segundo problema** referido a la confidencialidad de los datos está relacionado a las V de "variedad" y "veracidad" puesto que las soluciones clásicas trabajan principalmente con los datos internos de la empresa, ignorando las fuentes públicas como los sitios web que contienen datos no estructurados pero muy valiosos. Por supuesto que si la empresa decide utilizar datos públicos para realizar sus predicciones, debe tener la certeza de que éstos son fidedignos. Como solución, Big Data permite trabajar con nuevas fuentes de datos como las opiniones dejadas por los clientes en las redes sociales de la compañía móvil acerca del servicio, y poder predecir quienes tienen intención de portar.

El **tercer problema** referido a los datos no estandarizados e incoherentes, está relacionado a las V de "volumen" y "variedad". Los datos usados deben estar limpios y estandarizados antes de

comenzar un proceso de análisis; no puede trabajarse con los datos transaccionales directamente, sino que deben formatearse primero. Como solución, el enfoque de Big Data procesa los datos sin necesidad de realizar un proceso de limpieza.

Por último, el **cuarto problema** referido a la opinión cambiante de los clientes acerca del servicio vendría a estar relacionado con la V de "variabilidad", pues el enfoque de Big Data detecta los cambios que hay a lo largo del tiempo en los datos.

Como vemos, estos cuatro problemas son solucionados desde el enfoque de Big Data visto en la sección 3, por lo tanto, es posible construir una arquitectura de Big Data que implemente una solución para prevenir la portabilidad numérica. También, como se mencionó en la subsección 3.1, es importante reconocer cuál de estas 5 V tiene mayor importancia en la arquitectura. Se espera que la arquitectura de Big Data prediga lo más rápido posible qué clientes pretenden portar, por lo tanto la arquitectura debe ser diseñada tomando como premisa principal la velocidad.

5 Una arquitectura de Big Data para solucionar el problema de la portabilidad numérica

Luego de haber visto la relación que existe entre los problemas de la solución clásica y cómo Big Data y las 5 V ofrecen una solución a cada problema, pasamos a explicar los detalles de la arquitectura que se plantea en el presente trabajo.

5.1 Descripción de la solución

La solución propuesta en el presente trabajo aprovechará las nuevas fuentes de información con las que Big Data trabaja, específicamente las de redes sociales, explicadas en la subsección 3.2. El objetivo de la solución propuesta es obtener en tiempo real los comentarios dejados en las fanpages oficiales de Facebook de las operadoras móviles, y analizar el sentimiento expresado para poder predecir si un cliente tiene intención de realizar una portabilidad hacia la competencia.

5.2 Componentes de la arquitectura

En la Figura 6 mostramos la arquitectura diseñada en el presente trabajo para poder predecir la portabilidad numérica de clientes. Fue diseñada tomando como referencia la subsección subsección 3.4.

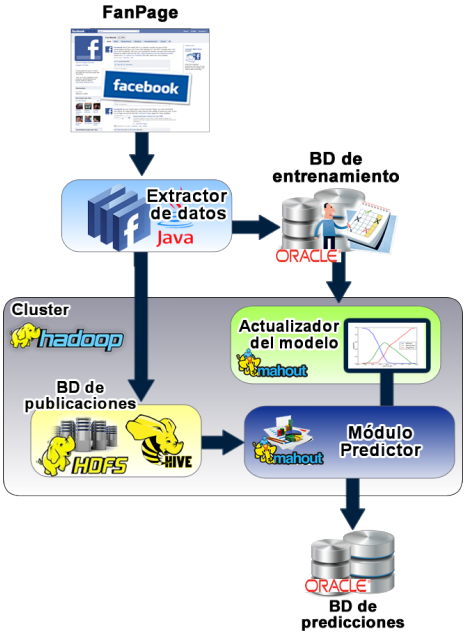


Figura 6: Arquitectura de Big Data para la predicción de la portabilidad numérica

Esta arquitectura está conformada por seis componentes:

El **módulo "extractor de datos"** es el encargado de revisar la página oficial de Facebook de la operadora móvil y descargar las publicaciones y/o comentarios que en ella se realicen. El módulo está encargado de extraer los datos cada hora y almacenarlos en la "base de datos de publicaciones". El módulo extractor de datos también es ejecutado para generar la "base de datos de entrenamiento" y entrenar al modelo. Está implementado en Java y utiliza el API de OpenGraph para conectarse a Facebook.

La **base de datos de publicaciones**, la cual almacena las publicaciones y/o comentarios de Facebook, guarda los datos de manera distribuida por el HDFS de Hadoop y utiliza Hive para gestionarlos, permitiendo trabajar con una sintaxis similar a SQL.

La **base de datos de entrenamiento** es el componente que almacena las publicaciones que servirán para entrenar al modelo de predicción. Cada publicación y/o comentario almacenado aquí debe indicarse si es "negativo", en el caso de opiniones que muestren intenciones de portar, o "positivo", en caso contrario. Esta base de datos está implementada como una base de datos relacional Oracle.

El **módulo "actualizador del modelo"**, es el encargado de generar y actualizar el modelo de

predicción usando como entrada la base de datos de entrenamiento. Está implementado en Mahout para aprovechar el procesamiento paralelo de MapReduce en Hadoop.

El **módulo "predictor"** tiene implementado el modelo de predicción y se encarga de predecir si el comentario de un cliente indica una portabilidad o no. También está implementado en Mahout para aprovechar el procesamiento paralelo de MapReduce en Hadoop.

Finalmente, se encuentra **la base de datos de predicciones** que almacena los resultados obtenidos por el módulo predictor, implementada en Oracle.

5.3 Algoritmo de predicción

El algoritmo de predicción usado en la arquitectura de Big Data es el de "Naive Bayes". Según (Shruti et al., 2014), en el aprendizaje automático, el clasificador Naive Bayes es parte de la familia de los clasificadores de Bayes, pero asume la independencia de las variables y gracias a esto el cálculo de las probabilidades se simplifica. Una ventaja de este clasificador es que requiere una pequeña cantidad de datos de entrenamiento para estimar los parámetros necesarios para la clasificación. Para el cálculo de probabilidades se usan las fórmulas mostradas por (Shruti et al., 2014) adaptándolas a nuestro caso.

$$P\left(\begin{matrix} palabra_i \\ es\ positiva \end{matrix}\right) = \frac{Cantidad\ de\ publicaciones\ positivas\ con\ palabra_i}{Cantidad\ de\ publicaciones\ positivas}$$

$$P\left(\begin{matrix} palabra_i \\ es\ negativa \end{matrix}\right) = \frac{Cantidad\ de\ publicaciones\ negativa\ con\ palabra_i}{Cantidad\ de\ publicaciones\ negativas}$$

Figura 7: Probabilidad de que una palabra sea positiva o negativa

Primero, con las fórmulas mostradas en la Figura 7, evaluamos el conjunto de publicaciones de entrenamiento y obtenemos la probabilidad de que cada palabra sea positiva o negativa.

Luego, con las fórmulas mostradas en la Figura 8, evaluamos todo el conjunto de publicaciones de entrenamiento y calculamos la probabilidad de que el dataset sea positivo o negativo.

$$P\left(\begin{matrix} el\ dataset \\ es\ positivo \end{matrix}\right) = \frac{Cantidad\ de\ publicaciones\ positivas}{Cantidad\ total\ de\ publicaciones}$$

$$P\left(\begin{matrix} el\ dataset \\ es\ negativo \end{matrix}\right) = \frac{Cantidad\ de\ publicaciones\ negativas}{Cantidad\ total\ de\ publicaciones}$$

Figura 8: Probabilidad de que el dataset sea positivo o negativo

Posteriormente, para calcular la probabilidad de que una publicación sea positiva o negativa, dadas las palabras que contienen, usaremos las fórmulas de la Figura 9.

$$P\left(\begin{matrix} publicación\ es\ positiva \mid \\ palabras\ en\ publicación \end{matrix}\right) = \frac{P\left(\begin{matrix} palabras\ en\ publicación \mid \\ publicación\ es\ positiva \end{matrix}\right) x\, P\left(\begin{matrix} el\ dataset \\ es\ positivo \end{matrix}\right)}{P\left(\begin{matrix} palabras \\ en\ publicación \end{matrix}\right)}$$

$$P\left(\begin{matrix} publicación\ es\ negativa \mid \\ palabras\ en\ publicación \end{matrix}\right) = \frac{P\left(\begin{matrix} palabras\ en\ publicación \mid \\ publicación\ es\ negativa \end{matrix}\right) x\, P\left(\begin{matrix} el\ dataset \\ es\ negativo \end{matrix}\right)}{P\left(\begin{matrix} palabras \\ en\ publicación \end{matrix}\right)}$$

Figura 9: Probabilidad de que una publicación sea positiva o negativa

Como "P(palabras en publicación)" es "1", puesto que cada palabra siempre está presente en la publicación, al aplicar Naive Bayes, tenemos las fórmulas mostradas en la Figura 10.

$$P\left(\begin{matrix} publicación\ es\ positiva \mid \\ palabras\ en\ publicación \end{matrix}\right) = P\left(\begin{matrix} el\ dataset \\ es\ positivo \end{matrix}\right) x \prod P\left(\begin{matrix} palabra_i \\ es\ positiva \end{matrix}\right)$$

$$P\left(\begin{matrix} publicación\ es\ negativa \mid \\ palabras\ en\ publicación \end{matrix}\right) = P\left(\begin{matrix} el\ dataset \\ es\ negativo \end{matrix}\right) x \prod P\left(\begin{matrix} palabra_i \\ es\ negativa \end{matrix}\right)$$

Figura 10: Naive Bayes aplicado en las probabilidadesa

Finalmente, para elegir si una publicación es positiva o negativa, se verifica cuál de las dos probabilidades es mayor por medio de la fórmula mostrada en la Figura 11.

$$Clasificación\ de\ publicación = \begin{cases} POSITIVA, si\ P\left(\begin{matrix} publicación\ es\ positiva \mid \\ palabras\ en\ publicación \end{matrix}\right) \geq P\left(\begin{matrix} publicación\ es\ negativa \mid \\ palabras\ en\ publicación \end{matrix}\right) \\ NEGATIVA, si\ P\left(\begin{matrix} publicación\ es\ positiva \mid \\ palabras\ en\ publicación \end{matrix}\right) < P\left(\begin{matrix} publicación\ es\ negativa \mid \\ palabras\ en\ publicación \end{matrix}\right) \end{cases}$$

Figura 11: Clasificación de una publicación

Para mejorar aún más la calidad del clasificador Naive Bayes, se usó el análisis de frecuencias de TF-IDF (Ambele et al., 2014) el cual permite medir la importancia relativa de las palabras y hacer que las palabras menos significativas (stopwords) sean ignoradas en el cálculo de las probabilidades.

El algoritmo clasificador de Naive Bayes fue implementado en Java por medio de la librería Mahout.

5.4 Flujo de predicción

El flujo de predicción que definimos para nuestra arquitectura es el mostrado en la Figura 12.

Primero, los comentarios son extraídos desde la página oficial de Facebook de la operadora móvil y almacenados en la base de datos de publicaciones. Esta extracción se realiza cada hora buscando aquellos comentarios que tengan una fecha

Figura 12: Flujo seguido en la arquitectura para realizar la predicción

posterior a la última extracción realizada. Luego, el módulo predictor analiza todos aquellos comentarios obtenidos con la última extracción y para cada uno predice si es una portabilidad o no. Si la predicción indicara una portabilidad, el comentario es almacenado en la base de datos de predicciones, además, si es la primera vez que un usuario realiza una publicación con intención de portar, los datos del usuario son almacenados.

6 Pruebas realizadas

Para analizar el comportamiento de nuestra arquitectura de Big Data, se realizaron dos tipos de pruebas. La primera de ellas mide el porcentaje de aciertos que tiene el modelo para predecir la portabilidad y la segunda mide la velocidad de respuesta del clúster.

Para la primera prueba se generó el modelo de predicción con Mahout usando unos 10'000 comentarios publicados en las páginas de Facebook de Claro Perú. Cada uno de estos comentarios fue etiquetado como "negativo" en caso de que la opinión indique intención de realizar una portabilidad, y "positivo" en caso contrario. Para etiquetar a un comentario como negativo éste debía cumplir con al menos alguna de las siguientes condiciones: manifestar que algún producto o servicio de la competencia era mejor, manifestar alguna queja sobre un servicio o producto de Claro Perú, manifestar directamente que realizaría una portabilidad, o sugerir una portabilidad a otros clientes. La forma en cómo debía etiquetarse cada comentario como positivo o negativo fue sugerido por analistas de negocio de Claro Perú.

Al ser la página de Facebook de Claro Perú una página orientada al público peruano, el idioma de los comentarios escritos es el español, la lengua más extendida del Perú, sin embargo, según (Shruti et al., 2014) los algoritmos de clasificación de texto son diseñados en su mayoría para trabajar con el idioma inglés y lenguajes europeos, por lo que esta arquitectura también puede ser probada con alguno de estos otros lenguajes.

Con el modelo de predicción generado, se cargaron otros 10'000 comentarios para su validación y se construyó una matriz de confusión para medir su efectividad. En total, para generar el modelo y validarlo se trabajó con 127MB de información.

Para las pruebas de velocidad del clúster, se analizó la mejora en tiempos de respuesta que se obtenía al incrementar el número de nodos en el clúster. Se midieron los tiempos de respuesta del clúster al procesar las 10'000 publicaciones con uno, dos, tres y cuatro nodos.

Todos los nodos usados para las pruebas tenían la misma configuración de software y hardware. Cada uno contaba con un procesador Intel Core i5, 4 GB de RAM, un disco duro rígido de 500 GB y Ubuntu 14.02 con Hadoop 1.2.1.

7 Resultados obtenidos

Los resultados obtenidos al medir el porcentaje de aciertos de publicaciones positivas y negativas fueron calculados a partir de la matriz de confusión generada al realizar las pruebas, la cual es mostrada en la Figura 13.

Predicción / Real	Positivo	Negativo
Positivo	5'839	275
Negativo	722	3'164

Figura 13: Matriz de confusión

Por el lado de las predicciones realizadas para los comentarios positivos, se obtuvo un total de 5'839 aciertos y un total de 722 errores, por lo que

el porcentaje de aciertos para la predicción de comentarios positivos fue de 88.99%.

Por el lado de las predicciones realizadas para los comentarios negativos, se obtuvo un total de 3'164 aciertos y un total de 275 errores, por lo que el porcentaje de aciertos para la predicción de comentarios negativos es de 92.00%.

En conjunto, para los comentarios positivos y negativos, la arquitectura de Big Data propuesta ha obtenido un porcentaje de acierto de un 90.03%.

Los resultados obtenidos en las pruebas del clúster Hadoop, al medir la mejora en tiempos de respuesta agregándole más nodos, son mostrados en la Figura 14.

Nodos	Respuesta (segundos)
1	158
2	100
3	63
4	14

Figura 14: Tiempos de respuesta

Inicialmente, el tiempo de respuesta con un solo nodo era de 158 segundos. Al agregarle otro nodo, el tiempo de respuesta mejora pues disminuye en 58 segundos, al agregarle dos nodos se produce una mejora de 95 segundos menos, y al agregarle tres nodos, el tiempo de respuesta disminuye en 144 segundos.

En la Figura 15, puede verse que al aumentar el número de servidores, los tiempos de respuesta disminuyen de manera lineal.

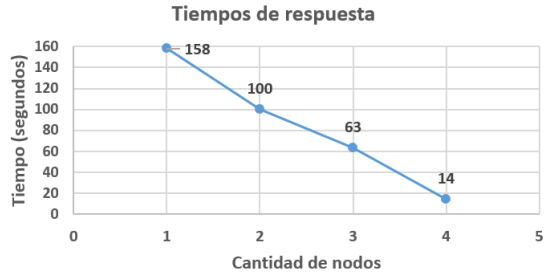


Figura 15: Nodos Vs Tiempos de respuesta

Esta escalabilidad lineal es obtenida gracias a MapReduce, por las razones explicadas en la subsección 3.3

8 Conclusiones y trabajos futuros

En el presente artículo se implementó una arquitectura basada en Big Data la cual permite predecir la portabilidad numérica de clientes en empresas de telefonía móvil. Para ello, se utilizó como fuente de datos los comentarios realizados en la página pública de Facebook de Claro Perú y con ellos se creó un modelo de predicción de análisis de sentimientos en Naive Bayes, el cual obtuvo un alto porcentaje de aciertos (90.03%) y permitió procesar 10'000 comentarios en 14 segundos. La arquitectura permite que la empresa de telefonía móvil pueda identificar en tiempo real qué clientes tienen intención de irse a la competencia, tomando como fuente los datos dejados en redes sociales.

Se comprobó que las 5 V de Big Data se encuentren relacionadas con los problemas planteados en la sección número cuatro del artículo y fueron resueltos, pues las predicciones se realizaron de manera rápida, se trabajó con datos públicos de Facebook y se realizó un proceso en el cual el cliente desconocía el análisis realizado sobre sus comentarios, el proceso fue realizado en tiempo real y sin necesidad de estandarizar los datos, y fue posible hacer seguimiento a la preferencia cambiante del cliente respecto a los servicios que las empresas de telefonía ofrecen.

Por otro lado, (Alan et al., 2015) menciona que los microblogs como Twitter también son utilizados a diario para expresar pensamientos personales en la web, y permite adquirir una valiosa cantidad de opiniones a los investigadores. Como trabajo futuro se propone incluir a Twitter en la arquitectura expuesta como una nueva fuente de datos. Para incluirla será necesario utilizar las API de REST y Streaming que Twitter provee de manera similar al OpenGraph de Facebook visto en el presente artículo.

Para la implementación de la arquitectura, se usaron las tecnologías Hadoop, Mahout y Hive, pero Mike (2013) también propone otras tecnologías como Spark, Storm e Impala que ayudan a mejorar las capacidades de una arquitectura en tiempo real. Como trabajo futuro, estas herramientas de Big Data pueden implementarse en la arquitectura propuesta y podemos llegar a comprobar si llegan a convertirse en una mejor opción tecnológica que solucione el problema.

Por último, es importante destacar que la predicción de la portabilidad numérica juega un rol importante en la industria de la telefonía móvil,

pues con el fin de reducir los diversos costos asociados a la pérdida de clientes, es imperativo que las empresas de telefonía móvil desplieguen modelos predictivos que permitan identificar qué clientes portarán a la competencia. Con estos modelos las empresas podrán formular estrategias de retención de clientes con el objetivo de aumentar sus utilidades y la rentabilidad del negocio.

Referencias

Abdullah Gani, Aisha Siddiqa, Fariza Hanum, y Shahaboddin Shamshirband. 2015. *A survey on indexing techniques for big data: taxonomy and performance evaluation*, Knowledge and Information Systems, vol. 44, n. 2, p. 1-44.

Alan Ritter, Preslav Nakov, Saif Mohammad, Sara Rosenthal, Svetlana Kiritchenko y Veselin Stoyanov. 2015. *SemEval-2015 Task 10: Sentiment Analysis in Twitter*, 9th International Workshop on Semantic Evaluation (SemEval 2015), p. 451-463.

Ambele Robert Mtafya, Dongjun Huang y Gaudence Uwamahoro. 2014. *On Objective Keywords Extraction: Tf-Idf based Forward Words Pruning Algorithm for Keywords Extraction on YouTube*, International Journal of Multimedia and Ubiquitous Engineering, vol. 9, n. 12, p. 97-106.

Carlos Acevedo Miranda, Consuelo V. García Mendoza, Ricardo Clorio Rodriguez y Roberto Zagal Flores. 2014. *Arquitectura Web para análisis de sentimientos en Facebook con enfoque semántico*, Research in Computing Science, n. 75, p. 59-69.

Clement Kirui, Hillary Kirui, Li Hong y Wilson Cheruiyot. 2013. *Predicting Customer Churn in Mobile Telephony Industry Using Probabilistic Classifiers in Data Mining*, International Journal of Computer Science Issues, vol. 10, n. 1, p. 165-172.

Debajyoti Mukhopadhyay, Chetan Agrawal, Devesh Maru, Pooja Yedale y Pranav Gadekar. 2014. *Addressing NameNode Scalability Issue in Hadoop Distributed File System using Cache Approach*, 2014 International Conference on Information Technology, Bhubaneswar, India, p. 321-326.

Dhruva Gajjar. 2014. *Implementing the Naive Bayes classifier in Mahout*, Journal of Emerging Technologies and Innovative Research, vol. 1, n. 6, p 449-454.

Francisco Barrientos y Sebastián A. Ríos. 2013. *Aplicación de Minería de Datos para Predecir Fuga de Clientes en la Industria de las Telecomunicaciones*, Revista Ingeniería de Sistemas, vol XXVII, p. 73-107.

G. Bramhaiah Achary, P. Venkateswarlu, y B.V. Srikant. 2015. *Importance of HACE and Hadoop among Big Data Applications*, International Journal of Research, vol. 2, n. 3, p. 266-272.

Kankana Kashyap, Champak Deka y Sandip Rakshit. 2014. *A Review on Big Data, Hadoop and its Impact on Business*, International Journal of Innovatite Research and Development, vol 3, n. 12, p. 78-82.

M. Vasuki, J. Arthi y K. Kayalvizhi. 2014. *Decision Making Using Sentiment Analysis from Twitter*, International Journal of Innovative Research in Computer and Communication Engineering, vol. 2, n. 12, p. 71-77.

Mario Arias, Carlos E. Cuesta, Javier D. Fernández y Miguel A. Martínez-Prieto. 2013. *SOLID: una Arquitectura para la Gestión de Big Semantic Data en Tiempo Real*, XVIII Jornadas de Ingeniería de Software y Bases de Datos, España, p. 8-21.

Mike Barlow. 2013. *Real-Time Big Data Analytics: Emerging Architecture*, O'Really Media.

Organismo Supervisor de Inversion Privada en Telecomunicaciones (OSIPTEL). 2015. *Estado de la portabilidad numérica en el primer trimestre del 2015*, Perú.

Pablo Gamallo, Juan Carlos Pichel, Marcos Garcia, José Manuel Abuín y Tomás Fernández Pena. 2014. *Análisis morfosintáctico y clasificación de entidades nombradas en un entorno Big Data*, Procesamiento del Lenguaje Natural, n. 53, p. 17-24.

Patricio Alfredo Pérez Villanueva. 2014. *Modelo de Predicción de Fuga de Cliente de Telefonía Móvil Post Pago*, Memoria para optar al Título de Ingeniero Civil Industrial. Departamento de Ingeniería Industrial, Universidad de Chile, Chile.

Rakesh Kumar, Neha Gupta, Shilpi Charu, Somya Bansal y Kusum Yadav. 2014. *Comparison of SQL with HiveQL*, International Journal for Research in Technological Studies, vol. 1, n. 9, p. 28-30.

Seyyed Mojtaba Banaei y Hossein Kardan Moghaddam. 2014. *Hadoop and Its Roles in Modern Image Processing*, Open Journal of Marine Science, vol. 4, n. 4, p 239-245.

Shruti Bajaj Mangal y Vishal Goyal. 2014. *Text News Classification System using Naïve Bayes Classifier*, International Journal of Engineering Sciences, vol. 3, p. 209-213.

Sunil B. Mane, Yashwant Sawant, Saif Kazi y Vaibhav Shinde. 2014. *Real Time Sentiment Analysis of Twitter Data Using Hadoop*, International Journal of Computer Science and Information Technologies, vol. 5, n. 3, p. 3098-3100.

Tania Lucía Cobos. 2011. *Y surge el Community Manager*, Razón y Palabra, vol. 16, n. 75.

Yiou Wang, Koji Satake, Takeshi Onishi y Hiroshi Masuichi. 2015. *Improving Churn Prediction with Voice of the Customer*, XXI Annual Meeting Lenguage Processiong Society, p. 816-819.

MapReduce and Relational Database Management Systems: competing or completing paradigms?

Dhouha Jemal
LARODEC / Tunis
dh.jemal@gmail.com

Rim Faiz
LARODEC / Tunis
rim.faiz@ihec.rnu.tn

Abstract

With the data volume that does not stop growing and the multitude of sources that led to diversity of structures, data processing needs are changing. Although, relational DBMSs remain the main data management technology for processing structured data, but faced with the massive growth in the volume of data, despite their evolution, relational databases, which have been proven for over 40 years, have reached their limits. Several organizations and researchers turned to MapReduce framework that has found great success in analyzing and processing large amounts of data on large clusters. In this paper, we will discuss MapReduce and Relational Database Management Systems as competing paradigms, and then as completing paradigms where we propose an integration approach to optimize OLAP queries process.

1 Introduction

The data is growing at an alarming speed in both volume and structure. The data explosion is not a new phenomenon; it is just accelerated in an incredible way and has an exponential number of technical and application challenges. Data generation is estimated of 2.5 trillion bytes of data every day. In addition, an IDC study predicts that overall data will grow by 50 times by 2020.

Data is the most precious asset of companies and can be mainspring of competitiveness and innovation. As presented in (Demirkan and Delen, 2013), many organizations noticed that the data they own and how they use it can make them different than others. That is why organizations need to be able to rapidly respond to market needs and changes, and it has become essential to have efficient and effective decision making processes with right data to make the decision the most adapted at a given moment. This explain the necessity to choose the right technology for processing and analyzing data.

As presented in (Ordonez, 2013), for more than forty, the relational database management systems have been the dominating technology to manage and query structured data.

However, the voluminous data which does not stop growing and the multitude of sources which led to diversity of structures challenge the needs of data processing. With these changes, the database world has been evolved and new models are presented. MapReduce is one such framework that met a big success for the applications that process large amounts of data. It is a powerful programing model characterized by its performance for heavy processing to be performed on a large volume of data that it can be a solution to have the best performance.

Several studies have been conducted to compare MapReduce and Relational DBMS. Some works present the MapReduce model as a replacement for the relational DBMSs due to its flexibility and performance, and other confirm the efficiency of the relational databases. In the other hand, many research works aim to use the two approaches together. In this environment of data explosion and diversity, the question that arises is what technology to use for data process and analysis for a particular application, how to benefit the data management systems diversity?

The aim of this work is to provide a broad comparison of the two technologies, presenting for each one its strengths and its weaknesses. Then, we propose an approach to integrate the two paradigms in order to optimize the online analytical processing (OLAP) queries process by minimizing the input/output cost in terms of the amount of

data to manipulate, reading and writing throughout the query's execution process.

The remainder of this paper is organized as follows. In section 2, we introduce MapReduce. In section 3, we describe Relational DBMS. In section 4, we present our proposed integration approach for optimizing the online analytical processing (OLAP) queries Input/Output execution cost. Finally, section 5 concludes this paper and outlines our future work.

2 MapReduce

MapReduce is a programming model completed by Google, which was introduced by Dean and Ghemawat (2004). It was designed for processing large data sets with a parallel, distributed algorithm on a cluster.

MapReduce was created in order to simplify parallel processing and distributed data on a large number of machines with an abstraction that hides the details of the hardware layer to programmers: it hides the details of parallelization, fault-tolerance, locality optimization, and load balancing. Google uses the MapReduce model to deploy large variety of problems such as: generation of data for Google's production web search service, data mining, machine learning, etc.

The MapReduce programming model has been successfully used for many different purposes. These included: parallelizing the effort; distributing the data; handling node failures.

The term MapReduce actually refers to two separate and distinct tasks: Map and Reduce. The mapper is responsible for reading the data stored on disk and process them; it takes a set of data and converts it into another set of data: reads the input block and converts each record into a Key/Value pair. The reducer is responsible for consolidating the results from the map and then write them to disk; it takes the output from a map as input and combines those data tuples into a smaller set of tuples.

At first, Google developed their own DFS: the Google File System (GFS). As described in (McClean et al., 2013), MapReduce tasks run on top of Distributed File Systems (DFS). The distributed storage infrastructure store very large volumes of data on a large number of machines, and manipulate a distributed file system as if it were a single hard drive. The DFS deals with data in blocks. In order to prevent data loss, each block will be replicated across several machines to overcome a possible problem of a single machine failure. So, this model allows the user to focus on solving and implementing his problem.

Nevertheless, the lack is that the MapReduce is independent of the storage system, it can not take into account all the input data for an available index. This explains the critics mainly from the database community. As described in (Gruska and Martin, 2010), the database community sees the MapReduce as a step backwards from modern database systems, in view of the MapReduce is a very brute force approach and it lacks the optimizing and indexing capabilities of modern database systems.

MapReduce, the powerful tool characterized by its performance for heavy processing to be performed on a large volume of data that it can be a solution to have the best performance hence makes it very popular with companies that have large data processing centers such as Amazon and Facebook, and implemented in a number of places. However, Hadoop, the Apache Software Foundation open source and Java-based implementation of the MapReduce framework, has attracted the most interest. Firstly, this is due to the open source nature of the project, additionally to the strong support from Yahoo. Hadoop has its own extensible, and portable file system: Hadoop Distributed File System (HDFS) that provides high-throughput access to application data.

Since it is introduced by Google, a strong interest towards the MapReduce model is arising. Many research works aim to apply the ideas from multi-query optimization to optimize the processing of multiple jobs on the MapReduce paradigm by avoiding redundant computation in the MapReduce framework. In this direction, MRShare (Nykiel et al., 2010) has proposed two sharing techniques for a batch of jobs. The key idea behind this work is a grouping technique to merge multiple jobs that can benefit from the sharing opportunities into a single job. However, MRShare incurs a higher sorting cost compared to the naive technique . In (Wang and Chan, 2013) two new job sharing techniques are proposed: The generalize d grouping technique (GGT) that relaxes MRShare's requirement for sharing map output. The second technique is a materialization technique (MT) that partially materializes the map output of jobs in the map and reduce phase.

On the other hand, the Pig project at Yahoo (Olston et al., 2008), the SCOPE project at Microsoft (Chaiken et al., 2008), and the open source Hive project 2 introduce SQL-style declarative languages over the standard MapReduce model, aim

to integrate declarative query constructs from the database community into MapReduce to allow greater data independence.

3 Relational DBMS

Since it was developed by Edgar Codd in 1970, as presented in (Shuxin and Indrakshi, 2005), the relational database (RDBMS) has been the dominant model for database management. RDBMS is the basis for SQL, and is a type of database management system (DBMS) that is based on the relational model which stores data in the form of related tables, and manages and queries structured data. Since the RDBMSs focuse on extending the database system's capabilities and its processing abilities, RDBMSs have become a predominant powerful choice for the storage of information in new databases because they are easier to understand and use. What makes it powerful, is that it is based on relation between data; because the possibility of viewing the database in many different ways since the RDBMS require few assumptions about how data is related or how it will be extracted from the database. So, an important feature of relational systems is that a single database can be spread across several tables which might be related by common database table columns. RDBMS also provide relational operators to manipulate the data stored into the database tables. However, as discussed in (Hammes et al., 2014), the lack of the RDBMS model resides in the complexity and the time spent to design and normalize an efficient database. This is due to the several design steps and rules, which must be properly applied such as Primary Keys, Foreign Keys, Normal Forms, Data Types, etc. Relational Databases have about forty years of production experience, so the main strength to point out is the maturity of RDBMSs. That ensure that most trails have been explored and functionality optimized. For the user side, he must have the competence of a database designer to effectively normalize and organize the database, plus a database administrator to maintain the inevitable technical issues that will arise after deployment.

A lot of work has been done to compare the MapReduce model with parallel relational databases, such as (Pavlo et al., 2009), where experiments are conducted to compare Hadoop MapReduce with two parallel DBMSs in order to evaluate both parallel DBMS and the MapReduce model in terms of performance and development complexity. The study showed that both databases did not outperformed Hadoop for user-defined function. Many applications are difficult to express in SQL, hence the remedy of the user-defined function. Thus, the efficiency of the RDBMSs is in regular database tasks, but the user-defined function presents the main ability lack of this DBMS type.

A proof of improvement of the RDBMS model comes with the introduction of the Object-Oriented Database Relational Model (ORDBMS). It aims to utilize the benefits of object oriented theory in order to satisfy the need for a more programmatic flexibility. The basic goal presented in (Sabàu, 2007) for the Object-relational database is to bridge the gap between relational databases and the object-oriented modeling techniques used in programming languages. The most notable research project in this field is Postgres (Berkeley University, Californie); Illustra and PostgreSQL are the two products tracing this research.

4 MapReduce-RDBMS: integrating paradigms

In this section, the use of relational DBMS and MapReduce as complementary paradigms is considered.

It is important to pick the right database technology for the task at hand. Depending on what problem the organization is trying to solve, it will determine the technology that should be used.

Several comparative studies have been co ducted between MapReduce and parallel DBMS such as (Mchome, 2011) and (Pavlo et al., 2009). MapReduce has been presented as a replacement for the Parallel DBMS. While each system has its strengths and its weaknesses. However, an integration of the two systems is needed, and as proposed in (Stonebraker et al., 2010), MapReduce can be seen as a complement to a RDBMS for analytical applications, because different problems require complex analysis capabilities provided by both technologies.

In this context, we propose a model that integrates the MapReduce model and a relational DBMS PostgreSQL presented in (Worsley and Drake, 2002), in a goal of queries optimization. We suggest an OLAP queries process model in a goal of minimizing Input/Output costs in terms of the amount of data to manipulate, reading and writing throughout the execution process.

The basic idea behind our approach is based on the cost model to approve execution and selectivity of solutions based on the estimated cost of execution. To support the decision making process for analyzing data and extracting useful

knowledge while minimizing costs, we propose to compare the estimates of the costs of running a query on Hadoop MapReduce compared to PostgreSQL to choose the least costly technology.

As the detailed analysis of the queries execution costs showed a gap mattering between both paradigms, hence the idea of the thorough analysis of the execution process of each query and the implied cost. So, to better control the cost difference between costs of Hadoop MapReduce versus PostgreSQL on each step of the query's execution process, we propose to dissect each query for a set of operations that demonstrates the process of executing the query and in order to control the different stages of the execution process of each query (decomposing the query to estimate the costs on each system for each individual operation). In this way, we can check the impact of the execution of each operation of a query on the overall cost and we can control the total cost of the query by controlling the partial cost of each operation in the information retrieval process. For this purpose, we suggest to provide a detailed execution plan for OLAP queries. This execution plan allows zooming on the sequence of steps of the process of executing a query. It details the various operations of the process highlighting the order of succession and dependence. In addition, it determines for each operation the amount of data involved and the dependence implemented in the succession of phases. These parameters will be needed to calculate the cost involved in each operation.

Having identified all operations performed during the query execution process the next step is then to calculate the cost implied in each operation independently, in both paradigms PostreSQL and MapReduce with the aim of controlling the estimated costs difference according to the operations as well as the total cost of query execution. At this stage we consider each operation independently to calculate an estimate of its cost execution on PostreSQL on one hand then on MapReduce on the other hand. For this goal, we propose to relay on a cost model for each system PostgreSQL and Hadoop MapReduce in order to estimate the I/O cost of each operation execution on both system independently.

We aim to estimate how expensive it is to preprocess each operation of each query on both systems. Therefore, controlling the cost implied by each operation as well as its influence on the total cost of the query, allows the control of the cost of each query to support the decision making process and the selectivity of the proposed solutions based on the criterion of cost minimization.
Based on a sample workload of OLAP queries, and having identifying the cost of each operation for each query executed on both systems independently, the results analysis can be useful to deduct a generalized smart model that integrates the two paradigms to process the OLAP queries in a cost minimization way.

5 Conclusion

Given the exploding data problem, the world of databases has evolved which aimed to escape the limitations of data processing and analysis. There has been a significant amount of work during the last two decades related to the needs of new supporting technologies for data processing and knowledge management, challenged by the rise of data generation and data structure diversity.

In this paper, we have investigated the MapReduce model in one hand, then the Relational DBMS technology in the other hand, in order to present strengths and weaknesses of each paradigm.

Although MapReduce was designed to cope with large amounts of unstructured data, there will be advantages in exploiting it in structured data processing. In this fashion, we have proposed in this paper a new OLAP queries process model integrating an RDBMS with the MapReduce framework in a goal of minimizing Input/Output costs in terms of the amount of data to manipulate, reading and writing throughout the execution process.

Combining MapReduce and RDBMS technologies has the potential to create very powerful systems. For this reason, we plan to investigate other types of integration for different applications.

Reference

Chaiken R., Jenkins B., Larson P.A., Ramsey B., Shakib D., Weaver S. and Zhou J. 2008. Scope: Easy and efficient parallel processing of massive data sets. *Proceedings of the VLDB Endowment*, volume 1 (2). 1265-1276.

Demirkan H. and Delen D. 2013. Leveraging the capabilities of service-oriented decision support systems: Putting analytics and big data in cloud. *Decision Support Systems*, Volume 55 (1). 412-421.

Dean and Ghemawat. 2004. MapReduce: Simplified Data Processing on Large Clusters. *Proceedings of the 6th Conference on Symposium on Opearting Systems Design & Implementation*, volume 6.

Gruska N. and Martin P. 2010. Integrating MapReduce and RDBMSs. *Proceedings of the 2010 Conference*

of the Center for Advanced Studies on Collaborative Research, IBM Corp. 212-223.

Hammes D., Medero, H. and Mitchell H. 2014. Comparison of NoSQL and SQL Databases in the Cloud. *Southern Association for Information Systems (SAIS) Proceedings*. Paper 12.

McClean A., Conceicao RC., and O'Halloran M. 2013. A Comparison of MapReduce and Parallel Database Management Systems. *ICONS 2013, The Eighth International Conference on Systems*: 64-68.

Mchome M.L. 2011. Comparison study between MapReduce (MR) and parallel data management systems (DBMs) in large scale data anlysis. *Honors Projects Macalester College*.

Nykiel T., Potamias M., Mishra C., Kollios G. and Koudas N. 2010. Mrshare: sharing across multiple queries in mapreduce. *Proceedings of the VLDB Endowment* ,volume 3 (1-2). 494-505.

Ordonez C. 2013. Can we analyze big data inside a DBMS?. *Proceedings of the sixteenth international workshop on Data warehousing and OLAP*, ACM. 85-92.

Olston C., Reed B., Srivastava U., Kumar R. and Tomkins A. 2008. Pig latin: a not-so-foreign language for data processing. *Proceedings of the 2008 ACM SIGMOD international conference on Management of data , ACM*. 1099-1110.

Pavlo A., Rasin A., Madden S., Stonebraker M., DeWitt D., Paulson E., Shrinivas L. and Abadi D.J. 2009. A comparison of approaches to large scale data analysis. *Proceedings of the 2009 ACM SIGMOD International Conference on Management of data, ACM*. 165-178.

Shuxin Y. and Indrakshi R. 2005. Relational database operations modeling with UML. *Proceedings of the 19th International Conference on Advanced Information Networking and Applications*. 927-932.

Sabàu G. 2007. Comparison of RDBMS, OODBMS and ORDBMS. *Informatica Economic*.

Stonebraker M., Abadi D., DeWitt D.J., Madden S., Paulson E., Pavlo A. and Rasin A. 2010. Mapreduce and parallel dbmss : friends or foes ?. *Communications of the ACM*, volume 53 (1). 64-71.

Wang G. and Chan CY. 2013. Multi-Query Optimization in MapReduce Framework. *Proceedings of the VLDB Endowment, 40th International Conference on Very Large Data Bases*, volume 7 (3).

Worsley J.C. and Drake J.D. 2002. Practical PostgreSQL. *O'Reilly and Associates Inc*

Performance of Alternating Least Squares in a distributed approach using GraphLab and MapReduce

Elizabeth Veronica Vera Cervantes, Laura Vanessa Cruz Quispe, José Eduardo Ochoa Luna
National University of San Agustin
Arequipa, Peru
elizavvc@gmail.com, lcruzq@unsa.edu.pe, eduardo.ol@gmail.com

Abstract

Automated recommendation systems have been increasingly adopted by companies that aim to draw people attention about products and services on Internet. In this sense, development of distributed model abstractions such as MapReduce and GraphLab has brought new possibilities for recommendation research tasks due to allow us to perform Big Data analysis. Thus, this paper investigates the suitability of these two approaches for massive recommendation. In order to do so, the Alternating Least Squares (ALS), which is a Collaborative Filtering algorithm, has been tested using recommendation benchmark datasets. Results on RMSE show a preliminary comparative performance analysis.

1 Introduction

Data on the Internet is increasing, e-commerce sites, blogs and social networks spread the word about new products and services everyday. This social media information overwhelms any user, who has a given profile and therefore could not be interested in most of these offers (Koren et al., 2009).

In this sense, recommendation systems have gained momentum, because they "filter" products and services for users according to behavior patterns. Traditional approaches for automated recommendation range from Content-Based, Collaborative Filtering and Deep Learning systems (Adomavicius, 2005; Shi et al., 2014). However, to handle the current amount of available data we need to resort to frameworks for large-scale data processing.

Recently, the machine learning community has been increasingly interested in the task of managing Big Data with parallelism (Zhou et al., 2008; De Pessemier et al., 2011; Xianfeng Yang, 2014). However, parallel algorithms are extremely challenging and traditional approaches, despite of being powerful like MPI, rely on low levels of abstraction. On the other hand, distributed models such as GraphLab(Low et al., 2012) and MapReduce(Xiao and Xiao, 2014; Dean and Ghemawat, 2008) foster high levels of abstraction and, therefore, they are more intuitive. The aim of this paper is to investigate whether these distributed models are suitable for recommendation tasks. In order to do so we evaluate the Alternating Least Squares (ALS) algorithm, a parallel collaborative filtering approach(Koren et al., 2009; Schelter et al., 2013), in both GraphLab and MapReduce frameworks. We evaluate the performance on the MovieLens and Netflix datasets. According to preliminary results, GraphLab outperforms MapReduce in RMSE, when Lambda, iterations number and latent factor parameters are considered. Conversely, MapReduce gets a better execution time than GraphLab using the same parameters in MoveLens dataset. The paper is organized as follows. In Section 2, related work is described. Background is given in Section 3. Our proposal is showed in Section 4. Preliminary results are depicted in section 5. Finally Section 5, concludes the paper.

2 Related Work

Several distributed platforms have been used for studying performance of machine learning algorithms, for instance, a Matrix Factorization based on collaborative filtering over MapReduce model was proposed in (Xianfeng Yang, 2014; De Pessemier et al., 2011). In Low et al. (2012), some advantages and disadvantages of using GraphLab and MapReduce were described. For instance, MapReduce fails when there are computational dependencies on data, but it can be used to extract features from a massive collection.

In addition, MapReduce is targeted for large data centers, it is optimized for node-failure and disk-centric parallelism. Conversely, In GraphLab it is assumed that processors do not fail, and all data is stored in shared memory.

In Low et al. (2012), the Alternating Least Squares (ALS) algorithm was implemented over several platforms: GraphLab, Hadoop/MapReduce and MPI. Comparison results show that applications created using GraphLab outperformed equivalent Hadoop/MapReduce implementations by 20-60 times(Xianfeng Yang, 2014) .

Our work is most related to Low et al. (2012), but we focus on the evaluation of different configurations of ALS algorithm over GraphLab and MapReduce. Thus, we aim at obtaining optimal parameters that allow us to improve algorithm performance. Moreover, comparisons were based on RMSE and time execution values. The parameters considered are:

- Lambda, which is the regularization parameter in ALS
- The number of latent factors
- The number of iterations

3 Background

3.1 Recommendation Systems

A recommendation system aims at showing items of interest to a user, considering the context of where the items are being shown and to whom they are being shown (Alag, 2008).

Figure 1, depicts inputs and outputs of a common recommendation system.

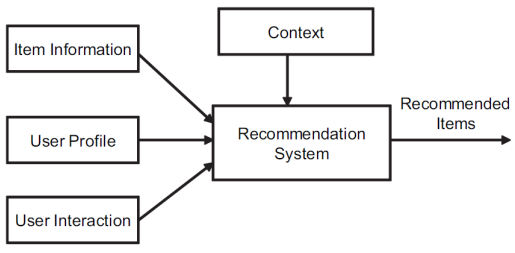


Figure 1: Inputs and Outputs of a Recommendation Engine

(Alag, 2008)

In Adomavicius (2005) three approaches for building a recommendation system are presented:

- Content-based Recommendation: Items similar to the ones he/she has preferred in the past, are recommended to the user.
- Collaborative Recommendation: Items that people with similar tastes and preferences liked in the past, are recommended to the user.
  - Collaborative Deep Learning: It is a recent kind of collaborative filtering using deep learning models Wang et al. (2014).
- Hybrid Approach: Recommendations are made using a combination of Content-based and Collaborative Recommendation methods.

3.2 Alternating Least Squares (ALS)

Alternating Least Squares (Low et al., 2012; Zhou et al., 2008; Koren et al., 2009) is an algorithm within the collaborative filtering paradigm. Input of ALS (in Figure 2) is a sparse user by items matrix R containing the rating of each user. The algorithm iteratively computes a low-rank matrix factorization $R = U \times V$ where U and V are d dimensional matrices. The loss function is defined as the squared error(Zhou et al., 2008), where the learning objective is to minimize the sum of the squared errors 1 between values predicted and real values of rantings.

$$(\hat{U}, \hat{V}) = \underset{U,V}{\operatorname{argmin}} \sum i, j \in R(r_{ij} - v_i^T u_j)^2 \quad (1)$$

Complexity and cost depend on the magnitude of the hidden variables d.

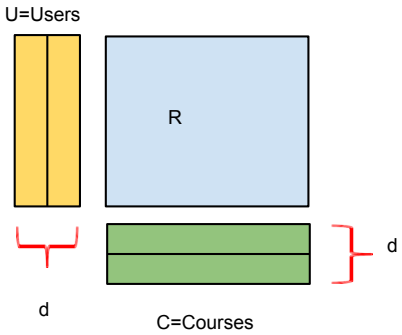


Figure 2: Matrix factorization of ALS

The ALS algorithm is computationally expensive, every iteration runs on $O(d^{-1}[Nr + (m +$

$n)r^2] + r^3)$, where m is length of items, and n is length of users (Schelter et al., 2013; Gemulla et al., 2011).

3.3 Alternating Least Squares on GraphLab

According to (Low et al., 2012; Gonzalez et al., 2012) ALS in GraphLab is implemented by using a bipartite two colorable graph and a chromatic synchronous engine with an edge consistency model for serializability.

Each vertex of the graph has a latent factor attached, that denotes a user or an item. Thus, they are linked to a column or a row in the matrix of ratings R. Each edge of the graph contains entry data (rating values), and the most recent error estimated by the algorithm. The goal of ALS algorithm is to discover values of latent parameters, such that non-zero entries in R can be predicted by the dot product of the row and column latent factor. ALS algorithm for GraphLab is implemented in the Gather-Apply-Scatter abstraction. ALS update considers adjacent vertices as X values and edges as observed y values, and then updates the current vertex value as a weight w:

$$y = X * w + noise \tag{2}$$

that is accomplished using the following equation:

$$w = inv(X' * X) * (X' * y) \tag{3}$$

In the Gather-Apply-Scatter model, the update is done as follows:

- Gather: it returns the tuple $(X' * X, X' * y)$
- Apply: it solves $inv(X' * X) * (X' * y)$
- Scatter: it schedules the update of adjacent vertices if this vertex has changed and the edge is not well predicted.

3.4 Alternating Least Squares on MapReduce

In Xianfeng Yang (2014; Zhou et al. (2008) MapReduce implementation is comprised by four tasks as shown in Figure 3. Each item in dataset is denoted as a triple (u, j, r). u denotes user, j is the label of item and r denotes corresponding rating. In the U-Update step, item matrix V is used as input and is sent to cluster nodes. Then, training rating R is used to compute user matrix U, including inputs as lambda parameter λ to regularization, number of latent factors. V-Update does the same as U-Update step, but its input is not an item matrix. On the contrary, it is a user matrix computed in U-Update step. Once U and V are learned, we can compute RMSE values using test dataset and estimated rating $\hat{r}$. So the Parallel ALS algorithm with Weighted--Regularization is as follows (Zhou et al., 2008): The objective function in

Algorithm 1 Alternating Least Square(ALS) with algorithm

1: Initialize V with random values between 0 and 1
2: Hold V constants, and solve U by minimizing the objective function.
3: Hold U constants, and solve M by minimizing the objective function.
4: **repeat** from step 2 and 3 until objective function converge.

1 is obtained from equation 4, which is just linear regression with lambda regularization(λ), to avoid overfits it penalize large parameters.

4 Proposal

In this paper we evaluate several parameter configurations (lambda, number of latent factor, number of iterations) for ALS algorithm over GraphLab and MapReduce. Our aim is to obtain the best performance, over clusters of two and four machines, for the Movielens Dataset, and NetFlix Dataset (further details will be given in the next section). We evaluate performance according to RMSE and execution time values.

In order to implement ALS algorithm under the MapReduce Paradigm, the Mahout [1] API has been used. ALS algorithm for MapReduce (Zhou et al., 2008) is shown in 3. User and movie factors have been computed using equation 4. where n_{ui} and n_{vj} are the numbers of ratings of user i and item j respectively. When objective function showed in equation 4 does not change after further iterations, we attain the final step. Output is the predicted rating for each user/item pair.

[1] http://mahout.apache.org/

$$f(U,V) = \sum_{i,j \in I} (r_{ij} - u_i^T v_j)^2 + \lambda (\sum_i n_{ui} \| u_i \|^2 + \sum_j n_{vj} \| v_j \|^2) \quad (4)$$

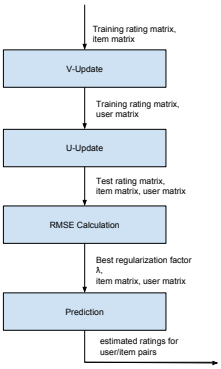


Figure 3: MapReduce ALS algorithms proposed (Zhou et al., 2008; Xianfeng Yang, 2014)

In order to evaluate ALS algorithm under GraphLab, the GraphLab API (Low et al., 2010) has been used. ALS algorithm for GraphLab (Low et al., 2012) is shown in Figure 4. User and movie factors have been computed using equation 5.

$$f[i] = \underset{w \in R^d}{\operatorname{argmin}} \sum_{j \in Neighbors(i)} (r_{ij} - w^T f[j]) \quad (5)$$

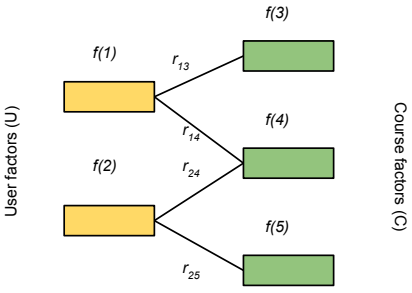


Figure 4: Matrix factorization of ALS using GraphLab

4.1 Movielens Dataset

MovieLens is a Web collaborative site that manages a recommender system for movies. This recommender system is based on a collaborative filtering algorithm developed by the GroupLens research group. The dataset is comprised by 6040 users, 3952 items and 100209 ratings for training. The data structure is: *user*, *item*, *rating*.

4.2 Netflix Dataset

We are using the Small Netflix Dataset. It is also a data-set for movie recommendation, it has 95526 users, 3561 items and 3298163 ratings. The structure of the data-set is: *user*, *item*, *rating*.

4.3 GraphLab Configuration

Setup of the GraphLab cluster is as follows. Two machines, one working as the master and the other as the worker node. The master machine operating system is Ubuntu 14.04, and its processor is Intel Core i3 CPU M 330@2.13GHzx4. The worker machine operating system is Ubuntu 13.10 of 64-bit, and its processor is Intel Core i3-2350M @2.30GHzx4. The cluster was configured using MPI(Message Passing Interface).

4.4 MapReduce Configuration

The setup is as follows. Four machines, three worker nodes and one master. The master machine operating system is Ubuntu 13.10 of 64-bit, and its processor is Intel Core i3-2350M @2.30GHzx4. Table.1 shows the configuration of the worker machines.

The cluster was configured using Hadoop, and

Machine	Operating System	Processor
1	Ubuntu 14.04	Intel Core i3 CPU M 330@2.13GHzx4
2	Ubuntu 13.10	Intel Core i3-2350M @2.30GHzx4
3	Ubuntu 14.04	Intel Core i7-4700MQ @2.40GHzx8

Table 1: Worker Machines Configuration

the HDFS(Hadoop Distributed File System). The ALS(Alternating Least Squares) algorithm implementation was taken from Mahout Library.

5 Experimental Results

This section shows experimental results conducted on MovieLens data set aforementioned. Experimental setting parameters are described in

Parameters	Value
Lambda	0.01 - 0.09
# Latent factors	10-50
# Iterations	2-30

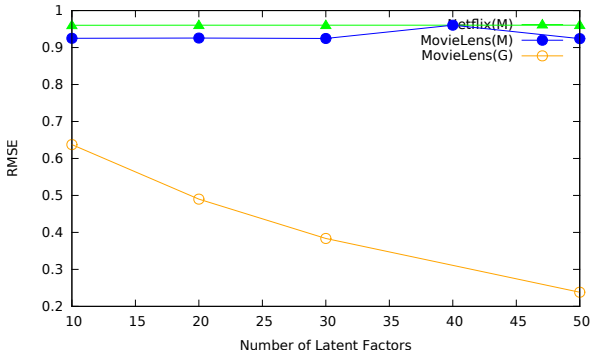
Table 2: Parameters used for ALS algorithms

Table 2. Latent factors have been increased for each test in 10 step size, Lambda has been increased in 0.01 step size, and Number of iterations in 1 step size. Results are showed in Figures 5,6,7.

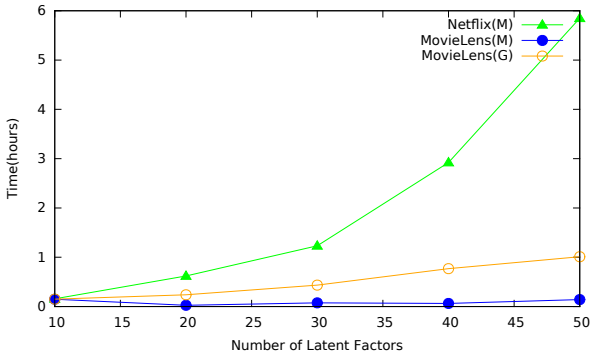
In Figure 5a, RMSE values for MapReduce do not change even if we increase the number of latent factors thus, RMSE values on MapReduce are independent on the number of latent factors. RMSE for MapReduce converges around 0.95. Conversely, RMSE values for GraphLab decreases while the number of latent factors increases. When the number of latent factors was 50, RMSE value reaches around 0.25. However, GraphLab spends more time than MapReduce, Figure 5b depicts MapReduce times almost as an horizontal line for MovieLens dataset, the line of execution time for Netflix dataset is much steeper. Between Graphlab and MapReduce lines representing Movilens dataset execution, Graphlab line is more pronounced.

Figure 6a depicts GraphLab and MapReduce performance according the Lambda parameter. While Lambda increases, RMSE decreases accordingly, i.e., if a greater value of Lambda is used then algorithm accuracy tends to be better. We also notice that Graphlab has lower values of RMSE compared to MapReduce. GraphLab RMSE values are around 0.5, and MapReduce RMSE values are around 1. Figure 6b illustrates a better execution time of MapReduce compare to GraphLab over Movielens dataset. However, now the execution time for GraphLab decreases, while the value of Lambda increases. Figure 6b also shows that It takes longer to process the data from netflix than Movielens.

In Figure 7a we notice that the value of RMSE is almost invariant to the increase of iterations for MapReduce execution, given that the number of iterations are small,nevertheless we notice clearly that RMSE value for GraphLab decreases as the number of iterations increases. RMSE value for Graphlab converges around 0.55. Figure 7b shows that MapReduce execution time over Movielens dataset is good, however it increases a lot for Netflix dataset. Graphlab execution time increases as the number of iterations grows.

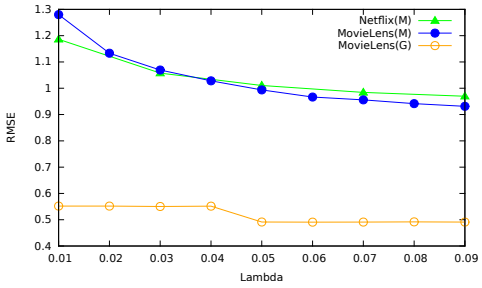


(a) Number of latent factors Vs. RMSE

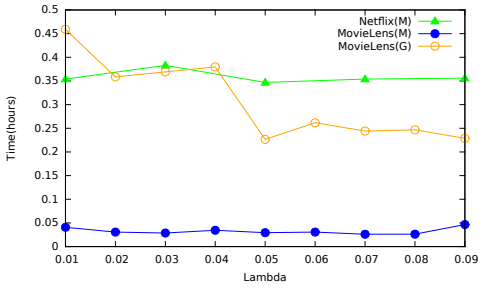


(b) Number of latent factors Vs. Time

Figure 5: Performance of MapReduce and GraphLab when number of features in ALS algorithms is increased.

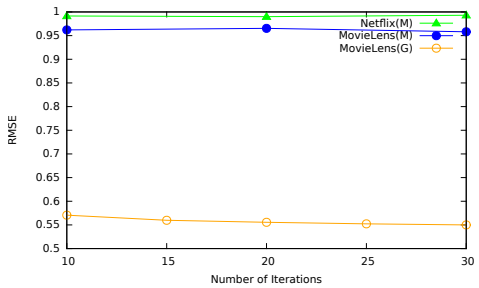


(a) Lambda Vs. RMSE

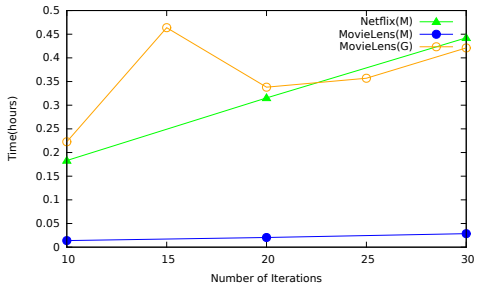


(b) Lambda Vs. Time

Figure 6: Performance of MapReduce and GraphLab when Lambda values in ALS algorithms are increased.



(a) Number of Iterations Vs. RMSE



(b) Number of Iterations Vs. RMSE

Figure 7: Performance of MapReduce and GraphLab when the number of iterations in ALS algorithms is increased.

6 Conclusion

We evaluated the Alternating Least Squares (ALS) algorithm, a parallel collaborative filtering in both GraphLab and MapReduce frameworks. Experiments were run over the MovieLens and Netflix datasets. The RMSE between MapReduce execution in NetFlix dataset and Movielens dataset in all the experiments was similar, but the execution time was longer in Netflix dataset.
Looking at the executions over Moviliens dataset, we can say, that even though GraphLab only ran in two machines and MapReduce in 4 machines, the first one outperformed the second one in RMSE. Considering lambda value variation, Figure.6a,the number of iterations Figure.7a, and the number of latent factors Figure.5a, GraphLab performed better (RMSE) than MapReduce. In all previous three cases MapReduce was faster than GraphLab, obviously by the difference between the number of machines in their configuration.

Thus, when scalability and distribution are evaluated, MapReduce performs better, because ALS does not require data dependency for computing. Moreover, it took less execution time when more latent factors were added. In this work we only used two nodes, however GraphLab demonstrated best results with few nodes.

In conclusion, GraphLab performed better when RMSE was considered but, there are open issues with shared-memory. GraphLab is also better for computing recommendations in real time. However, for more sophisticated computations MapReduce performs better so far as to an offline environment and all data is used.

References

Tuzhilin A. Adomavicius, G. 2005. Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. *Knowledge and Data Engineering, IEEE Transactions on*, 17:734 – 749.

Satnam Alag. 2008. *Collective Intelligence in Action*.

Toon De Pessemier, Kris Vanhecke, Simon Dooms, and Luc Martens. 2011. Content-based recommendation algorithms on the hadoop mapreduce framework. In *7th international conference on Web Information Systems and Technologies, Proceedings*, pages 237–240. Ghent University, Department of Information technology.

Jeffrey Dean and Sanjay Ghemawat. 2008. Mapreduce: Simplified data processing on large clusters. *Commun. ACM*, 51(1):107–113, January.

Rainer Gemulla, Erik Nijkamp, Peter J Haas, and Yannis Sismanis. 2011. Large-scale matrix factorization with distributed stochastic gradient descent. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 69–77. ACM.

Joseph E. Gonzalez, Yucheng Low, Haijie Gu, Danny Bickson, and Carlos Guestrin. 2012. Powergraph: Distributed graph-parallel computation on natural graphs. In *Proceedings of the 10th USENIX Conference on Operating Systems Design and Implementation*, OSDI'12, pages 17–30, Berkeley, CA, USA. USENIX Association.

Y Koren, R Bell, and C Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37.

Yucheng Low, Joseph Gonzalez, Aapo Kyrola, Danny Bickson, Carlos Guestrin, and Joseph M. Hellerstein. 2010. Graphlab: A new parallel framework for machine learning. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, July.

Yucheng Low, Danny Bickson, Joseph Gonzalez, Carlos Guestrin, Aapo Kyrola, and Joseph M Hellerstein. 2012. Distributed graphlab: a framework for machine learning and data mining in the cloud. *Proceedings of the VLDB Endowment*, 5(8):716–727.

Sebastian Schelter, Christoph Boden, Martin Schenck, Alexander Alexandrov, and Volker Markl. 2013.

Distributed matrix factorization with mapreduce using a series of broadcast-joins. In *Proceedings of the 7th ACM Conference on Recommender Systems*, RecSys '13, pages 281–284, New York, NY, USA. ACM.

Yue Shi, Martha Larson, and Alan Hanjalic. 2014. Collaborative filtering beyond the user-item matrix: A survey of the state of the art and future challenges. *ACM Comput. Surv.*, 47(1):3:1–3:45, May.

Hao Wang, Naiyan Wang, and Dit-Yan Yeung. 2014. Collaborative deep learning for recommender systems. *CoRR*, abs/1409.2944.

Pengfei Liu Xianfeng Yang. 2014. Collaborative filtering recommendation using matrix factorization: A mapreduce implementation. *International Journal of Grid and Distributed Computing*.

Zhifeng Xiao and Yang Xiao. 2014. Achieving accountable mapreduce in cloud computing. *Future Gener. Comput. Syst.*, 30:1–13, January.

Yunhong Zhou, Dennis Wilkinson, Robert Schreiber, and Rong Pan. 2008. Large-scale parallel collaborative filtering for the netflix prize. In *Proceedings of the 4th International Conference on Algorithmic Aspects in Information and Management*, AAIM '08, pages 337–348, Berlin, Heidelberg. Springer-Verlag.

Data Modeling for NoSQL Document-Oriented Databases

Harley Vera, Wagner Boaventura, Maristela Holanda, Valeria Guimarães, Fernanda Hondo
Department of Computer Science
University of Brasília
Brasília, Brasil.
{harleyve,wagnerbf}@gmail.com, mholanda@cic.unb.br,
{valeriaguimaraes, fernandahondo}@hotmail.com

Abstract

In database technologies, some of the new issues increasingly debated are non-conventional applications, including NoSQL (Not only SQL) databases, which were initially created in response to the needs for better scalability, lower latency and higher flexibility in an era of bigdata and cloud computing. These non-functional aspects are the main reason for using NoSQL database. However, currently there are no systematic studies on data modeling for NoSQL databases, especially the document-oriented ones. Therefore, this article proposes a NoSQL data modeling standard in the form of ER diagrams, introducing modeling techniques that can be used on document-oriented databases. On the other hand the purpose of this article is not structure the data using the model proposed, but it does helping with the visualization of data. In addition, to validate the proposed model, a study case was implemented using genomic data.

1 Introduction

Huge amounts of data are produced daily. They are generated by smart phones, social networks, banks transactions, machines measured by sensors are part of Internet of Things provide information that is growing exponentially. The management of this data is currently performed in most cases by relational databases that provide centralized control of data, redundancy control and elimination of inconsistencies (Elmasri and Navathe, 2010); but, some of these factors restrict the use of alternative database models. Consequently, certain limiting factors have led to alternative models of databases in these scenarios. Primarily, motivated by the issue of system scalability, a new generation of databases, known as NoSQL, is gaining strength and space in information systems. The NoSQL databases emerged in the mid-90s, from a database solution that did not provide an SQL interface. Later, the term came to represent solution that promote an alternative to the Relational Model, becoming an abbreviation for Not Only SQL.

The purpose, therefore, of NoSQL solutions is not to replace the Relational Model as a whole, but only in cases in which there is a need for scalability and bigdata. In the recent years, a variety of NoSQL databases has been developed mainly by practitioners looking to fit their specific requirements regarding scalability performance, maintenance and feature-set. Subsequently, there have been various approaches to classify NoSQL databases, each with different categories and subcategories, such as key-value stores, column-oriented and graph databases, oriented-document. MongoDB (MongoDB, 2015), Neo4j (Partner et al., 2013), Cassandra (D. Borthakur et al., 2011) and HBase (F. Chang et al., 2008) are examples of NoSQL databases. This article only applies to NoSQL document-oriented databases, because of the heterogeneous characteristics of each NoSQL database classification.

Nonetheless, data modeling still has an important role to play in NoSQL environments. The data modeling process (Elmasri and Navathe, 2010) involves the creation of a diagram that represents the meaning of the data and the relationship between the data elements. Thus, understanding is a fundamental aspect of data modeling (R. F. Lans, 2008), and a pattern for this kind of representation has few contributions for NoSQL databases.

Addressing this issue, this article proposes a standard for NoSQL data modeling. This proposal uses NoSQL document-oriented databases, aiming to introduce modeling techniques that can be

used on databases with document features.

The remainder of the paper is organized as follows: Section II presents related works. Section III explores the concepts of modeling for NoSQL databases based on documents, introducing the different types of relationships and associations. Section IV shows the proposal model in the context of NoSQL databases based on documents. Section V presents the study case to validate the proposal model. Finally in Section VI, presents the conclusion of the research and future works.

2 Related Works

Katsov (H. Scalable, 2015) presents a study of techniques and patterns for data modeling using different categories of NoSQL databases. However, the approach is generic and does not define a specific modeling engine to each database.

Arora and Aggarwal (R. Arora and R. Aggarwal, 2013) propose a data modeling, but restricted to MongoDB document database, describing a UML Diagram Class and JSON format to represent the documents.

Similarly, Banker (K. Banker, 2011) provides some ideas of data modeling, but limited to MongoDB database and always referring to JSON (D. Crockford, 2006) format as a modeling solution.

Kaur and Rani (K. Kaur, K.Rani, 2013) present a work for modeling and querying data in NoSQL databases, especifically present a case study for document-oriented and graph based data model. In the case of document-oriented propose a data modeling restricted to MongoDB document database, describing the data model by UML diagram class to represent documents.

3 Data Modeling For Document-Oriented Database

An important step in database implementation is the data modeling, because it facilitates the understanding of the project through key features that can prevent programming and operation errors. For relational databases, the data modeling uses the Entity-Relationship Model (Elmasri and Navathe, 2010). For NoSQL, it depends on the database category. The focus of this article is NoSQL document-oriented databases, where the data format of these documents can be JSON, BSON, or XML (S. J. Pramod, 2012).

Basically, the documents are stored in collections. A parallel is made with relational databases, the equivalent for a collection is the record (tuple) and for a document it is the relation (table). Documents can store completely different sets of attributes, and can be mapped directly to a file format that can be easily manipulated by a programming language. However, it is difficult to abstract the modeling of documents for the entity relationship model (R. F. Lans, 2008).

3.1 Modeling Paradigm for document-oriented Database

The relational model designed for SQL has some important features such as integrity, consistency, type validation, transactional guarantees, schemes and referential integrity. However, some applications do not need all of these features. The elimination of these resources has an important influence on the performance and scalability of data storage, bringing new meaning to data modeling.

Document-oriented databases have some significant improvements, e.g., index management by the database itself, flexible layouts and advanced indexed search engines (H. Scalable, 2015). By associating these improvements (some being denormalization and aggregation) to the basic principles of data modeling in NoSQL, it is possible to identify some generic modeling standards associated to document-oriented databases. Analyzing the documentation of the main document-oriented databases, MongoDB (MongoDB, 2015) and CouchDB (CouchDB, 2015), similar representations of data mapping relationships can be found: **References** and **Embedded Documents**, a structure which allows associating a document to another, retaining the advantage of specific performance needs and data recovery standards.

3.2 References Relationship

This type of relationship stores the data by including links or references, from one document to another. Applications can solve these references to access the related data in the structure of the document itself (MongoDB, 2015). Figure 1 shows two documents one of them for **Fastq** files and the other to **Activities**.

3.3 Embedded Documents

This type of relationship stores in a single document structure, where the embedded documents are disposed in a field or an array. These denormalized data models allow data manipulation in a single database transaction (MongoDB, 2015).

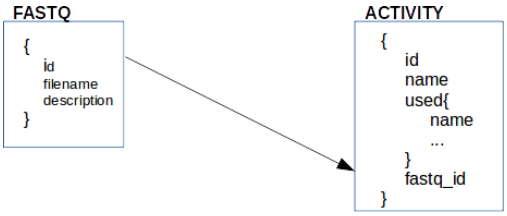


Figure 1: Example of documents referenced

Figure 2 shows a document of a genome **Project** with a **Activity** embedded document.

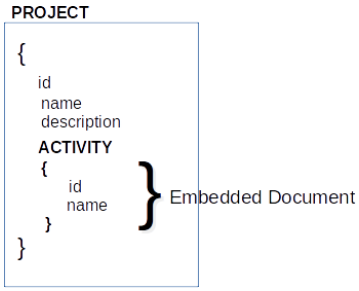


Figure 2: Example of Embedded documents

4 Proposal For Document-Oriented Databases Viewing

Unlike traditional relational databases that have a simple form in the disposition in rows and columns, a document-oriented database stores information in text format, which consists of collections of records organized in key-value concept, ie, for each value represented a name (or label) is assigned, which describes its meaning. This storage model is known as JSON object, and the objects are composed of multiple name/value pairs for arrays, and other objects.

In this scenario, the number of objects in a database increases the abstraction complexity of the logical relationship between the stored information, especially when objects have references to other objects. Currently, there is a lack of solutions to conceptually represent those associated with a NoSQL document-oriented database. As described in (R. Arora and R. Aggarwal, 2013), there is no standard to represent this kind of object modeling, several diferent manners of modeling may arise, depending on each data administrator's understanding, which makes learning difficult for those who need to read the database model.

Therefore, this section proposes a standard for document-oriented database viewing. Our proposal has some properties, considering the conceptual representation modeling type, such as:

- Ensuring a single way of modeling for the several NoSQL document-oriented databases.
- Simplifying and facilitating the understanding of a document-oriented database through its conceptual model, leveraging the abstraction and making the correct decisions about the data storage.
- Providing an accurate, unambiguous and concise pattern, so that database administrators have substantial gains in abstraction, understanding.
- Presenting different types of relationships between collections defined as References and Embedded documents.
- Assisting the recognition and arrangement of the objects, as well as its features and relationships with other objects.

The following subsections present the concepts and graphing to build a conceptual model for NoSQL document-oriented databases.

4.1 Assumptions

Before starting the discussion about the approach of each type of the conceptual modeling representation, it is important to highlight some basic concepts about objects and relationships in a document-oriented database:

- A document (or object) describes a set of attributes that have their properties organized in a key-value structure.
- Information contained in an document is described by the identifier (key) and the value associated with the key.
- Different types of relationships between documents are defined as **References** and **Embedded Documents**
- Because NoSQL is a non-relational data database, the concepts of normalization, do not apply.
- Some concepts of relationships between objects are similar to ER modeling, such as cardinality (one-to-one, one-to-many, many-to-many).

4.2 Basic Visual Elements

The proposed solution for a conceptual modeling to the NoSQL document-oriented databases has two basic concepts: **Document** and **Collections**.

As noted previously, a document is usually represented by the structure of a JSON object, and as many fields as needed may be added to the document. For this solution, a document and a collection of documents is represented by Figure 3.

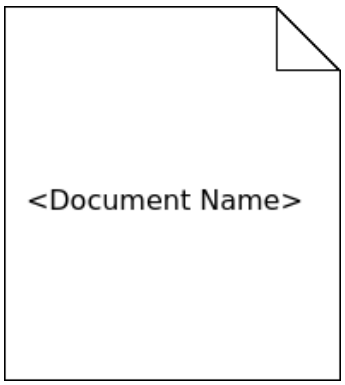


Figure 3: Graphical representation of a Document

The following section presents the definitions of relationship types and degrees for the objects features.

4.3 Embedded Documents 1..1

This section proposes a model that represents the one-to-one relationship for documents embedded in another document. In this case, the proposal is to use the representation of an individual Document within another element that represents a Document. In Figure 4, cardinality is also suggested to specify the one-to-one relationship type.

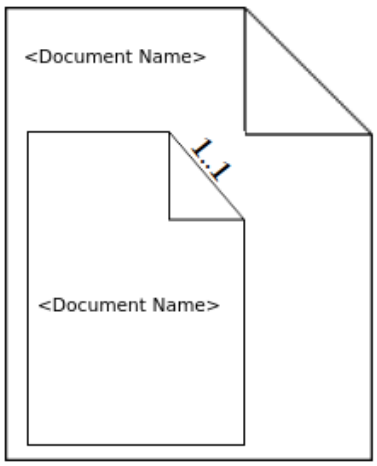


Figure 4: One-to-one relationship for embedded documents

4.4 Embedded Documents 1..N

A one-to-many relationship in embedded documents is represented by the Figure 5. This is the case when the notation to represent the cardinality is the same used in UML (F. Booch et al., 2005) and is placed in the upper right corner of the embedded documents. According to the cardinality one-to-many the larger document has embedded multiple documents within it.

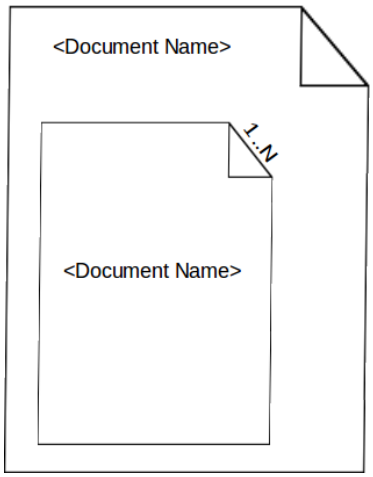


Figure 5: One-to-many relationship for embedded documents

4.5 Embedded Documents N..N

A many-to-many relationship in embedded documents is represented by the Figure 6. According to the cardinality many-to-many the larger document has a many to many relationship with the embedded document. The representation of the cardinality is the same used in UML (F. Booch et al., 2005).

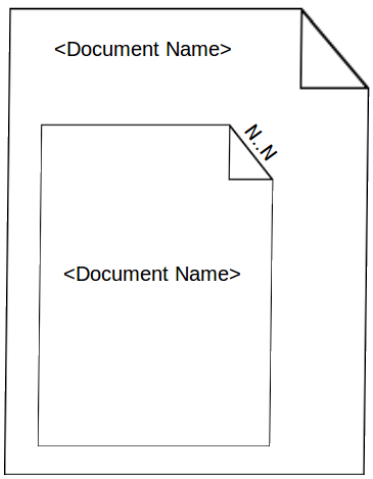


Figure 6: Many-to-many relationship for embedded documents

4.6 References 1..1

A document can reference another, and in this case, one must use an arrow directed to the referenced document, as shown in Figure 7. One can see that the directed arrow makes the left document references to the right document. Furthermore, the cardinality of the relationship should be specified above the arrow. The notation of cardinality is based on UML (F. Booch et al., 2005).

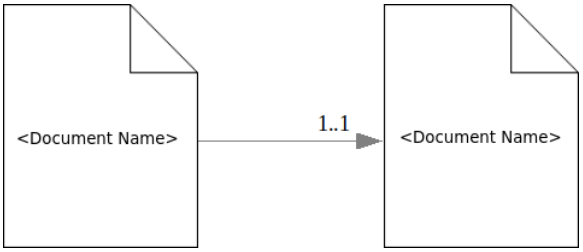


Figure 7: One-to-one relationship for documents referenced

4.7 References 1..N

In NoSQL, a document can reference multiple documents. To represent this relationship one should use an arrow directed to the referenced documents, as shown in Figure 8. The left document references multiple documents on the right side, by the directed arrow. Furthermore, the cardinality of the relationship is represented by the notation "1..N" as in UML (F. Booch et al., 2005).

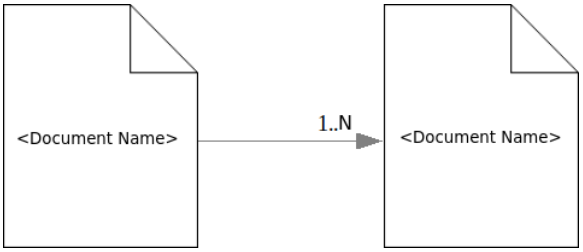


Figure 8: One-to-many relationship for documents referenced

4.8 References N..N

To represent this relationship a bidirectional arrow is used between reference documents, as shown in Figure 9. The left document references multiple documents on the right side and the right document references multiple documents on the left side. Furthermore, the cardinality of the relationship is represented by the notation "N..N" as in UML (F. Booch et al., 2005).

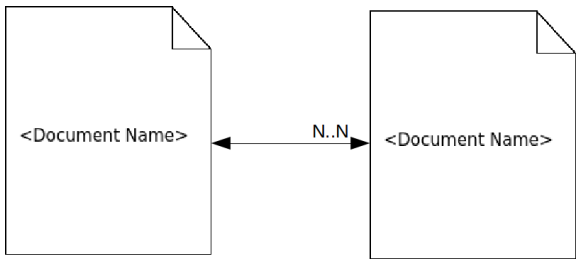


Figure 9: Many-to-many relationship for documents referenced

5 Case Study

In order to evaluate our proposal, part of the workflow described in (J. C. Marioni et al., 2014) was used. This workflow aimed at identify and comparing expression levels of human kidney and liver RNA samples sequenced by Illumina. The workflow was designed in three phases (Figure. 10):

- Filtering: all the sequenced transcripts were filtered, generating new files with good quality sequences.
- Alignment: transcripts were mapped to the human genome used as reference.
- Statistical Analysis: a sort process was first executed, followed by a statistical analysis with the mapped transcripts to discover which genes are mostly expressed both in kidney and liver samples.

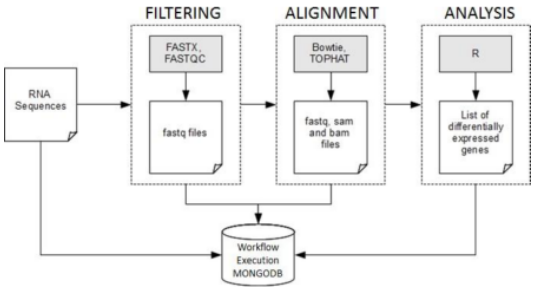


Figure 10: Workflow for analysis of differential expression among kidney and liver RNA samples

After analyzing the previously mentioned concepts, we have chosen to create a collection of documents for each PROV-DM type used to create a graph node. We also defined a collection for genomic documents (raw data). The reference relationship approach was chosen to connect all PROV-DM components, complementary information of PROV-DM and genomic documents. Based on (R. de Paula et al., 2013) we defined the documents and the attributes. A set of minimum information related to each one of these entities.

Figure. 11 shows our document based data representation, explained as follows:

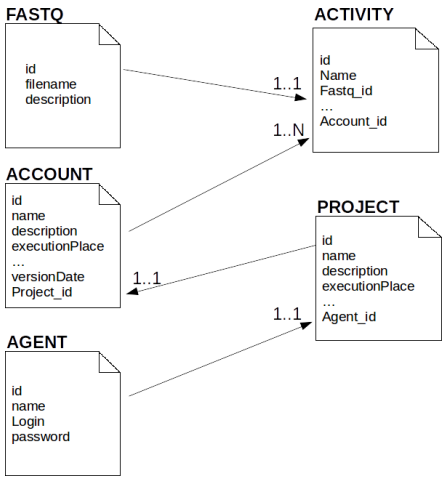


Figure 11: Data Modeling for Genomic Provenance for NoSQL Based on Documents.

- Project: stores different experiments of one agent. Attributes: Id, name, description, coordinator, start date, end date and observation.
- FASTAQ: files used or generated in the activity; Attributes: Id, filename and description.
- Activity: represents the execution of a program; Attributes: Id, name, program, version program, command line, function, start date, end date, account ID, used (name FASTQ, local, size), wasGenerateBy (name FASTQ, local, size), and wasAssociatedWith (Agent name).
- Account: represents the performance of an experiment; Attributes: Id, name, description, execution place, star date, end date, observation, version and version date and project Id.
- Agent: represents the person responsible for a program or a phase in the workflow. Attributes: Id, name, login and password.

5.1 Implementation

In this case study, we have considered the MongoDB NoSQL database to store provenance and data files. The primary motivation for this choice was MongoDB's ability to manipulate large volumes of data. MongoDB is an open-source Document-Oriented database designed to store large amounts of data from multiple servers. It uses JSON- style documents with dynamic schemas. The number of fields, content and size of the document can differ from one document to another. In practice, however, the documents in a collection share a similar structure (MongoDB, 2015) and can be mapped directly to a file format that can be easily manipulated by a programming language.

MongoDB documents have a maximum size of 16MB. This feature is important to ensure that a single document cannot use excessive amounts of RAM. In order to store files larger than the maximum size, MongoDB provides a GridFS API (MongoDB, 2015). It automatically divides large data into 256 KB pieces and maintains metadata for all pieces. GridFS allows for the retrieval of individual pieces as well as entire documents.

GridFS uses two collections to store the data: fs.files collections, containing metadata about files, and fs.chunks collections, which store the actual 256k data chunks. The collections FS.file contains the name of the FASTQ file. Thus, it was possible to implement the relationship between MongoDB Collection Activity using Reference Document. In other words, we implemented the connection between Level 1 and Level 2 through the File Name attribute that was present in fileprovenance.files and Activity Collection. Figure. 12 illustrates this particular implementation.

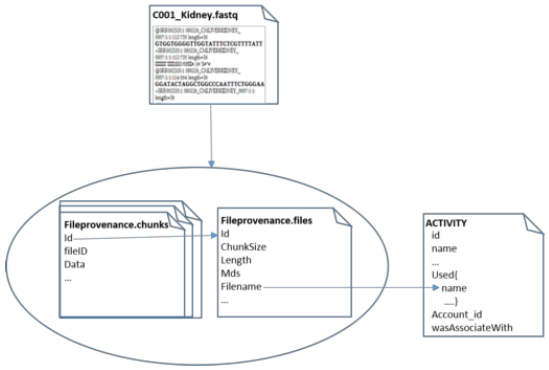


Figure 12: GridFS Implementation

6 Conclusions and Future Works

In contrast to relational database management systems, NoSQL databases are designed to be schemaless and flexible. Therefore, the challenge of this work was to introduce a data modeling standard for NoSQL document-oriented databases, in contrast to the original idea for NoSQL databases.

The objective was to build compact, clear and intuitive diagrams for conceptual data modeling for NoSQL databases. While the current studies propose generic techniques and do not define a specific modeling engine to NoSQL database, our idea was to present a graphical model for any NoSQL document-oriented database. Moreover, while other studies describe techniques based on UML Diagram Class and JSON format as a modeling solution, we have a new approach to solve the conceptual data modeling issue for NoSQL document-oriented databases.

Future work includes: verifying our model for other NoSQL database classifications, such as key-value and column.

References

R. Elmasri and S. Navathe. 2010. *Fundamentals of Database Systems*. Pearson Addison Wesley.

MongoDB. 2015. *Document database*. [Online] Available: `http://www.mongodb.org/` [Retrieved:April, 15].

J. Partner, A. Vukotic, and N. Watt. 2013. *Neo4j in Action*, O'Reilly Media.

D. Borthakur et al. 2011. *Apache hadoop goes realtime at facebook*, in Proceedings of the 2011 ACM SIGMOD International Conference on Management of data. ACM, 2011, pp. 1071–1080.

F. Chang et al. 2008. *"Bigtable: A distributed storage system for structured data*, ACM Transactions on Computer Systems (TOCS), vol. 26, no. 2, 2008, p. 4.

R. F. Lans. 2008. *Introduction to SQL: mastering the relational database language*, Addison-Wesley Professional.

H. Scalable. 2015. *Nosql data modeling techniques*. [Online] Available: `http://highlyscalable.wordpress.com/2012/03/01/nosql-data-modeling-techniques/` [Retrieved:April, 15].

R. Arora and R. Aggarwal, 2013. *Modeling and querying data in mongodb*, International Journal of Scientific and Engineering Research (IJSER 2013), vol. 4, no. 7, Jul. 2013, pp. 141–144.

K. Banker, 2011. *MongoDB in action*, Manning Publications Co.

D. Crockford, 2006. *RFC 4627 (Informational) The application json Media Type for JavaScript Object Notation (JSON)*, IETF (Internet Engineering Task Force)

K. Kaur, K.Rani, 2013. *Modeling and querying data in NoSQL databases*, In Big Data, IEEE International Conference on (pp. 1-7). IEEE.

S. J. Pramod, 2012. *Nosql distilled: A brief guide to the emerging world of polyglot persistence*,

MongoDB, 2015. *Data modeling introduction*, Online Available: `http://docs.mongodb.org/manual/core/data-modeling-introduction/` [Retrieved: April, 15].

CouchDB, 2015. *Modeling entity relationships in couchdb*, [Online]. Available: http://wiki.apache.org/couchdb/ [retrieved: April, 15]

G. Booch, J. Rumbaugh, and I. Jacobson, 2005. *The unified modeling language user guide.*, Pearson Education India.

J. C. Marioni, C. E. Mason, S. M. Mane, M. Stephens, and Y. Gilad, 2014. *RNA-SEQ: An assessment of technical reproducibility and comparison with gene expression arrays*, Genome Research, vol. 18, no. 9, pp. 1509–1517.

R. de Paula, M. Holanda, L. SA Gomes, S. Lifschitz and M. E. MT. Walter, 2013. *Provenance in bioinformatics workflows.*, BMC Bioinformatics 14 (Suppl 11):S6.

Hipi, as alternative for satellite images processing

Wilder Nina Choquehuayta
UNSA / Arequipa
wninac@unsa.edu.pe

René Cruz Muñoz
UNSA / Arequipa
rcruzm@unsa.edu.pe

Juber Serrano Cervantes
UNSA / Arequipa
jserranoc@unsa.edu.pe

Alvaro Mamani Aliaga
UNSA / Arequipa
amamani@unsa.edu.pe

Pablo Yanyachi
UNSA / Arequipa
raulpab@unsa.edu.pe

Yessenia Yari
UNSA / Arequipa
yyari@unsa.edu.pe

Abstract

These days, in different fields of both industry and academia, large amounts of data is generated. The use of several frameworks with different techniques is essential , for processing and extraction of data. In the remote sensing field, large volumes of data are generated (satellite images) over short periods of time. Information systems for processing these kind of images were not designed with scalable features. In this paper, we present an extension of the HIPI framework (Hadoop Image Processing Interface) for processing satellite image formats.

KeyWords: Big Data, Remote Sensing, Hadoop, HIPI, MapReduce, Satellite Images

1 Introduction

The field of remote sensing is helpful in different areas of both industry and academia, because it uses images of the earth's surface that are acquired from different sources like antennas and satellites, which provide increasingly better image resolution as technology advances. Nowadays, several open source frameworks are available to process big data, such as **Hadoop** (Shvachko et al., 2010), **H2O** (0xdata:H2O, 2015), **Spark** (Apache:Spark, 2015), etc., which are used for distributed and parallel processing of large volumes of data. HIPI (Hadoop Image Processing Interface) is an image processing library designed to be used with the Apache Hadoop MapReduce parallel programming framework. HIPI facilitates efficient and high-throughput image processing with MapReduce style parallel programs typically executed on a cluster. In the present work the HIPI library was modified, giving additional functionalities to read and process the GeoTIFF format (format provided by USGS -United States Geological Survey- for *Landsat* satellite images).

2 State of the Art

There are different techniques used for image classification in semantic taxonomy categories such as vegetation, water, etc. (Codella et al., 2011), however these methods don't consider scalability as part of its solutions, *Noel C. F. Codella et. al.*. In *Wanfeng Zhang, et. al.* (Zhang et al., 2013) An infrastructure for massive processing of satellite images in a multi-dataCenter environment, consisting of a DataCenter, where *Access Security*, *Information Service* and strategy *Scheduling* for data management us introduced. It's important to consider HIPI (Sweeney et al., 2011) as a state of the art, extensible library for image processing and computer vision applications, which helps to avoid the problem of small files and achieving improvements in memory and response time.

3 Proposal

That is why this paper is a modification of HIPI, to extends its functionality to work with TIFF images or GeoTIFF type. To achieve this, we proceeded as follows:

- It was decided to use the *Tiff* format from satellite images obtained from the USGS since this format unlike others has no compression or data loss (Adobe, 1992).
- JAI API was chosen to read and write the chosen format, JAI has more *codecs* and features available that can be useful for reading multiple formats.
- Classes needed to upload, encode and decode images of *Tiff* type were modified.

Based on the tests, the possibility of using HIPI for processing multispectral and hyperspectral images

was analyzed. For such images, operations such as PCA are important to process all spectral bands, it is proposed to keep them in the same zip file, or as different images belonging to a *tiff* format, and then decode and interpret as a conventional multi-band image.

4 Experiments

The experiments were performed on a Local Heterogeneous *Cluster* depicted in Table 1 where the characteristics of *slaves* and *master* are shown. We used satellite images from *LandSat 7*, only considering the first 4 bands so we compressed a satellite image in .zip then we used the .zip up to 0.5GB, 1GB, 5GB and 10GB. The algorithm was tested about the average of channels which is explained in the official website of HIPI. In each task map, we iterated over each read of band of satellite image as FloatImage, added each value of pixel depending of channel then divided for number of pixels (width x height) and returned the key of the satellite image and array of data calculated. In each reduce task, we only calculated the average of the average of channels from each satellite image.

Node	characteristics
master	*Core i7, RAM 8GB, Disk 100GB, S.O Ubuntu 64 bits*
slave 1	*Core i7, RAM 8GB, Disk 100GB, S.O Ubuntu 64 bits*
slave 2	*Core 2 Duo, RAM 4GB, Disk 100GB, S.O Ubuntu 64 bits*
slave 3	*Core 2 Duo, RAM 4GB, Disk 100GB, S.O Ubuntu 64 bits*

Table 1: Characteristics of the cluster

In Table 2 shows in axis x the amount of data in GBs and in y axis the execution time. The Hadoop configuration was 1 replication of data, chunks of 32MB, 64MB and 128MB, 4096MB in memory for task reduce and map. The java virtual machine for task reduce and map was configured with 4096MB at the most.

5 Conclusions

Based on the review conducted and theoretical experimental tests HIPI modified version of the article concludes as follows: It is possible to perform various image processing operations, such as filters, variance, clustering or dimensionality reduction by using the MapReduce algorithm and also while the information in compressed format occupies less space, this does not necessarily mean faster times when processing, since the matrix calculations are done on the same decompression.

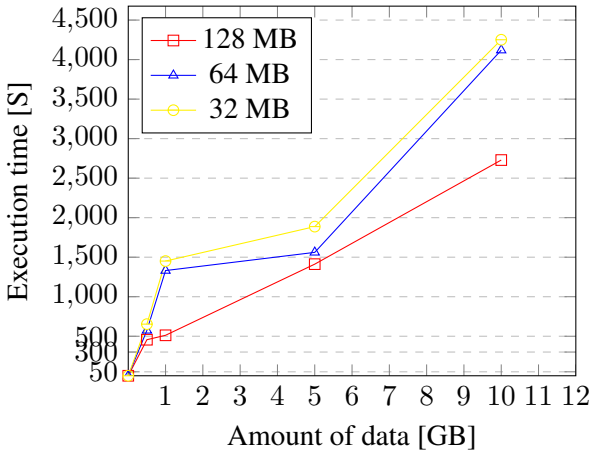


Table 2: Execution time vs Amount of data spatial

References

0xdata:H2O. 2015. H2o@ONLINE. Website. 0xdata:H2O, In: `http://0xdata.com/product/`,accessed(may 2015).

Apache:Spark. 2015. H2o@ONLINE. Website. Spark, Lightning-fast cluster computing , In : `https://spark.apache.org`, `accessed(accessed(2015-05-20))`.

Noel C.F. Codella, Gang Hua, Apostol Natsev, and John R Smith. 2011. Towards large scale land-cover recognition of satellite images. In *Information, Communications and Signal Processing (ICICS) 2011, 8th International Conference on*, pages 1–5. IEEE.

Konstantin Shvachko, Hairong Kuang, Sanjay Radia, and Robert Chansler. 2010. The hadoop distributed file system. In *Mass Storage Systems and Technologies (MSST), 2010 IEEE 26th Symposium on*, pages 1–10. IEEE.

Chris Sweeney, Liu Liu, Sean Arietta, and Jason Lawrence. 2011. Hipi: a hadoop image processing interface for image based mapreduce tasks. *Chris,University of Virginia*.

Wanfeng Zhang, Lizhe Wang, Dingsheng Liu, Weijing Song, Yan Ma, Peng Liu, and Dan Chen. 2013. Towards building a multi-datacenter infrastructure for massive remote sensing image processing. *Concurrency and Computation: Practice and Experience*, 25(12):1798–1812.

Multi-agent system for usability improvement of a university administrative system.

Jorge Leoncio Guerra Guerra
Universidad Nacional Mayor de San Marcos
Facultad de Ingeniería de Sistemas e Informática
Laboratorio de Robótica e Internet de las Cosas
jguerrag@unmsm.edu.pe

Félix Armando Fermín Pérez
Universidad Nacional Mayor de San Marcos
Facultad de Ingeniería de Sistemas e Informática
Laboratorio de Robótica e Internet de las Cosas
fferminp@unmsm.edu.pe

Abstract

The implementation of multi-agent systems is one of the specific ways to integrate heterogeneous information systems, by creating a software agent that performs specific tasks within a field of known action, as in the case of the travelling salesman problem or an agent searching clinical data of a patient. In this paper, the development of a mobile agent is proposed for the implementation of university administrative services across heterogeneous servers, interacting in turn with other intelligent agents in an heterogeneous communication environment. This will generate a single-access interface, which will serve to improve the usability of the system when the client makes an administrative request regardless of the server on which the requested procedure is processed.

Keywords: Integration, usability, agents, multi-agent systems, heterogeneous services.

1. Introduction

The portal of the University Inca Garcilaso de la Vega (http://www.uigv.edu.pe) is a web application that offers different services available in each of the offices of the institution, which is accessed through hyperlinks from Faculties, Research Units, Distance Education, etc.

An integrated system by software agents (Shiao, 2004); in which, with a single interface, a user can access different services without having to go through different pages to finish their request; will make possible to link these services regardless of the Faculty where they are and make the process requested by the user in a transparent manner, so that the usability of the system is also improved.

2. Proposed implementation

The structure of the proposed multi-agent system is shown in Figure 1.

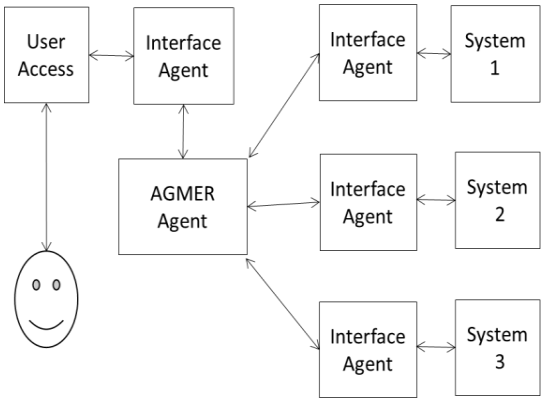


Figure 1 – Proposed multi-agent system

Using the GAIA methodology (Zambonelli et al, 2003) for modeling a system of agents, agents that will be built on the prototype are defined:

a.- Client Agent: captures requirements or service requests from users; interacts with the mobile agent to process the request and sends the response to the client system for viewing or joining the client process.

b.- Server Agent: interacts with the server of the corresponding office to transfer the request from the mobile agent, processes the requested service, and sends the response to the mobile agent.

c.- AGMER Agent: mobile agent moving through JADE middleware to interact with clients and servers, has the ability to divide received requests into subtasks performed by different servers.

One advantage of using GAIA is that it can be combined with diagrams from UML as described in Kang et al (2004), for this reason UML diagrams and GAIA models (Wooldridge et al is used., 2000) will be used for its construction.

Once developed, the model will be implemented as shown in Figure 2.

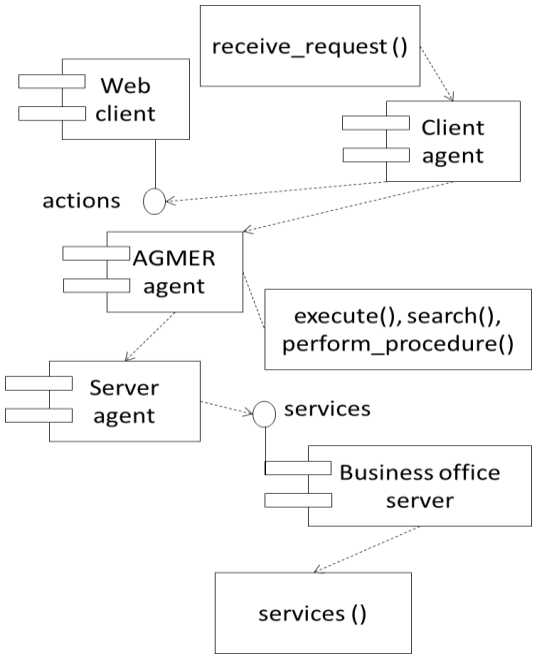


Figure 2. Components.

In the development process is suggested the use of Eclipse Luna, JDK 8.0_41 and JADE 4.3.2 for creating agents. In the prototype, it was initially considered the access to these basic services for a university student from the case study:

a. Specialized Library System with Struts 2, running on Google App Engine cloud, a PAAS solution.

b. PRODI (English Learning Program) registration system with Node.JS, OpenShift, a PAAS solution.

All modules will communicate via web services, with interface agents using AXIS 2.

3. Conclusions and future work

The single-interface model for administrative procedure systems is suitable to improve the usability of a web portal to reduce the number of interactions.

A multi-agent system allows an heterogeneous implementation of components as well as asynchronous communication similar to a request of a university administrative procedure.

JADE is a suitable platform for developing agents due to its ease of use, and adaptability to development methodologies of multi-agent systems.

As future work it is proposed to develop indicators to measure implemented services, the frequency of use of the services, and others, that will be collected and stored in the cloud for future analysis using Big Data tools.

4. References

Dan Shiao. 2004. *Mobile Agent: New Model of Intelligent Distributed Computing*. IBM China, October, 2004.

Franco Zambonelli, Nicholas Jennings, Michael Wooldridge. 2003. *Developing Multiagent Systems: The Gaia Methodology*. ACM Transactions on Software Engineering and Methodology. 12(3): 317–370.

Miao Kang, Lan Wang, Kenji Taguchi. 2004. *Modelling Mobile Agent Applications in UML2.0 Activity Diagrams*, Proceedings of the 6th International Conference on Enterprise Information Systems. Porto. Portugal. Vol. IV: 519-522.

Michael Wooldridge, Nicholas Jennings, David Kinny. 2000. *The Gaia Methodology for Agent-Oriented Analysis and Design*. Journal of Autonomous Agents and Multi-Agent Systems, vol. 15. Kluwer Academic Publishers, Boston. The Netherlands.