Avances en la Ingeniería del Lenguaje y del Conocimiento

Research in Computing Science

Series Editorial Board

Research in Computing Science es una publicación trimestral, de circulación internacional, editada por el Centro de Investigación en Computación del IPN, para dar a conocer los avances de investigación científica y desarrollo tecnológico de la comunidad científica internacional. **Volumen 85,** diciembre de 2014. Tiraje: 500 ejemplares. *Certificado de Reserva de Derechos al Uso Exclusivo del Título* No. : 04-2005-121611550100-102, expedido por el Instituto Nacional de Derecho de Autor. *Certificado de Licitud de Título* No. 12897, *Certificado de licitud de Contenido* No. 10470, expedidos por la Comisión Calificadora de Publicaciones y Revistas Ilustradas. El contenido de los artículos es responsabilidad exclusiva de sus respectivos autores. Queda prohibida la reproducción total o parcial, por cualquier medio, sin el permiso expreso del editor, excepto para uso personal o de estudio haciendo cita explícita en la primera página de cada documento. Impreso en la Ciudad de México, en los Talleres Gráficos del IPN – Dirección de Publicaciones, Tres Guerras 27, Centro Histórico, México, D.F. Distribuida por el Centro de Investigación en Computación, Av. Juan de Dios Bátiz S/N, Esq. Av. Miguel Othón de Mendizábal, Col. Nueva Industrial Vallejo, C.P. 07738, México, D.F. Tel. 57 29 60 00, ext. 56571.

Editor responsable: *Grigori Sidorov, RFC SIGR651028L69*

Research in Computing Science is published by the Center for Computing Research of IPN. **Volume 85**, December 2014. Printing 500. The authors are responsible for the contents of their articles.

Printed in Mexico City, in the IPN Graphic Workshop – Publication Office.

Volume 85

Avances en la Ingeniería del Lenguaje y del Conocimiento

David Pinto
Darnes Vilariño (ed.)

Instituto Politécnico Nacional, Centro de Investigación en Computación
México 2014

ISSN: 1870-4069

Instituto Politécnico Nacional (IPN)
Centro de Investigación en Computación (CIC)
Av. Juan de Dios Bátiz s/n esq. M. Othón de Mendizábal
Unidad Profesional "Adolfo López Mateos", Zacatenco
07738, México D.F., México

http://www.rcs.cic.ipn.mx
http://www.ipn.mx
http://www.cic.ipn.mx

Indexed in LATINDEX and Periodica / Indexada en LATINDEX y Periódica

Printing: 500 / Tiraje: 500

Printed in Mexico / Impreso en México

Prefacio

El simposio en Ingeniería del Lenguaje y del Conocimiento (LKE‘2014) es la segunda edición de esta serie de eventos. Esta conferencia ha sido organizada en el seno de la Facultad de Ciencias de la Computación de la Benemérita Universidad Autónoma de Puebla (BUAP) por dos años consecutivos. Nace como una iniciativa del laboratorio de Ingeniería del Lenguaje y del Conocimiento con la finalidad de ofrecer un espacio académico y de investigación en el cual sea posible reportar trabajos relacionados con el área.

Por supuesto que este evento promueve la cooperación entre diferentes grupos de investigación pues permite el intercambio de resultados científicos, prácticas y nuevo conocimiento.

El presente volumen contiene una serie de contribuciones originales que han sido seleccionadas a partir de un proceso de evaluación ciega doble (*double blind*), lo cual significa que los nombres de los autores de los artículos y los nombres de los revisores son ambos desconocidos. Este procedimiento es ejecutado en aras de proveer una evaluación anónima que derive en artículos de mayor calidad para este volumen; particularmente, en esta ocasión la tasa de rechazo fue del 26%, cuidando que en todos los casos, al menos dos especialistas del comité revisor hicieran una evaluación de la pertinencia, originalidad y calidad de cada artículo sometido.

Esperamos que este volumen sea de utilidad para el lector y que este segundo simposio en sí mismo sea un espacio de intercambio científico productivo que enriquezca la colaboración entre estudiantes y académicos en el ámbito de la ingeniería del lenguaje y del conocimiento.

El proceso de revisión y selección de artículos se llevó a cabo usando el sistema libremente disponible llamado EasyChair, http://www.easychair.org.

Diciembre 2014

David Eduardo Pinto Avendaño
Darnes Vilariño Ayala

Table of Contents

Page

Bilingual and Cross Domain Politics Analysis

Jean-Valère Cossu[1], Rocío Abascal-Mena[2], Alejandro Molina[4], Juan-Manuel Torres-Moreno[1,3] and Eric SanJuan[1]

[1]LIA/Université d'Avignon et des Pays de Vaucluse, Avignon, France
[2]Universidad Autónoma Metropolitana-Cuajimalpa, México DF, México
[3]École Poytechnique de Montréal, Montréal, (Québec) Canada
[4]Comisión Nacional para el Conocimiento y Uso de la Biodiversidad, México DF, México
rocio.abascal@gmail.com, molina@conabio.gob.mx,
{jean-valere.cossu, juan-manuel.torres, eric.sanjuan}@univ-avignon.fr

Abstract. Opinion mining on Twitter recently attracted research interest in politics using Information Retrieval (IR) and Natural Language Processing (NLP). However, getting domain-specific annotated data still remains a costly manual step. In addition, the amount and quality of these annotation may be critical regarding the performance of machine learning (ML) based systems. An alternative solution is to use cross-language and cross-domain sets to simulate training data. This paper describe a ML approach to automatically annotate Spanish tweets dealing with the online-reputation of politicians. Our main finding is that a simple statistical NLP classifier without in-domain training can provide as reliable annotation as humans annotators and outperform more specific resources such as lexicon or in-domain data.

1 Introduction

Sentiment Analysis (SA) is useful in the study of online communication because it gives researchers the ability to automatically measure emotion in online texts [30]. The rise of social web, particularly Twitter[1], provides new opportunities to collect real time data in large quantities directly from users. Tweets can be analysed in order to track reactions to events. One important aspect of the tweets is that they have been used as a way to participate in social movements as well to make public opinions and reactions to different events. These opinions are charged with sentiment whether they can be positive or negative, toward a movement or event, as they can change over time.

Since Twitter provides the possibility to extract tweets and compose actual corpus there have been a lot of linguistic research applied in tweets. Using publicly available online data to perform sentiment studies significantly reduces

[1] http://www.twitter.com

the costs, efforts and time needed to administer large-scale public surveys and questionnaires [8]. Specially, we find that Twitter is often used to analyse political preferences by studying the use of humor contained in tweets [8] where POMS (Profile of Mood States) was used to distill, from Twitter messages, time series corresponding to 6 different emotional attributes (tension, depression, anger, vigor, fatigue and confusion). POMS is a psychometric instrument that provides a list of adjectives for which the patient has to indicate the level of approval. Each adjective is related to a state of mind and, therefore, the list can be exploited as the basis for a mood-analyser of textual data.

Politics have already been addressed in previous works but mostly in English as [22] or [35]. [23], used a subjective lexicon that comes from the Opinion Finder in order to determine positive and negative scores for each data set corresponding to a tweet. In this case, the raw numbers of positive and negative tweets about a given topic are used to calculate a confidence score (the relation between the number of positive and negative tweets). The authors indicated that by a simple manual inspection of the tweets they have found examples that have been classified incorrectly based on a feeling. Nevertheless, the authors used this method to measure some issues such as the consumer confidence, the presidential approval and the 2008 presidential election in the United States.

To the extend of our knowledge, political studies from a ML point-of-view in Spanish or French are rare [34]. However the role of social networks during the Presidential campaign of 2012 in Mexico gained great importance as the principal instrument for exercising public opinion, especially for young people. The youth organization "yosoy132" born during the election campaign in Mexico in 2012, thanks to the social networks, joining youth groups from all universities in the country regardless their social conditions. Studies conducted by the *Instituto Nacional de Estadística, Geografía e Informática* (INEGI)[2] show that 40.3% of users of ICT in Mexico are young people who communicate via social networks and mobile phones and that they remain connected most of the day. This percentage equals 15.3 million people aged 18 to 34 who were potential voters of the political election [29]. In this way, in 2012, we found that participation via Twitter increased creating significant social and political communities around specific problems.

Usual studies in the domain assume that a great effort of acquisition of the tweets and a subsequent manual labeling process is required. In addition, a validation process is needed to correct the errors introduced by manual labeling. Even using crowd-sourcing-based solutions it is a very expensive task both in money and time. Moreover important political events will always occur faster than our capacities of getting manually annotated data in several languages. In this context, we propose an approach that can provide a reliable pre-annotation using out-of-domain data which needs only a light supervision before validation in order to obtain a reliable corpus that can be used for more complex political studies such as user Political Tendency or to monitor politicians reputation.

The rest of this paper is organized as following. Section 2 gives a focused

[2] National Institute of Statistics, Geography and Informatics of Mexico.

overview of related work. In section 3 we describe the main characteristics of the data-set. In section 4 we propose our approaches while section 5 is devoted to a thoroughly evaluation. Finally, section 6 gives some conclusions about our work and opens several perspectives.

2 Tweets mining and sentiment analysis

[32] presented in 2010 a job with two distinct parts: in the first one LIWC (Linguistic Inquiry and Word Count) is used to perform a superficial analysis of the tweets related to the different political parties that competed for the German Federal election in 2009. In the second part, the authors claim that the mere counting of tweets with references to one of the parties, accurately reflects the election results. On the other hand, they established that the MAE (Mean Absolute Error) of the "prediction" based on Twitter data was very close to the real surveys that were carried out.

An increasing number of empirical analyses of sentiment and mood are based on textual collections of data generated on Twitter as they used sophisticated algorithms to pre-process, apply grammatical rules and classify them in mood categories. In this way we find, for example, the use of a lexicon based classifier as a dataset is also classified using SVM and/or Naive Bayes [36,20]. A classifier is developed specifically tuned for tweets, using key words, phrases and emoticons to determine the mood of each tweet [6]. Several methods have been already proposed for exploiting tweets in order to detect people's mood changes throughout the day [20,19].

[10] has measured changes in the mood of the U.S. population, over three years, from tweets, providing policy relevant indicators. In general, studies analysing tweets by combining different sentiment analysis algorithms have been able to give new insights into human behaviour as a result [11,12,17,18]. These works show that there is a tremendous ambition to develop opinion mining tools for social media in order to be able to distinguish what is important and interesting [21].

The research field of SA, also known as opinion mining, has developed many algorithms to identify whether an online text is subjective or objective, and whether any opinion expressed [24]. Another way to identify polarity is based on the use of lexicons. There are lexicons like SentiWordNet[3] [3] that attaches real-valued sentiment scores to WordNet synsets [15]. [7] maintains and freely distributes a sentiment lexicon which contains morphological variants, slang, and social-media mark-up in order to be able to analyse sentences even if they are misspelling.

Nevertheless theses approaches miss an important aspect of Spanish language and politics domain: the irony. Irony is the creative use of language in order to make fun of something or someone. The boundaries between irony and sarcasm are vague. It seems that Spanish is a language that uses a lot of irony in the

[3] `http://sentiwordnet.isti.cnr.it/`

communications [1]. The Spanish language considered irony a "soft and veiled mockery", while sarcasm is a "hard and bloody mockery". Thus, the only difference is the degree of cruelty associated. For example, to ironize over a retarded person saying: "He is a genius" would be a sarcasm, and the same remark referred to a colleague who has been successful by chance solve a problem, an irony. Sarcasm is then an extreme form of irony [1]. Detection of irony is a difficult subject, and has been subject matter of various disciplines such as linguistics and psychology, among other [26]. Studies of [5] and [4] on Tweeter data shown ironic English tweets classification using Decision Trees. Sarcasm detection in tweets and Amazon has been realized by SASI [13] using the Mechanical Turk to create a gold-standard. To our knowledge, there are no studies of irony or sarcasm on Spanish or French Tweets.

While [9], [33] and [34] obtained agreement percentage quite similar to other studies over sentiment analysis task, they all agreed that human interpretation of these kind of more or less consensual contents is prone to mistakes. As both facts and opinions have to be considered, regardless of whether the content is opinionated or not, it is sometimes hard to tell what implications a message may have on the reputation of a given entity. The political context finally makes the task harder.

In this work we investigate how much ML without correct training data can perform compared to humans annotators.

3 Approaches

We mainly used two approaches in this paper: the lexicon approach and the Machine Learning approach.

3.1 Lexicon approach

Our collection was analysed by using a lexicon approach combined with a linguistic analysis in order to detect sentiments, during a period of time, in social and political tweets. The lexicon approach starts with a list of positive and negative words, which are already pre-coded for polarity. A linguistic analysis, in contrast, exploits the grammatical structure of text to predict its polarity, in conjunction with the lexicon [30]. Words contained in a tweet are classified into positive or negative by using the previous lexicon. Nevertheless, this methodology does not takes into account the sarcasm which transforms the polarity of an apparently positive or negative utterance into its opposite [16]. But by analysing a big corpus the sarcasm rest minimum and do not contributes to inflate in a big amount the percentage of the total results. The corpus is pre-processed in order to extract stop-words, punctuation, links, etc. Then, the Spanish and the English translated lexicon, respectively, are used to count for each tweet and for each corpus the number of positive and negative words contained in each tweet. All the process is automatically performed by using R[4].

[4] R is an interpreted computer language designed for statistical data analysis (`http://www.r-project.org/`

3.2 Machine learning approaches

As the process described above is not sufficient, we propose an alternative approach using probabilistic ML. In works such as [9] and during TASS [34] ML was partly used to assist annotators and propose annotations. [14] showed that a small annotated set coupled to ML could perform competitively to annotators to answer text mining tasks. The annotation was addressed as a classification problem that consisted of determining the polarity of each tweet. A very large collection of ML algorithms can be used for classification tasks. We chose to use a multi-class SVM-based approach[5], a Cosine similarity and a baseline. The baseline was computed as simple memory test which consists in tagging each tweet with the most similar tweet in the training set (according to Jaccard index). All these approaches rely on a discriminant bag-of-words representation (DF, IDF and Gini) [31] of each tweet (we considered n-gram with $n \leq 3$). The bag-of-words is built after the following pre-processing:

- words are lower-cased;
- stop-words[6], links and punctuation are removed;

Then we estimate the similarity of a given tweet by comparing it to each class's bag-of-words and rank tweets according to the classifier value.

4 Datasets and Descriptive Statistics

In this section we provide a short description of each data-set we used. To investigate to classification issue we wanted to compare the use of similar data with a French annotated corpora of tweets dealing with politics and with a collection of tweets dealing with corporate entities reputation.

4.1 Description of the Spanish political set

The corpus analysed concerns 800 tweets containing #AMLO that were extracted between the period of 9, 10 and 11 June 2012. AMLO is the acronym for Andrés Manuel López Obrador, who was a left candidate to the Presidential elections in Mexico. AMLO has built a strong base of support among people who feel that they have been left behind as Mexico's economy grows and evolves. He had the loyalty of a young generation who were frustrated with the country's monopolistic media sector (mainly represented by Televisa[7]).

As shown in table 1 classes are well balanced with only a slightly difference with negative tweets for the complete collection as well as for the French subpart.

[5] Multi-Class Support Vector Machine, see: `http://www.cs.cornell.edu/people/tj/svm_light/svm_multiclass.html`

[6] Stop-list from Oracle (`http://docs.oracle.com`).

[7] Which currently dominates approximately 70% of the Mexican broadcasting and pay-TV market: `http://www.nasdaq.com/article/strict-telecom-laws-to-check-mexican-duopoly-analyst-blog-cm334020`

Table 1. Class distribution in both complete and French sub-part collection

Class	Class-Distribution	Class-Distribution (French)
Negative	0.41	0.37
Neutral	0.29	0.30
Positive	0.30	0.33

4.2 Statistics on the French political

We provide here some details about the French data-set (more detailed statistics are available in [33]) about the two main candidates at the last french presidential election (May, 2012). Tweets were extracted from March 2012 (before the election) to December 2012 (after the election) This collection has been manually annotated by thirty post-graduate persons regarding polarity (6 levels) and topic. It comprises more than 11,527 annotations (7,283 unique tweets, half have been annotated twice or more). In our study we only consider the polarity annotation and for a reasonable analysis, we considered only 3 polarity level.

Table 2. Class distribution in the French political collection

Class	Class-Distribution
Negative	0.60
Neutral	0.12
Positive	0.28

Table 2 shows that the main tendency is negative wit a very few number of neutral tweets. The main reason is that politics in France unleash passions between people.

4.3 Statistics on the Spanish annotated set

We used the Spanish sub-part (23,100 tweets of the RepLab 2013 campaign [2]) collection which is a large multilingual collection of tweets referring to a set of 61 entities spread into four domains: automotive, banking, universities and music/artists. For each entity, at least 2,200 tweets have been collected covering a period going from the 1^{st} of June 2012 to the 31^{st} of December 2012. These tweets have been manually annotated by experts according to filtering (related or not to the entity), polarity for reputation (3 levels), topic (clustering) and priority (topic ranking). We will only use the polarity annotation. Table 3 shows that the main tendency of the RepLab set is positive. Crossing this point with the negative view from the French collection should provide an interesting result.

Table 3. Class distribution in the Spanish reputation collection

Class	Class-Distribution
Negative	0.24
Neutral	0.28
Positive	0.48

4.4 Metrics

The measures selected to evaluate our approach were the Precision, the Recall, and the F-Score for each class. The F-Score reported in tables 4 and 5 is the the Macro Averaged F-Score computed as mean of each class F-Score.

$$\text{F-Score}_c = \frac{2 \times (\text{Precision}_c \times \text{Recall}_c)}{\text{Precision}_c + \text{Recall}_c} \tag{1}$$

With Precision P_c for class c as:

$$P_c = \frac{\text{Num. of documents correctly assigned in class}_c}{\text{Num. of documents assigned in class}_c} \tag{2}$$

And Recall R_c for class c as:

$$R_c = \frac{\text{Num. of documents correctly assigned in class}_c}{\text{Num. of documents belonging to class}_c} \tag{3}$$

Results are also compared according to Accuracy as it is a easy interpretable measure and it represents the informativeness of a system.

5 Experimental evaluation

Tables 4 and 5 summarize the experimental results of our proposal concerning the tweets polarity.

5.1 Classification using translated data

Table 4. Polarity classification results using French set

Method	F-Score	Accuracy
Baseline	**0.39**	**0.42**
Cosine	0.24	0.36
SVM	0.33	0.37

As a first experiment we choose to manually translate a sample (220 tweets) of our unlabeled set in order to perform a classification using the French annotated set training set. According to Table 4 systems performance using same context data is really low. The main reasons are that the vocabulary used to described the French candidates is not the same as the one used for AMLO but also that both class distributions are too different.

5.2 Classification over Spanish contents

Table 5. Polarity classification results using Spanish set

Method	F-Score	Accuracy
Baseline	0.50	0.51
Cosine	**0.74**	**0.74**
ElhPolar Lexicon	0.25	0.32
Translated Lexicon	0.21	0.33
SVM	0.17	0.31

In a second experiment we performed the classification using RepLab 2013 reputation set. Table 5 shows classification performance over Spanish contents according to F-Score and Accuracy. An interesting result is the systems' ranking while the Cosine similarity was outperformed with the French sets it is interesting to notice that we are able to obtain quite good classification results that are close to inter-annotator agreements observed in the literature [2,34,25], while SVM performance dramatically decreases. Baseline performance is also quite interesting since his performance increases but remains lower than the Cosine it was significantly better on the French set. Both lexicon approaches (ElhPolar lexicon [27] and Bing Liu translated one) seem to not fit our data-set or this kind of analysis since they do not perform well.

Some contents such as:
"RT 1. Naces 2. Eres AMLO 3. Creces 4. No eres presidente. 5. No eres presidente. 6. No eres presidente. 7. No eres presidente. 8. Mueres. JAJA" (In English: 1. You're born 2.You're AMLO 3. You grow 4. You're not president. 4. You're not president. 6. You're not president. 7. You're not president 8. You die. LOL LOL")
are tagged positive by the systems while they are really negative for AMLO. It is a typical example of humoristic contents that systems are not able to handle properly.

In this another example:
"AMLO gran orador cada vez que abre la boca sueña #elpejeaburrehastaalospejezombies" ("AMLO great speaker every-time he opens the mouth he dreams" in English)
is an irony because people are not dreaming about a better country instead they are becoming tired and almost falling at sleep every time that AMLO speaks.

6 Conclusions

In this paper we described and compared several approaches for a fast political classification of Spanish tweets. Our experimental evaluation (although our test set was limited) establish that without specific training material we can achieve results comparable to state of art. Then, while the literature insists on the need

of specific training data, our experiments have shown that the need of costly experts annotation can be reconsidered.

At first we intend to apply this process to others candidates and other existing data-set. We have several ideas on how to improve our approach to identifying the polarity in political tweets using information carried in Hashtags and Twitter users'name. The detection of irony and the study of re-tweet phenomena [38] can be two important elements to well classify tweets. In forthcoming works, we think to study in detail the impact of these phenomena in the micro-blogs classification.

6.1 Acknowledgment

This work is funded in part by the project ImagiWeb ANR-2012-CORD-002-01 (France).

References

1. Alonso Quecuty M-L. and Castillo M-D. *Detectando la ironía: La hipótesis aditiva como alternativa a las de referencia y la intención* In Cognitiva, Vol. 3, N^o 1, 1991, pp. 95-122
2. Amigó, E., Corujo, A., Gonzalo, J., Meij, E., de Rijke, M. *Overview of RepLab 2013: Evaluating Online Reputation Management Systems* CLEF 2013 Labs and Workshop Notebook Papers (2013).
3. Baccianella S., Esuli A., and Sebastiani F. *SentiWordNet 3.0: an enhanced lexical resource for sentiment analysis and opinion mining* In Proceedings of the Seventh Conference on International Language Resources and Evaluation, pp. 2200-2204. European Language Resources Association.
4. Barbieri F. and Saggion H. *Modelling Irony in Twitter.* Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, EACL, April 26-30, 2014, Gothenburg, Sweden, pp. 56–64.
5. Barbieri F. and Saggion H. *Modelling Irony in Twitter: Feature Analysis and Evaluation* In Proceedings of Language Resources and Evaluation Conference (LREC) 2014 4258-4264
6. Bertrand, K. Z., Bialik, M., Virdee, K., Gros, A., and Bar-Yam, Y. *Sentiment in New York City: A High Resolution Spatial and Temporal View* arXiv preprint arXiv:1308.5010.
7. Bing Liu *Sentiment Analysis and Opinion Mining* Morgan & Claypool Publishers, May 2012.
8. Bollen J. Mao H. and P. A. *Modelling public mood and emotion: Twitter sentiment and socio-economic phenomena* In Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media. 2011.
9. Carrillo-de-Albornoz, J., Amigó, E., Spina, D., and Gonzalo, J. *ORMA: A Semi-automatic Tool for Online Reputation Monitoring in Twitter* In Advances in Information Retrieval (pp. 742-745). Springer International Publishing.
10. Cha M., Haddadi H., Benevenuto F. and Gummadi K. P. *Measuring User Influence in Twitter: The Million Follower Fallacy* In Proceedings of the 4th International AAAI Conference on Weblogs and Social Media (ICWSM), Washington, DC, 2010.

11. Chung, J. E., and Mustafaraj, E. *Can collective sentiment expressed on Twitter predict political elections ?* In Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence (AAAI 2011) pp. 1768-1769. Menlo Park, CA: AAAI Press.
12. Dodds, P. S., and Danforth, C. M. *Measuring the happiness of large-scale written expression: Songs, blogs, and presidents* Journal of Happiness Studies, 11(4), 441-456.
13. Dadidov D., Tsur O., and Rappoport A. *Semi-supervised recognition of sarcastic sentences in Twitter and Amazon* In Proceedings of the Fourteenth Conference on Computational Natural Language Learning. Association for Computational Linguistics, 2010. p. 107-116.
14. Fabbrizio GD, Tur G and Hakkani-Tür D 2004 *Bootstrapping spoken dialogue systems with data reuse* Proceedings of the 5th SIGdial Workshop on Discourse and Dialogue.
15. Fellbaum C. (1998) *WordNet: An Electronic Database* Cambridge, MIT Press.
16. González-Ibáñez, R., Muresan, S. and Wacholder, N. *Identifying sarcasm in Twitter: a closer look* In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies. Short papers-Volume 2. Association for Computational Linguistics, 2011. p. 581-586.
17. Gruzd, A., Doiron, S., and Mai, P. *Is happiness contagious online? A case of Twitter and the 2010 Winter Olympics* In Proceedings of the 44th Hawaii International Conference on System Sciences. Washington, DC: IEEE Computer Society.
18. Kramer, A. D. I. *An unobtrusive behavioral model of "Gross National Happiness"* In Proceedings of CHI 2010 (pp. 287-290). New York: ACM Press.
19. Lampos V, Preotiuc-Pietro D, and Cohn T *A user-centric model of voting intention from Social Media* In Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL 2013).
20. Martínez, V., and González, V. M. *Sentiment Characterization of an Urban Environment via Twitter* In Ubiquitous Computing and Ambient Intelligence. Context-Awareness and Context-Driven Interaction (pp. 394-397). Springer International Publishing.
21. Maynard, D., Bontcheva, K., and Rout, D. *Challenges in developing opinion mining tools for social media* In Proceedings of NLP can u tag # user generated content, 2012.
22. Malouf R. and Mullen T. 2008. *Taking sides: User classification for informal online political discourse* In Internet Research, volume 18, pages 177-190.
23. O'Connor B., Balasubramanyan R., Routledge B. and Smith N. 2010. *From tweets to polls: Linking text sentiment to public opinion time series* In International AAAI Conference on Weblogs and Social Media.
24. Pang, B., and Lee, L. *Opinion mining and sentiment analysis. Foundations and Trends* In Information Retrieval 1(1-2), 1-135.
25. Pla F. and Hurtado L. *Political Tendency Identification in Twitter using Sentiment Analysis Techniques* In Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers pp183-192, Dublin (Ireland)
26. Reyes A., Rosso P., and Veale T. *A Multidimensional Approach for Detecting Irony in Twitter.* In Language Resources and Evaluation, vol. 47, issue 1, pp. 239-268.
27. Saralegi X., San Vicente I. *Elhuyar at TASS 2013.* In Proceedings of "XXIX Congreso de la Sociedad Española de Procesamiento de Lenguaje Natural". Workshop on Sentiment Analysis at SEPLN (TASS2013). Madrid. pp. 143-150.
28. Stone, P. J., Dexter C. D., Marshall S. S., and Ogilvie D. *The General Inquirer: A Computer Approach to Content Analysis.* Cambridge, MA: MIT Press.

29. Tello-Leal E., Tello-Leal D. A. and Sosa Reyna C.M. *Reflexiones sobre el uso de las tecnologías de información y comunicación en las campañas electorales en México: e-campañas.* Revista Virtual Universidad Católica del Norte, 36 (2012), 33-47.
30. Thelwall, M., Buckley, K., and Paltoglou, G. *Sentiment in Twitter events.* Journal of the American Society for Information Science and Technology, 62(2), 406-418.
31. Torres-Moreno, J.-M., El-Beze, M., Bellot, P. and Bechet, *Opinion detection as a topic classification problem* In in Textual Information Access. Chapter 9, pp 337, John Wiley & Son. 2013
32. Tumasjan, A., Sprenger, T. O., Sandner, P. G., and Welpe, I. M. *Predicting Elections with Twitter: What 140 Characters Reveal about Political Sentiment* ICWSM, 10, 178-185.
33. Velcin J., Kim Y.-M, Brun C., Dormagen J.-Y, SanJuan E., Khouas L., Peradotto A., Bonnevay S., Roux C., Boyadjian J., Molina A. and Neihouser M. *Investigating the Image of Entities in Social Media: Dataset Design and First Results* Proceedings of Language Resources and Evaluation Conference (LREC), 2014
34. Villena-Román J. and García-Morera, J. *TASS 2013-Workshop on Sentiment Analysis at SEPLN 2013: An overview.*
35. Wang H., Can D., Kazemzadeh A., Bar F. and Narayanan S. *A system for real-time twitter sentiment analysis of 2012 U.S. presidential election cycle* In Proceedings of the ACL 2012 System Demonstrations, ACL'12, pages 115-120. Association for Computational Linguistics
36. Wijaya, V., Erwin, A., Galinium, M., and Muliady, W. *Automatic mood classification of Indonesian tweets using linguistic approach* In 2013 International Conference on Information Technology and Electrical Engineering (ICITEE). pp. 41-46. IEEE.
37. Wilson T., Wiebe J., and Hoffmann P. *Recognizing Contextual Polarity in Phrase-Level Sentiment Analysis.* In Proceedings of HLT-EMNLP-2005.
38. Morchid M., Dufour R., Bousquet P-M., Linarès G., Torres-Moreno, J-M. *Feature selection using Principal Component Analysis for massive retweet detection*, Pattern Recognition Letters ; DOI: 10.1016/j.patrec.2014.05.020

Pertinence of Lexical and Structural Features for Plagiarism Detection in Source Code*

A. Ramírez-de-la-Cruz, G. Ramírez-de-la-Rosa,
C. Sánchez-Sánchez, H. Jiménez-Salazar, and E. Villatoro-Tello

Departamento de Tecnologías de la Información,
División de Ciencias de la Comunicación y Diseño,
Universidad Autónoma Metropolitana Unidad Cuajimalpa. México D.F.
aaron.rc24@gmail.com
{gramirez,csanchez,hjimenez,evillatoro}@correo.cua.uam.mx

Abstract. Source code plagiarism can be identified by analyzing several and diverse views of a pair of source code. In this paper we present three representations from lexical and structural views of a given source code. We attempt to show that different representations provide diverse information that can be useful to identify plagiarism. In particular, we present representations based on 3-grams of characters, data type of function's signatures and names of the identifiers of function's signatures. While we used only three representations, more representations can be added. We conducted our analysis over a collection of 79 source code written in C language. Our results show that n-gram representation is very informative, but also that representations taken from the function's signatures are, to some extend, complementaries.

Key words: Lexical and Structural Features, Similarity Calculation, Document Representation, Plagiarism Detection, Natural Language Processing

1 Introduction

Plagiarism detection in source code is a topic of growing interest of many researches since 1980; specifically when institutions such as the Carnegie-Mellon University created policies for their computer science departments [1] to prevent plagiarism among their students. Recent studies [2,3] have shown that there has been an increment in the number of students that plagiarize source codes. For example, in 1995, according to Rosales et al. [2], less than 2% of the students' code passed through the software `pk2` where found guilty of plagiarism, while in 2006 this percentage grows to an almost 10%.

This problem has became an important topic such that researchers started to define the problem more formally. For instance, in 1987 Faidhi and Robinson

* This work was partially supported by CONACyT México Project Grant CB-2010/153315, and SEP-PROMEP UAM-PTC-380/48510349. We also thank to SNI-CONACyT for their support.

[4] proposed a seven level hierarchy that aimed at representing most of the program's modifications used by students when they plagiarize code. As a consequence, many approaches try to identify code plagiarized based on these levels of complexity, that include: no changes at all (level 0), modification in comments, identifiers, variable position, procedures combination, program statements and logical control, from level 1 to 6 respectively.

In addition, over the past years, several methods have been dealing with detecting source code reuse focusing on two main aspects: structural and superficial[1]. However, in order to detect modifications on structural aspects, a deeper and strict analysis is required and it is often imply to have a complete knowledge of some particular programming language.

It is important to mention that similar to plagiarism over text documents, plagiarist of source code also apply several techniques in order to obfuscate or camouflage the plagiarized sections. Accordingly, it is very difficult to focus on all the possible types of obfuscation and incorporate them into a single method. Nevertheless, as in text documents (written in natural language), source code has structure and meaning, and also has inherently (to some extend) the particularities of the original author's written style. In the knowledge of these similarities between text documents and programming languages, we propose the use of natural language processing techniques to present a methodology that incorporate several views (representations) of the source code, which aim at providing more elements to accurately identify source code reuse. Specifically, this paper proposes and analyzes different representations of a source code, namely: n-grams of characters, data types, and identifiers' names. Our intuitive idea indicates that by means of considering different aspects from a source code (including those evaluated here), it will be possible to capture some of the most common practices performed by the plagiarist when they are camouflaging plagiarized sections.

The rest of this paper is organize as follows. In Section 2 we report related methods and its way to tackle the problem. Then, we present our proposed representations in Section 3. Section 4 shows the experimental settings to test our methodology; also, in this section we present the results obtained over a set of source codes written in the programing language C. Finally, in Section 5, we depict some conclusions and future work lines.

2 Related Work

The plagiarism detection problem has been tackled through several approaches, mainly NLP techniques adapted to the specific content of source code. One such works, take into account a trace that remains after a copy of source code: whitespace patterns [5]. In this work the file is converted to a pattern, namely 'whitespace' format: replacing any visible character by `X`, any whitespace by `S`, and leaving newlines as they appear. The method calculates a similarity

[1] From the more external and internal levels of the hierarchy proposed by [4], respectively

index based on the longest common substring (LCS) of both patterns, LCS: $LCS/\max\{l_1, l_2\}$, where l_i is the size of pattern i. To perform their tests, authors used C source code from free software Apache and Linux Kernel. Using the distribution of similarity index for source code pairs, corroborate their hypothesis: pairs of similar code (different Kernel Linux version) have mean and standard deviation high, and pairs of different source code have mean and standard deviation low. However, there is not significative difference for similar and different source code.

Another very common approach is to determine the fingerprint, as the Moss tool, based on n-grams of words from the source code [6]. It is important to consider several features of source code, such as identifiers, number of lines, number of terms per line, number of hapax, etc. In [7] authors carried out an experiment composing a similarity measure which uses a particular weighting scheme aiming to combine the extracted features.

G. Cosma and M. Joy [8] performed a detailed analysis supported on LSA. They focused their work on three components: preprocessing (keeping or removing comments, keywords or program skeleton), weighting (combining diverse local or global weights) and the dimensionality of LSA. The experiments were based on information retrieval: given a query as a source code aimed to obtain the most similar source codes. Furthermore, they build four corpora; using for this purpose the tools Moss and Sherlock [9], and human judgements to clean results given by the tools. In order to achieve a good trade-off between components, they used the best preprocessing (remove comments) then, measured the performance based on MAP to select the best weighting (local frequency, normal global frequency, and normalization of document). Observing curves of MAP over dimensionality were determined 15 dimensions. Besides, they tuned the threshold analyzing the lowest positive matching, highest false matching, and separation (difference between lpm and hfm). Finally, the dimensionality was 30 for such corpora.

Notice that a common aspect among the previous works is that authors try to capture several aspects from source codes into one single/mixed representation (*i.e.,* a single view) in order to detect plagiarism. However, our hypothesis indicates that each aspect (*i.e.,* either structural or superficial elements) provides its own important information that can not be mixed with other aspect when representing source codes. Accordingly, we perform an analysis of two main aspects that we consider among the most discriminative for detecting source code reuse.

3 Proposed Source Code Representations

In this section we describe in detail our proposed representations. These representations are divided into two views (*i.e.,* lexical and structural). The goal is to to determine if it is possible to identify pairs of source code with high similarities, thus providing evidence of plagiarism.

The first view considers the lexical characteristics of the source code and tries to capture some superficial modifications. The second view considers some structural characteristics, *i.e.* signature's programming function from source codes, and it is subdivided into two types: *i)* similarities between data type of function's signatures, and *ii)* similarities between identifiers of function's signatures.

3.1 Lexical view: character 3-grams representation

The approach used in this representation was proposed by Flores Sáez [10]. The main idea was to represent a given source code into a bag of n-grams of characters, B_j, where all the blanks and line-breaks are deleted and the letters are changed into lowercase.

Comparison of two codes Given two codes, C_α and C_β, their bag of 3-grams is computed as we mentioned before; then, each code is represented as a vector $\mathbf{B}_\alpha$ and $\mathbf{B}_\beta$ according to the vector space model proposed by [11]. Finally, the similarity between a pair of source codes is computed using the cosine similarity, which is defined as follows:

$$sim_{lexical}(C_\alpha, C_\beta) = \cos(\theta) = \frac{\mathbf{B}_\alpha \cdot \mathbf{B}_\beta}{\|\mathbf{B}_\alpha\|\|\mathbf{B}_\beta\|} \quad (1)$$

3.2 Structural view: data types from the function's signature representation

As we mentioned before, the proposed structural view consists of two representations. The first representation considers the data types of the function's signatures[2]. The idea behind this representation is based on the intuition that plagiarists often are willing to change function's and argument's names, but not the data types of such elements. Thus, by means of using the data types of function's signatures we attempt to compare some elements that belong, to some extent, to the structure of the program.

Accordingly, first we represent each function's signature into a list of data types. For example, the following function's signature '`int sum(int numX, int numY)`' will be translated into '`int (int, int)`'.

In order to represent each source code, we need a vocabulary formed by all used data types within the source codes in revision, which is called the *data-type vocabulary*. Then, we form the respective vector for each function. For instance, consider that our vocabulary is formed by the elements '<`char, double, float, int`>', then, the representation of a function with the following signature '`int sum(int x, int y)`' will be `(0,0,0,2)`'. Notice that we are not considering the return type at this step.

[2] We will refer just as function to every programming function within a source code.

Below it is defined how two functions are compared. Later, we define a similarity equation that takes into account all the functions in a pair of source code.

Comparison of two functions. To calculate the similarity between two functions, we need to compare two parts of the function's signature: return data type and arguments data types. We measured the importance of each part independently and then we merge them.

Given two functions, m^α and m^β from C_α and C_β respectively; and the return data type of those functions (*i.e.,* m_r^α and m_r^β), we can compute their similarity as Equation 2 states.

$$sim_r(m_r^\alpha, m_r^\beta) = \begin{cases} 1 \; if \; m_r^\alpha = m_r^\beta \\ 0 \; otherwise \end{cases} \tag{2}$$

For the same two functions, m^α and m^β, the data types of their arguments are represented as vectors $\mathbf{m}^\alpha = [a_1^{m^\alpha}, a_2^{m^\alpha}, ..., a_k^{m^\alpha}]$ and $\mathbf{m}^\beta = [a_1^{m^\beta}, a_2^{m^\beta}, ..., a_k^{m^\beta}]$. Where each positions in the vector represents a data type from the vocabulary of data-type, and the value for each element is the frequency of that type in the function's signature. Hence, the similarity of the arguments' data types is calculated as defined in Equation 3.

$$sim_a(\mathbf{m}^\alpha, \mathbf{m}^\beta) = \frac{\sum_{i=0}^{k} \min(a_i^{m^\alpha}, a_i^{m^\beta})}{\sum_{i=0}^{n} \max(a_i^{m^\alpha}, a_i^{m^\beta})} \tag{3}$$

Once we have the similarities from the return data-type and the arguments' data-type we merge them into a linear combination to find the similarity between m^α and m^β; as in Equation 4.

$$sim(m^\alpha, m^\beta) = \sigma * sim_r(m_r^\alpha, m_r^\beta) + (1 - \sigma) * sim_a(\mathbf{m}^\alpha, \mathbf{m}^\beta) \tag{4}$$

where σ is a scalar that weights the importance of each term and it satisfies that $0 \leq \sigma \leq 1$. For our performed experiments, presented in a following section, we established $\sigma = 0.5$ so both parts are considered equally important.

So far, we have described how to compute the similarities among functions considering both, their data types as well as their arguments names. Next, we describe how we measure the final similarity of two source codes considering all previous computed information.

Comparison of two codes Given two codes, C_α and C_β, we compute a function-similarity matrix $\mathbf{M}_{\alpha,\beta}^{type}$, where all functions in C_α are compare against all functions in C_β. Thus, the final values of similarity between two codes are defined as in Equation 5.

$$sim_{DataTypes}(C_\alpha, C_\beta) = f(\mathbf{M}_{\alpha,\beta}^{type}) \tag{5}$$

where $f(x)$ represents either the maximum value contained in the matrix, or the average value among all values from the matrix. Therefore, if we select the maximum value it implies that we are being more strict for determining when a pair of source codes are in fact plagiarized; and if we select an average value it indicates that we are being less strict, in other words, less rigorous for determining when two source codes are plagiarized. For example, if it does not matter how many functions are present in both source codes, and if just the occurrence of one exact match (*i.e.,* an equal function) is enough for labeling two source codes as plagiarized, then we must employ the maximum value from the function-similarity matrix. However, in a more relaxed criterion, we can use the average similarity among functions and use this value as the final similarity value between a pair of source codes.

3.3 Structural view: names from the function's signatures representation

As a complement for the previous representation, this one considers the structure by using the names of the functions as well as the name of the arguments. Our intuition is that some plagiarists might try to obfuscate the copied elements by means of changing data types, but not the variable's names.

This representation concatenate the name of the function's name with the name of the arguments. First, we convert every character to its lowercase form and we remove white spaces (if present). Thus, the function '`int sum(int x, int y)`' will be represented as '`sumxy`'. After that, we extracted all the 3-grams of characters and form a bag of 3-grams.

Once we have computed the bag of n-grams, we can compute how similar are a pair of functions. Next, we explain how we compare two functions, and then we define how several functions extracted from a pair of source codes are compare in order to determine the similarity between them.

Comparison of two functions Given two functions, m^α and m^β from C_α and C_β respectively; and their corresponding bag of 3-grams b^{m^α} and b^{m^β}, we compute the similarity of this two functions using the Jaccard coefficient as follows:

$$sim(m^\alpha, m^\beta) = \frac{b^{m^\alpha} \cap b^{m^\beta}}{b^{m^\alpha} \cup b^{m^\beta}} \tag{6}$$

Comparison of two codes Similarly to the previous approach, every function in C_α is compared against every function in C_β. From this comparison we obtain a name-similarity matrix $\mathbf{M}^{names}_{\alpha,\beta}$. Hence, the final similarity values of C_α and C_β is defined as established in Equation 7.

$$sim_{Names}(C_\alpha, C_\beta) = f(\mathbf{M}^{names}_{\alpha,\beta}) \tag{7}$$

where $f(x)$ can be set to the maximum value in the matrix, or the average value from the matrix. The meaning of such selection indicates the level of strength in the criteria to determine plagiarism between two source codes.

4 Experimental Results

The experiments performed aim at analyzing the pertinence of each of the proposed representations when determining the similarity between a pair of source codes. As we mentioned in previous sections, three representations were proposed: *i)* a lexical view, described in Section 3.1; and *ii)* a structural view that is composed by two other representations, namely the data-type representation (section 3.2) and the function and arguments' names (section 3.3).

In order to conduct an analysis of the proposed representations we evaluated our proposed approaches using a subset from the training collection of the competition of Detection of Source Code Re-use (SoCO 2014)[3]. This subset consists of 79 source codes in C language, where 26 pairs were tagged as cases of plagiarism by human experts. It is worth to mention that the relevance judgments do not indicate the direction of the plagiarism , *i.e.,* they do not indicate which source code is the original and which is the copy, hence, we do not detect such phenomena.

For each representation we compute the similarities values of each source code files given in the collection. Then, we measure the performance of each proposed representation by means of establishing a manual threshold for considering when two codes are plagiarized (re-used). That threshold was set from 10 to 90 percent of similarity. For each threshold we evaluated the precision, recall and F-measure (based on the relevance judgments). Notice that, at this stage of our investigation we still not identify cases of source code re-use, rather we want to analyze the pertinence of each of the proposed representations for finding similarities within a pair of codes.

4.1 Experiment 1. Lexical view

For this experiment, we use the implementation done by Picazo et al.[12] of the method proposed by Flores [10] (See section 3.1). The results are presented in Figure 1, where we can see values of precision, recall and f-measure for different threshold's similarities values.

As we can see in the figure, it is clear that when the threshold is very relaxed, the recall (i.e. the number of source code pairs that under the lexical representation are labelled as plagiarize and they actually are) is very good. On the contrary, the precision is very poor since the method identify too many pairs of source code as similar. The opposite situation is presented when we are very strict in the decision of how much similar two codes must be to labeled as plagiarize. However, we found that a good compromise is reached at 80% of similarity, when the f-measure is 0.56.

[3] http://users.dsic.upv.es/grupos/nle/soco/

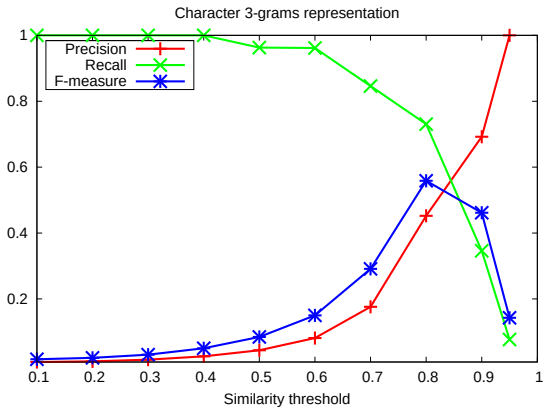

Fig. 1. Lexical view. Best result is obtained with the 80% of similarity between two methods

4.2 Experiment 2. Structural view

In this experiment we evaluated the two approaches: Comparison of Function Signature's Data Types and Comparison of Function Signature's Names.

In both cases, from Equations 5 y 7, we implemented function $f(x)$ as the maximum values of similarity among all the compared functions, and the average of similarities of the methods contained in the files of source codes. With this decision we attempt to compare the performance of this two polices. In this implementation we considered all the functions within the source code except for the `main()` function. The results for the *Comparison of Function Signature's Data Types* approach are presented in Figure 2 and Figure 3.

From the graphs in Figures 2 and 3, we can notice the considerable low performance obtained in comparison with the values obtained with the lexical view. However, this decrement in f-measure is also understandable, since the information from where we computed the similarities is a small part of the whole file of source code, that is function's signature only. Another important remark is that, as we expected, the more strict we are in the policy to determine a pair of source code as plagiarized the similarity among this pair has to be greater (as in the case of using the maximum similarity from the matrix of method similarities computed). The best results are obtained when the similarity is 90% (0.14 of f-measure) when considering the maximum, and 50% (0.16 of f-measure) when considering the average.

An important different between the results are the behavior of precision-recall values. When using the average we can see a expected precision-recall relation (*i.e.*, precision improving while recall decreasing) and we can get a good compromise to find the best configuration. We can not see this pattern when using the maximum.

Regarding to the second approach, *Comparison of Function Signature's Names*,

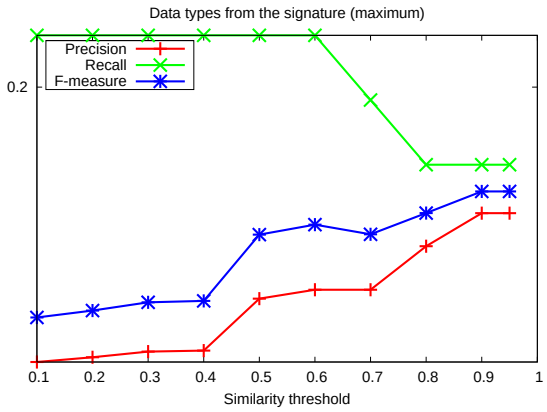

Fig. 2. Structural view: data type of function's signatures using the maximum value of similarities between functions. Best result is obtained with more than 90% of similarity between two methods

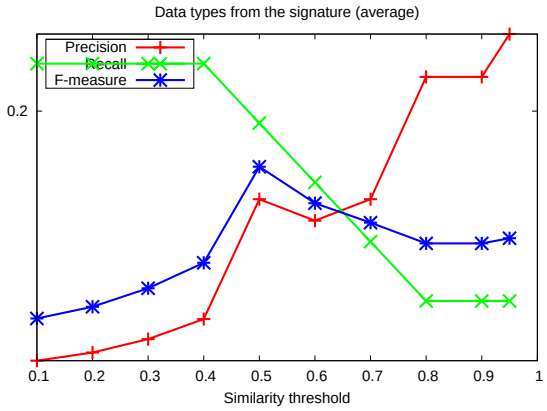

Fig. 3. Structural view: data type of function's signatures using the average value of similarities between functions. Best result is obtained with 50% of similarity between two methods

the graphs in Figure 4 and Figure 5 show that the best F-measure, *i.e.*, 0.26 and 0.22 was obtained when the similarity between codes was 40% and 20%, respectively.

Notice that even when the F-measure is poor for both representation in the structural view, they gives complementary information. While the recall is significant better when the data-type are considered, the precision is much better when the names of the function's signature are taking into account.

Another important remark is that given the results obtained with the lexical view and the fact that this representation uses more information, it might serve

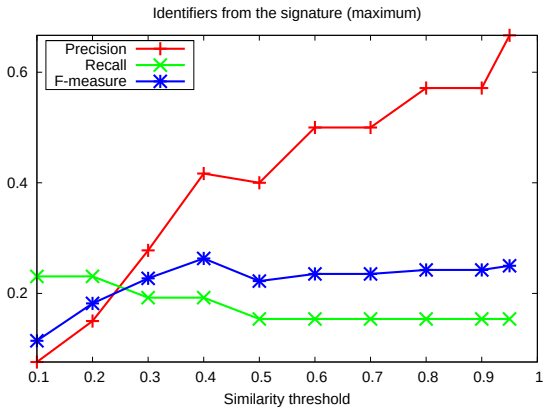

Fig. 4. Structural view: identifiers of function's signatures using the maximum value of similarities between functions. Best result is obtained with 40% of similarity between two methods

as the base approach to consider (or the most important similarity) to identify source code reuse.

More experiments must be done to analyze a good combination of this similarities to build a general method that identify source code plagiarism. Also, this methodology can be extended no only to C language, but to any other language, since our proposed views do not depend of any particularity of a programming language.

5 Conclusions and Future Work

We presented several representations for source code that highlight different views of a source code. In particular, we presented two views: lexical and structural. From the lexical view, we used an implementation of the Flores's method. For the structural view, we proposed two similarities that takes into account function's signatures within the source code. The conducted experiments help us to see that the information from this different approaches can be complementary. Also, we see that the lexical view gives the best similarity since uses the entire source code file. The proposed method can be extended to other views as well to other programming languages.

The future immediate work is to combine different views to determine if a pair of source code is or not been plagiarize. The first idea is learn the most important view, this can be done by looking at the view with the best F-measure. Another idea is to learn this weight in an automatic fashion, using a learning algorithm.

The experiment were performed with a collection of 79 source code in C programing language, but we believe that we can translate the views to another languages, for example Java, without any adjustments to the presented views.

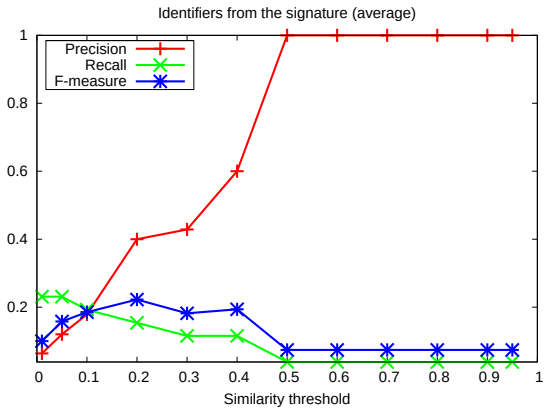

Fig. 5. Structural view: identifiers of function's signatures using the average value of similarities between functions. Best result is obtained with more than 20% of similarity between two methods

References

1. Shaw, M., Jones, A., Knueven, P., McDermott, J., Miller, P., Notkin, D.: Cheating policy in a computer science department. SIGCSE Bull. **12** (1980) 72–76
2. Rosales, F., Dopico, A.G., Rodríguez, S., Pedraza, J.L., Méndez, R., Nieto, M.: Detection of plagiarism in programming assignments. IEEE Trans. Education **51** (2008) 174–183
3. Zhang, D., Joy, M., Cosma, G., Boyatt, R., Sinclair, J., Yau, J.: Source-code plagiarism in universities: a comparative study of student perspectives in china and the uk. Assessment & Evaluation in Higher Education **39** (2014) 743–758
4. Faidhi, J.A.W., Robinson, S.K.: An empirical approach for detecting program similarity and plagiarism within a university programming environment. Comput. Educ. **11** (1987) 11–19
5. Baer, N., Zeidman, R.: Measuring whitespace pattern sequence as an indication of plagiarism. Journal of Software Engineering and Applications **5** (2012) 249–254
6. Aiken, A.: Moss, a system for detecting software plagiarism (1994)
7. Narayanan, S., Simi, S.: Source code plagiarism detection and performance analysis using fingerprint based distance measure method. In: Computer Science Education (ICCSE), 2012 7th International Conference on. (2012) 1065–1068
8. Cosma, G., Joy, M.: Evaluating the performance of lsa for source-code plagiarism detection. Informatica **36** (2013) 409–424
9. Joy, M., Luck, M.: Plagiarism in programming assignments. Education, IEEE Transactions on **42** (1999) 129–133
10. Flores, E.: Reutilización de código fuente entre lenguajes de programación. Master's thesis, Universidad Politécnica de Valencia, Valencia, España (2012)
11. Baeza-Yates, R.A., Ribeiro-Neto, B.: Modern Information Retrieval. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA (1999)
12. Picazo-Alvarez, R., Villatoro-Tello, E., Luna-Ramírez, W.A., Jaimez-Gonzáles, C.R.: Herramienta de apoyo en la detección de reutilización de código fuente. Journal of Research in Computing Science **73** (2014) 45–57

Algunas semánticas lógicas para reconocer implicación textual

José-de-Jesús Lavalle-Martínez[1,2] *, Manuel Montes-y-Gómez[1], Héctor Jiménez-Salazar[3], Luis Villaseñor-Pineda[1], David Pinto-Avendaño[2]

[1] Coordinación de Ciencias Computacionales
Instituto Nacional de Astrofísica, óptica y Electrónica
[2] Facultad de Ciencias de la Computación
Benemérita Universidad Autónoma de Puebla
[3] Departamento de Tecnologías de la Información
Universidad Autónoma Metropolitana-Cuajimalpa

Resumen En el presente trabajo se presentan algunos métodos lógicos que en general se han utilizado para el procesamiento de lenguaje natural y últimamente en la tarea para reconocer implicación textual, estos son: Teoría de Representación de Discurso, Gramáticas Categoriales, Lógica de Predicados Dinámica y Lógica Natural. El interés principal es reunir, en un solo artículo en español, los principales formalismos lógicos que se han utilizado para atender aspectos semánticos propios del procesamiento de lenguaje natural.

Palabras clave: Teoría de Representación de Discurso, Gramáticas Categoriales, Lógica de Predicados Dinámica, Lógica Natural, Semántica de Lenguaje Natural.

1. Introducción

Si bien en el *Procesamiento de Lenguaje Natural* (NLP) el enfoque lógico no ha sido preferido, debido principalmente al mejor desempeño que han mostrado otros enfoques [1], en la tarea de *Implicación Textual* se ha vuelto a retomar [2]. En primer lugar porque la lógica es quien formaliza la noción de implicación; en segundo lugar porque, gracias a la especificación y verificación formal de software y hardware, se han desarrollado herramientas [3] tanto teóricas como prácticas que han permitido razonar sobre sistemas de tamaño industrial; en tercer lugar dado que en la tarea de implicación textual ningún enfoque ha demostrado supremacía sobre los demás [4,5].

De acuerdo con el Portal de Implicación Textual de la ACL, la *Implicación Textual* es una relación direccional entre dos fragmentos de texto. La relación se cumple siempre que la veracidad del segundo fragmento de texto se sigue de la veracidad del primer fragmento de texto. En el marco de la Implicación Textual

* Este trabajo ha sido apoyado por PROMEP mediante el convenio PROMEP/103.5/13/5618 y número de becario BUAP-803.

al texto implicante se le llama *texto* (t) y al texto implicado se le llama *hipótesis* (h).

Dos son los principales enfoques lógicos que se han usado para el reconocimiento de implicación textual.

Por un lado está el de Bos y Markert [6] quienes utilizan Gramáticas Categoriales (CG), Teoría de Representación de Discurso (DRT), Demostradores Automáticos de Teoremas y Chequeadores de Modelos para reconocer cuando hay implicación textual entre el par (t, h).

Por otro lado está el trabajo de MacCartney y Manning [7], ellos utilizan Lógica Natural para el reconocimiento de la implicación textual.

Ambos métodos tienen ventajas y desventajas: Bos y Markert logran buena precisión pero mala cobertura; MacCartney y Manning, por el contrario, logran buena cobertura pero mala precisión.

Desde un punto de vista más técnico los diversos métodos usados tienen también ventajas y desventajas: la DRT tiene como ventaja que fue creada para solucionar los problemas de anáfora que las gramáticas de Montague no podían manejar [8], pero tiene el inconveniente de que su salida es una fórmula de la lógica de primer orden, por tanto razonar sobre su validez es indecidible en general.

Las CG [9] tienen la ventaja de que sintaxis y semántica están íntimamente ligadas y que la semántica se encuentra evaluando una expresión del cálculo lambda, su principal desventaja es que no pueden manejar el fenómeno de la anáfora.

La Lógica de Predicados Dinámica (DPL) [10] fue creada inicialmente para razonar sobre los cambios de estado que tiene un programa imperativo, posteriormente se ha utilizado para tratar la anáfora.

Por último, la Lógica Natural [11] tiene la desventaja de que no puede tratar con el fenómeno de la anáfora creemos que por ello es baja en precisión, su ventaja es que es una teoría decidible.

En este trabajo se presentan algunas teorías lógicas que se han utilizado en el procesamiento del lenguaje natural en general y, particularmente, en el reconocimiento de implicación textual.

2. Teoría de Representación de Discurso

La Teoría de Representación de Discurso [12,8], es una de las teorías para semántica dinámica. El interés principal de estas teorías es tomar en cuenta la dependencia del contexto que tiene el significado. Es una característica ubicua de los lenguajes naturales que lo que se expresa es interpretable sólo cuando el intérprete toma en cuenta el contexto en el que se hizo, el significado de lo expresado depende del contexto.

Aún más, la interacción entre el contexto y lo expresado es recíproco. Cada expresión contribuye (vía la interpretación que se le da) al contexto en el que se hace. Modifica el contexto en un nuevo contexto, en el que esta contribución

se refleja, es este nuevo contexto el que influirá en la interpretación de cualqueir cosa que se exprese posteriormente.

2.1. Definición

Una Estructura de Representación de Discurso [13] se obtiene a partir de un conjunto de constantes de relación, un conjunto de constantes individuales y un conjunto infinito de variables individuales (a las constantes y variables individuales se les llama *referentes de discurso*). Las condiciones y las cajas se construyen a partir de estos conjuntos y de las siguientes cláusulas:

- Si R es una relación constante de aridad n y $\delta_1, \ldots, \delta_n$ son referentes de discurso entonces $R(\delta_1, \ldots, \delta_n)$ es una condición;
- Si δ_1 y δ_2 son referentes de discurso entonces δ_1 **is** δ_2 es una condición.
- Si K_1 y K_2 son cajas, entonces **not** K_1, K_1 **or** K_2 y $K_1 \Rightarrow K_2$ son condiciones;
- Si $\gamma_1, \ldots, \gamma_m$ son condiciones ($m \geq 0$) y $x_1, \ldots, x_n$ son variables ($n \geq 0$) entonces $[x_1 \ldots x_n | \gamma_1, \ldots, \gamma_m]$ es una caja.
- Si K_1 y K_2 son cajas entonces $K_1; K_2$ es una caja.

El lenguaje generado por las cláusulas anteriores se interpreta como modelos ordinarios de primer orden. Los modelos se definen como pares $< D, I >$, donde D es un conjunto arbitrario no vacío e I es una función que tiene como dominio el conjunto de constantes tal que $I(c) \in D$ para cada constante individual c e $I(R) \subseteq D^n$ para cada constante de relación R de aridad n.

Una *asignación* para tal modelo de primer orden $M =< D, I >$ es una función del conjunto de variables de referentes de discurso al dominio D. Escribimos $a[x_1, \ldots, x_n]a'$ como una abreviación de "las asignaciones a y a' difieren a lo más en sus valores para $x_1, \ldots, x_n$. Como es usual, definimos $\|\delta\|^{M,a}$ como $a(\delta)$ si δ es una variable y como $I(\delta)$ si δ es una constante.

Las cláusulas siguientes definen el valor semántico $\|\gamma\|^M$ de una condición γ en un modelo M como un conjunto de asignaciones, el valor semántico $\|K\|^M$ de una caja K en M se define como una relación binaria entre asignaciones (el super índice M se omitirá).

- $\|R(\delta_1, \ldots, \delta_n)\| = \{a \mid \langle \|\delta_1\|^a, \ldots, \|\delta_n\|^a \rangle \in I(R)\}$
- $\|\delta_1$ **is** $\delta_2\| = \{a \mid \|\delta_1\|^a = \|\delta_2\|^a\}$
- $\|$**not** $K\| = \{a \mid \neg \exists a' \langle a, a' \rangle \in \|K\|\}$
- $\|K_1$ **or** $K_2\| = \{a \mid \exists a' (\langle a, a' \rangle \in K_1 \vee \langle a, a' \rangle \in K_2)\}$
- $\|K_1 \Rightarrow K_2\| = \{a \mid \forall a' (\langle a, a' \rangle \in K_1 \rightarrow \exists a'' \langle a', a'' \rangle \in K_2)\}$
- $\|[x_1 \ldots x_n | \gamma_1, \ldots, \gamma_m]\| = \{\langle a, a' \rangle \mid a[x_1, \ldots, x_n]a' \wedge a' \in \|\gamma_1\| \cap \ldots \cap \|\gamma_m\|\}$
- $\|K_1; K_2\| = \{\langle a, a' \rangle \mid \exists a'' (\langle a, a'' \rangle \in \|K_1\| \wedge \langle a'', a' \rangle \in \|K_2\|)\}$

Una caja K es *verdadera* en un modelo M bajo una asignación a si y sólo si existe alguna asignación a' tal que $\langle a, a' \rangle \in \|K\|^M$; una condición γ es verdadera en M bajo a si y sólo si $a \in \|\gamma\|^M$.

2.2. Ejemplo

Dado el siguiente texto en inglés "*A man adores a woman. She abhors him.*" La caja cerrada que le corresponde a la sentencia "*A man adores a woman.*" es (1), la única caja razonable que se puede asociar con la sentencia abierta "*She abhors him.*" es la caja abierta (2) la cual es verdadera bajo una asignación a si y sólo si la condición $x_2\ abhors\ x_1$ es verdadera bajo a.

Los pronombres anafóricos "*she*" y "*him*" obtienen cualquier valor que la asignación de entrada asocie con los referentes de discurso x_2 y x_1. La caja (2) se puede interpretar como una prueba: dada cualquier asignación de entrada a, prueba si $a(x_2)\ abhors\ a(x_1)$, si es así regresa a a como salida, si no la prueba falla y ninguna salida se regresa.

$$[x_1\ x_2 | man\ x_1, woman\ x_2, x_1\ adores\ x_2] \tag{1}$$

$$[|x_2\ abhors\ x_1] \tag{2}$$

De tal manera que la caja correspondiente al texto "*A man adores a woman. She abhors him.*" es:

$$[x_1\ x_2 | man\ x_1, woman\ x_2, x_1\ adores\ x_2]; [|x_2\ abhors\ x_1]$$

3. Gramáticas Categoriales

Las gramáticas categoriales [9] son una forma de gramáticas lexicalizadas, en las que la aplicación de las reglas sintácticas está condicionada completamente por el tipo sintáctico, o la *categoría* de sus entradas.

Las categorías identifican sus constituyentes como *categorías primitivas* o *funciones*. Las categorías primitivas, tales como N, NP, PP, S, etc, pueden enriquecerse con características tales como número, caso, inflección y similares.

Las funciones (tales como los verbos) portan categorías que identifican el tipo de su resultado y el de sus argumentos/complementos (ambos pueden ser a su vez funciones o categorías primitivas). Las categorías función también definen el orden en el que los argumentos se deben combinar y si ocurren a la derecha o la izquierda del funtor.

Cada categoría sintáctica se asocia con una forma lógica cuyo tipo semántico está determinado completamente por la categoría sintáctica.

En gramáticas categoriales la información sintáctica, de la clase que se puede capturar para el Inglés mediante reglas de producción como (3), (4) y (5), se transfiere a entradas léxicas como (6):

$$S \rightarrow NP \quad VP \tag{3}$$

$$VP \rightarrow TV \quad NP \tag{4}$$

$$TV \rightarrow \{proved, finds, \ldots\} \tag{5}$$

$$proved := (S \backslash NP)/NP \tag{6}$$

Esta categoría sintáctica identifica al verbo transitivo como una función, especifica el tipo y direccionalidad de sus argumentos y el tipo de su resultado. Aquí se usa la notación del "resultado más izquierdo" en la que una función que *combina a la derecha* sobre un dominio β en un rango α se escribe α/β, el correspondiente funtor que *combina a la izquierda* se escribe $\alpha\backslash\beta$, donde α y β también pueden ser categorías función.

3.1. Definición del Lenguaje Categorial

En Gramáticas Categoriales los *tipos* (también llamados *categorías*) se definen como sigue:

$$L ::= P|(L/L)|(L\backslash L)$$

donde P es el conjunto de tipos primitivos, los cuales son llamados tipos atómicos o categorías básicas, los tipos más usuales son S (para sentencias), NP (para frases nominales), y puede incluir PP (para frases preposicionales), INF (para infinitivos), etc.

Es usual decir que una fórmula de tipo X/Y o $X\backslash Y$ es un funtor, siendo la fórmula Y su argumento y la fórmula X su resultado.

3.2. Reglas de aplicación funcional, categorías sintácticas

Para permitir que funtores como (6) combinen con sus argumentos necesitamos reglas combinatorias, de éstas las dos más simples son las reglas de aplicación funcional siguientes:

$$\frac{X/Y \quad Y}{X} > \qquad\qquad \frac{Y \quad X\backslash Y}{X} <$$

3.3. Ejemplo de categorías sintácticas

Dadas la oración "*Marcel proved completeness*", y las siguientes entradas léxicas $Marcel := NP$, $proved := (S\backslash NP)/NP$ y $completeness := NP$, se procede como sigue: dado que $Marcel$ y *completeness* son elementos del tipo primitivo NP, no se les puede aplicar alguna de las reglas de aplicación funcional, pero notamos que *proved* tiene el tipo funcional $(S\backslash NP)/NP$ que es el tipo de los verbos transitivos.

Así, para combinar el tipo funcional $(S\backslash NP)/NP$ necesitamos a la derecha de *proved* un elemenot del tipo NP en este caso *completeness*, usando la regla $>$ obtenemos el tipo funcional $(S\backslash NP)$. Para combinar este tipo mediante la regla $<$ necesitamos un elemento de NP a la izquierda de *proved*, en este caso tenemos a $Marcel$, aplicando dicha regla obtenemos el tipo S que es el tipo de las oraciones sintácticamente correctas.

El razonamiento anterior se puede expresar gráficamente mediante un árbol de derivación como el siguiente (al final de las líneas horizontales se marca que regla se usó para la derivación, se marca con Lex cuando se usa una entrada léxica).

$$\dfrac{\dfrac{Marcel}{NP}\ \text{Lex} \quad \dfrac{\dfrac{proved}{(S\backslash NP)/NP}\ \text{Lex} \quad \dfrac{completeness}{NP}\ \text{Lex}}{S\backslash NP}\ >}{S}\ <$$

3.4. Reglas de Aplicación Funcional, Tipos Semánticos

Se puede considerar que las categorías codifican el tipo semántico de su traducción. La traducción se puede hacer explícita asociando una forma lógica con la categoría sintáctica completa a través del operador :, se asume que éste tiene menor precedencia que los operadores categoriales / y \.

Por supuesto se deben de expandir las reglas de aplicación funcional de acuerdo a los tipos semánticos, recordando que los operadores / y \ definen categorías funcionales el resultado de aplicar las reglas > y <, cuando ya se tienen los tipos semánticos, debe ser la evaluación de una función, como se indica a continuación.

$$\dfrac{X/Y : f \quad Y : a}{X : fa}\ > \qquad\qquad \dfrac{Y : a \quad X\backslash Y : f}{X : fa}\ <$$

3.5. Ejemplo de tipos semánticos

Enriqueciendo con tipos semánticos las entradas léxicas del ejemplo "*Marcel proved completeness*" se tiene

$$\begin{aligned} Marcel &:= NP : marcel' \\ proved &:= (S\backslash NP)/NP : \lambda x.\lambda y.prove' xy \\ completeness &:= NP : completeness' \end{aligned}$$

Como se observa las categorías funcionales se enriquecen semánticamente mediante cálculo lambda y las categorías primitivas mediante una constante que se forma primando la entrada léxica, pero sin considerar su tipo sintáctico.

Aplicado la regla > para tipos semánticos a $(S\backslash NP)/NP : \lambda x.\lambda y.prove' xy$ y $NP : completeness'$ obtenemos $S\backslash NP : \lambda y.prove' completeness' y$. Si ahora le aplicamos la regla < a dicho resultado y a $NP : marcel'$, se obtiene $S : prove' completeness' marcel'$. Nuevamente este razonamiento se puede expresar mediante el siguiente árbol de derivación semántica.

$$\dfrac{\dfrac{Marcel}{NP : marcel'}\ \text{Lex} \quad \dfrac{\dfrac{proved}{(S\backslash NP)/NP : \lambda x.\lambda y.prove' xy}\ \text{Lex} \quad \dfrac{completeness}{NP : completeness'}\ \text{Lex}}{S\backslash NP : \lambda y.prove' completeness' y}\ >}{S : prove' completeness' marcel'}\ <$$

4. Lógica Dinámica de Predicados

En DPL [10] la dinámica se refiere a la información sobre cosas que se van introduciendo en un discurso y que sirven como posibles antecedentes para pronombres anafóricos subsecuentes. Como es usual en lingüística, las frases nominales (frases nominales indefinidas y pronombres) se asocian con índices o variables, para poder indicar casos de correferencia y ligado. La información relevante es información sobre los posibles valores de dichas variables, las cuales pueden ser cambiadas y actualizadas conforme avanza el discurso.

4.1. Definición

Se define $[\![\phi]\!]_M$, la interpretación de una fórmula ϕ de la lógica de predicados de primer orden relativa a un modelo ordinario M de la lógica de predicados de primer orden, como un conjunto de pares de asignaciones a variables, asignaciones entrada/posible-salida $\langle g, h\rangle$. La idea es que un par $\langle g, h\rangle$ está en la interpretación de ϕ relativa a M si y sólo si sobre la asignación de entrada g la fórmula ϕ puede interpretarse exitosamente y da como posible-salida la asignación h. Si no hacen falta se omitirán las referencias a M.

Un lenguaje L para DPL es el de la lógica de predicados de primer orden ordinaria, basado en un conjunto C de constantes individuales c y conjuntos R^n de constantes relacionales R de aridad n y un conjunto numerable de variables V. El conjunto de términos $T = C \cup V$ consiste de las constantes individuales y las variables del lenguaje, las fórmula atómicas $Rt_1 \dots t_n$ se componen de predicados R de aridad n y una secuencia de n términos $t_1, \dots, t_n$, también pueden ser de la forma $t_i = t_j$, enunciando la identidad de los valores de los términos t_i y t_j. Las fórmulas se construyen a partir de las fórmulas atómicas usando negación ($\neg$), cuantificadores existencial y universal ($\exists x, \forall y$), conjunción ($\wedge$), disyunción ($\vee$) e implicación ($\rightarrow$).

Un modelo $M = \langle D, V\rangle$ es un modelo usual de primer orden con un dominio de individuos D y una función de interpretación V para las constantes individuales y relacionales de nuestro lenguaje. La función V asigna un individuo $V(c) \in D$ a las constantes individuales de L y un conjunto de $n-$tuplas de individuos $V(R^n) \subseteq D^n$ a sus constantes relacionales de aridad n. En la interpretación de DPL también usamos asignaciones variables f, g, h, k, l las cuales asignan individuos $f(x) \in D$ a las variables $x \in V$, así que son funciones de V a D. La interpretación $[t]_{M,g}$ de un término t en un modelo M y relativo a una asignación g es $V(t)$ si t es una constante individual y $g(t)$ si t es una variable.

Usamos $g[x/d]$ para la asignación variable h que es como g excepto que asigna d a x, así para toda $y \in V$, si $x \neq y$ entonces $g[x/d](y) = g(y)$ y si $x = y$ entonces $g[x/d](y) = d$. Escribimos $g[x]h$ si y sólo si $h = g[x/d]$ para algún individuo d y $g[X]h$ si y sólo si $X = \{x_1, \dots, x_n\}$ y existen $k_1, \dots, k_{n-1}$ tal que $g[x_1]k_1, \dots, k_{n-1}[x_n]h$. Usando estos dispositivos notacionales podemos enunciar la semántica de DPL como sigue:

- $[\![Rt_1 \dots t_n]\!]_M = \{\langle g, h\rangle \mid g = h \wedge \langle [t_1]_{M,g}, \dots, [t_n]_{M,g}\rangle \in V(R)\}$
- $[\![t_i = t_j]\!]_M = \{\langle g, h\rangle \mid g = h \wedge [t_i]_{M,g} = [t_j]_{M,g}\}$
- $[\![\neg\phi]\!]_M = \{\langle g, h\rangle \mid g = h \wedge$ para ningún $k : \langle g, k\rangle \in [\![\phi]\!]_M\}$
- $[\![\exists x\phi]\!]_M = \{\langle g, h\rangle \mid$ para algún $k : g[x]k \wedge \langle k, h\rangle \in [\![\phi]\!]_M\}$
- $[\![\forall x\phi]\!]_M = \{\langle g, h\rangle \mid g = h \wedge$ para todo $k :$ si $g[x]k$ entonces existe $h :$ $\langle k, h\rangle \in [\![\phi]\!]_M\}$
- $[\![\phi \wedge \psi]\!]_M = \{\langle g, h\rangle \mid$ para algún $k : \langle g, k\rangle \in [\![\phi]\!]_M \wedge \langle k, h\rangle\} \in [\![\psi]\!]_M\}$
- $[\![\phi \vee \psi]\!]_M = \{\langle g, h\rangle \mid g = h \wedge$ para algún $k : (\langle g, k\rangle \in [\![\phi]\!]_M \vee \langle g, k\rangle \in [\![\psi]\!]_M)\}$
- $[\![\phi \rightarrow \psi]\!]_M = \{\langle g, h\rangle \mid g = h \wedge$ para todo $k :$ si $\langle g, k\rangle \in [\![\phi]\!]_M$ entonces existe $h : \langle k, h\rangle \in [\![\psi]\!]_M\}$

4.2. Ejemplo de Interpretación Dinámica de un Discurso

Considere el siguiente texto en inglés "*A farmer owned a donkey. It was unhappy. It didn't have a tail.*", a éste le corresponde la siguiente fórmula de la lógica dinámica de predicados

$$\exists x(Fx \wedge \exists y(Dy \wedge Oxy)) \wedge (Uy \wedge \neg\exists z(Tz \wedge Hyz))$$

Relativo a la asignación de entrada g se tendrá un asignación de salida h si podemos encontrar asignaciones k y l tales que k es una salida posible al interpretar $\exists x(Fx \wedge \exists y(Dy \wedge Oxy))$ relativo a g, y l es una salida posible al interpretar Uy relativo a k, y h es una salida posible al interpretar $\neg\exists z(Tz \wedge Hyz)$ relativo a l.

Ya que la segunda fórmula es atómica y la tercera una negación, sabemos que en este caso $k = l$ y $l = h$. La asignación k (esto es: h) se obtiene de g al reiniciar el valor de x tal que $k(x) = h(x) \in I(F)$, y después reiniciando el valor de y tal que $k(y) = h(y) \in I(D)$ y $\langle h(x), h(y)\rangle \in I(O)$. Esto es, $h(x)$ es un granjero que posee un burro $h(y)$. Observe que para cualquier granjero f y burro d que f posee, existe una asignación correspondiente $h' : g[\{x, y\}]h'$ y tal que $h(x) = f$ y $h(y) = d$.

El segundo elemento de la conjunción primero prueba si y es infeliz, esto es, si $l(y) = k(y) = h(y) \in I(U)$. El tercer elemento de la conjunción, una negación, prueba si la asignación h no puede servir como entrada para satisfacer la fórmula interna $\exists z(Tz \wedge Hyz)$. Esta subfórmula se satisface relativa a h si y sólo si existe una asignación h' tal que $h[z]h'$ y $h'(z) \in I(T)$ y $\langle h'(y), h'(z)\rangle \in I(H)$, esto es, si y sólo si podemos cambiar la valuación h de z en cualquier cosa que tenga cola mediante $h(y)$.

La negación de la subfórmula prueba si no podemos cambiar la valuación de z en dicha manera. Juntando todo, $\langle g, h\rangle$ está en la interpretación de nuestro ejemplo si y sólo si $g[x, y]h$ y $h(x)$ es un granjero que posee un burro $h(y)$ el cual es infeliz y no tiene cola. Observe, una vez más, que para cualquier granjero f y burro sin cola e infeliz d al que f posee, existe una asignación correspondiente $h' : g[x, y]h'$ tal que $h(x) = f$ y $h(y) = d$.

5. Lógica Natural

Para desarrollar un enfoque cognitivo de razonamiento es prometedor factorizar el aspecto sintáctico (aquel que tiene que ver con coincidencia de patrones y estructuras sintácticas) del resto. Un candidato obvio para esta tarea es el llamado cálculo de monotonía o cálculo de la lógica natural [11]. Este cálculo tiene un lado sintáctico y un lado semántico.

El fundamento semántico del razonamiento monótono es una generalización de la noción de consecuencia lógica para tipos arbitrarios, lo cual se logra definiendo órdenes parciales $\Longrightarrow$ sobre todos los tipos (no sólo el tipo de oraciones, también el de frases verbales, predicados, adjetivos, cuantificadores, etc.).

En estos términos se puede definir que significa que una función del tipo α en el tipo β *preserve el orden* o *invierta el orden*. Las funciones que preservan el orden son las funciones f tales que si $x \Longrightarrow y$ entonces $f(x) \Longrightarrow f(y)$. Las funciones que invierten el orden son las funciones f tales que si $x \Longrightarrow y$ entonces $f(y) \Longrightarrow f(x)$.

El lado sintáctico del cálculo de monotonía tiene que ver con marcar la monotonía de las componentes de una estructura sintáctica. Sea S una estructura sintáctica, y sea A una componente de esa estructura. Suponga que A tiene tipo α y que S tiene tipo β. Considere la función sintáctica F que consiste en reemplazar la componente A por otra componente adecuada de tipo α. En otras palabras, considere la función $F = \lambda Y.S[Y/A]$. Entonces la contraparte semántica de F es una función f de tipo $\alpha \rightarrow \beta$. La validez y completitud de un cálculo de monotonía tiene que ver con la relación entre F y f.

Un algoritmo de marcación de monotonía es *válido* si se cumple lo siguiente: si A se marca + en S entonces la función que interpreta $\lambda Y.S[Y/A]$ preserva la monotonía, si A se marca - en S entonces la función que interpreta $\lambda Y.S[Y/A]$ invierte el orden.

Un algoritmo de marcación de monotonía es *completo* si se cumple lo siguiente: si la función que interpreta $\lambda Y.S[Y/A]$ preserva el orden entonces A se marca + en S, si la función que interpreta $\lambda Y.S[Y/A]$ invierte el orden entonces A se marca - en S.

5.1. Semántica de monotonía

Así como podemos decir que "*Gaia is smiling*" implica lógicamente "*Gaia is smilling or Gaia is crying*", nos gustaría decir que "*smilling*" implica lógicamente "*smilling or crying*", o que "*dancing*" implica lógicamente "*moving*", también que "*at least three*" implica lógicamente "*at least two*", etc.

"*Gaia is smilling*" es una oración, "*smilling*" es un predicado, "*at least three*" es un cuantificador. Sabemos que una sentencia implica a otra si siempre que la primera es verdadera la segunda también lo es. La manera obvia de trasladar esta noción a predicados es estipulando que un predicado implica a otro si se cumple que para todo sujeto la oración que se obtiene al combinar un sujeto con el primer predicado implica a la oración que se obtiene al combinar ese mismo sujeto con el segundo predicado. Similarmente para cuantificadores, para obtener una oración a partir de "*at least three*", se tiene que combinar el cuantificador con un sustantivo y un verbo. Ya que en verdad se cumple que para todo sustantivo N y verbo V "*at least three N V*" implica "*at least two N V*", podemos decir que "*at least three*" implica "*at least two*".

Iniciaremos con los tipos básicos t (valores veritativos, el tipo de las oraciones) y e (entidades, el tipo de los nombres propios). Los tipos complejos se definen por recursión como sigue:

1. e y t son tipos,
2. si α y β son tipos, entonces $\alpha \rightarrow \beta$ es un tipo.

La relación de implicación se define como sigue (usamos $E :: \alpha$ para expresar que "la expresión sintáctica E tiene tipo semántico α"):

1. Si $E, E' :: e$ entonces $I(E) \Longrightarrow I(E')$ si $I(E) = I(E')$,
2. Si $E, E' :: t$ entonces $I(E) \Longrightarrow I(E')$ si $I(E) \leq I(E')$,
3. Si $E, E' :: \alpha \rightarrow \beta$ entonces $I(E) \Longrightarrow I(E')$ si y sólo si para todo $x \in D_\alpha, I(E)(x) \Longrightarrow I(E')(x)$.

Aquí $I(E)$ denota la interpretación de E, y D_α se usa para el dominio de objetos de tipo α. Si $E :: \alpha$ entonces $I(E) \in D_\alpha$, es decir, la interpretación de E es un objeto en D_α, el dominio de objetos del tipo α.

5.2. Estructura general de las reglas para razonamiento monótono

Una función F que preserva monotonía se puede representar de la siguiente manera:

$$\frac{X \Longrightarrow Y}{F(X) \Longrightarrow F(Y)} \; F\uparrow$$

Aquí se asume que X y Y son expresiones del tipo lógico α que está ordenado parcialmente mediante $\Longrightarrow$, que $F(X)$ y $F(Y)$ son expresiones del tipo β que está ordenado parcialmente mediante $\Longrightarrow$, y que F es una función que preserva el orden de tipo $\alpha \Longrightarrow \beta$.

Una forma de leer la regla es como una explicación del hecho de que F preserva el orden (monótona creciente). Otra manera de leer la regla es como una regla de inferencia disparada por una función F que se sabe que preserva el orden. $F\uparrow$ expresa que F preserva el orden.

Si la función F invierte el orden tenemos la siguiente regla:

$$\frac{X \Longrightarrow Y}{F(Y) \Longrightarrow F(X)} \; F\downarrow$$

Nuevamente, hay varias maneras de leer esta regla. $F\downarrow$ expresa que F invierte el orden (o que es monótona decreciente).

Para apreciar la generalidad de la regla de monotonía, veamos algunos casos especiales. Si $X, Y, F(X)$ y $F(Y)$ tienen tipo t, entonces $\Longrightarrow$ es consecuencia lógica (o implicación lógica), y $F(X)$ y $F(Y)$ son oraciones, de esta manera obtenemos:

$$\frac{X \Longrightarrow Y \quad F(X)}{F(Y)} \; F\uparrow$$

Un ejemplo de aplicación de esta regla es: inferir de "*Mary dances implies Mary moves*" (cuando tomamos "*Mary dances*" como X y "*Mary moves*" como Y) y "*Mary dances gracefully*" (con "*gracefully*" como F) que "*Mary moves gracefully*".

Para el caso de inversión de orden tenemos:

$$\frac{X \Longrightarrow Y \quad F(Y)}{F(X)} \; F\downarrow$$

Con X y Y como antes y leyendo F como negación, tenemos el siguiente ejemplo de esta regla: inferir de "*Mary dances implies Mary moves*" y "*Mary does not move*" (con "*does not move*" como F) que "*Mary does not dance*".

En el caso de que X y Y son conjuntos (con tipo $e \to t$) y $F(X)$ y $F(y)$ son valores veritativos, F tiene tipo $(e \to t) \to t$ (el tipo de los cuantificadores), obtenemos:

$$\frac{Q(X) \quad X \subseteq Y}{Q(Y)} \; Q\uparrow$$

Como ejemplo, que X sea "*dancing*", que Y sea "*moving*" y que Q sea "*everyone*". Entonces la regla dice que podemos concluir de "*everyone is dancing*" y "*dancing involves moving*" que "*everyone is moving*".

$$\frac{Q(Y) \quad X \subseteq Y}{Q(X)} \; Q\downarrow$$

Para este caso, que X sea "*dancing*", que Y sea "*moving*" y que Q sea "*nobody*". Entonces la regla dice que podemos concluir de "*nobody is moving*" y "*dancing involves moving*" que "*nobody is dancing*".

En efecto, F puede tener más estructura interna, es decir, $F(X)$ puede tener la forma de un cuantificador generalizado binario $Quant(X, P)$ o $Quant(P, X)$. Lo cual nos da cuatro posibles reglas de monotonía para cuantificadores binarios. Ejemplos de cuantificadores binarios son: *all*, con propiedades de monotonía $(\downarrow, \uparrow)$; *some*, con propiedades $(\uparrow, \uparrow)$; *no*, con $(\downarrow, \downarrow)$ y *most*, con $(-, \uparrow)$.

$$\frac{Quant(X, P) \quad X \subseteq Y}{Quant(Y, P)} \; Quant(\uparrow, -)$$

Ejemplo, infiera de "*some philosophers are mortal*" y "*philosophers are humans*" que "*some humans are mortal*".

$$\frac{Quant(P, X) \quad X \subseteq Y}{Quant(P, Y)} \; Quant(-, \uparrow)$$

Ejemplo, infiera de "*most philosophers are human*" y "*humans are mortal*" que "*most philosophers are mortal*".

$$\frac{Quant(Y, P) \quad X \subseteq Y}{Quant(X, P)} \; Quant(\downarrow, -)$$

Ejemplo, infiera de "*all humans are mortal*" y "*philosophers are human*" que "*no philosophers are mortal*".

$$\frac{Quant(P, Y) \quad X \subseteq Y}{Quant(P, X)} \; Quant(-, \downarrow)$$

Ejemplo, infiera de "*no philosophers are mortal*" y "*humans are mortal*" que "*no philosophers are human*".

6. Conclusiones

Es importante ofrecer un mejor balance entre precisión y cobertura en el reconocimiento de implicación textual, para ello creemos que un camino por explorar es aquel que le dé a la lógica natural la posibilidad de manejar la anáfora. Es decir, fusionar la lógica natural con alguna forma de lógica dinámica.

Referencias

1. Jurafsky, D., Martin, J.H.: Speech and Language Processing, An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition. Second edn. Pearson Education Inc. (2009)
2. Bos, J.: Is there a place for logic in recognizing textual entailment? Linguistic Issues in Language Technology **9**(3) (July 2013)
3. Burch, J., Clarke, E.M., McMillan, K.L.: Symbolic model checking: 10^{20} states and beyond. Information and Computation **98**(2) (1992) 142–170
4. Koller, A., Pinkal*, M.: Semantic research in computational linguistics. In Maienborn, C., von Heusinger, K., Portner, P., eds.: Semantics: An International Handbook of Natural Language Meaning. HSK Handbooks of Linguistics and Communication Science Series. Mouton de Gruyter (2012)
5. Androutsopoulos, I., Malakasiotis, P.: A survey of paraphrasing and textual entailment methods. Journal of Artificial Intelligence Research **38**(1) (2010) 135–187
6. Bos, J., Markert, K.: Recognising textual entailment with robust logical inference. In: MLCW 2005, volume LNAI 3944. (2006) 404–426
7. MacCartney, B., Manning, C.D.: Natural logic for textual inference. In: Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing, Prague, Association for Computational Linguistics (June 2007) 193–200
8. Kamp, H., Reyle, U.: FROM DISCOURSE TO LOGIC-Introduction to Modeltheoretic Semantics of Natural Language, Formal Logic and Discourse Representation Theory. Kluwer Academic Publishers (1993)
9. Steedman, M., Baldridge, J.: Combinatory categorial grammar. In Borsley, R., Borjars, K., eds.: Non-Transformational Syntax: Formal and Explicit Models of Grammar. Wiley-Blackwell (2011)
10. Dekker, P.: A guide to dynamic semantics. Technical report, ILLC/Department of Philosophy, University of Amsterdam (2008)
11. van Eijck, J.: Natural logic for natural language. In ten Cate, B., Zeevat, H., eds.: 6th International Tbilisi Symposium on Logic, Language, and Computation Batumi, Georgia. Springer (2007) 216–230
12. Kamp, H., Van Genabith, J., Reyle, U.: Discourse representation theory. In Gabbay, D.M., Guenthner, F., eds.: Handbook of Philosophical Logic. Volume 15 of Handbook of Philosophical Logic. Springer Netherlands (2011) 125–394
13. Muskens, R.: Combining montague semantics and discourse representation. Linguistics and Philosophy **19** (1996) 143–186

Metodología para la identificación de secuencias verbales fijas

Belém Priego Sánchez[1], David Pinto[2], Salah Mejri[1]

[1] Laboratoire Lexiques, Dictionnaires, Informatique, CNRS (UMR 7187)
Université Paris 13, Sorbonne Paris Cité, Francia

[2] Facultad de Ciencias de la Computación
Benemérita Universidad Autónoma de Puebla, México
belemps@gmail.com, dpinto@cs.buap.mx, smejri@ldi.univ-paris13.fr

Resumen Las locuciones verbales fijas designan un tipo particular de construcciones fijadas. En nuestro enfoque concebimos una secuencia verbal fija como un grupo de palabras en las que al menos una es un verbo que funciona como núcleo del predicado. En este artículo se presenta una aproximación computacional que permite reconocer automáticamente este tipo de estructuras lingüísticas en corpus de diferentes dominios. En el contexto de esta investigación, cuando hablamos de "reconocer" nos referimos al hecho de identificar los limites inferior y superior que enmarcan una secuencia de palabras que tienen un alto grado de probabilidad de ser una expresion verbal fija.

Palabras clave: Secuencia verbal fija, aprendizaje automático, léxico

1. Introducción

La fijación ha tenido un auge remarcable en los últimos años, sobre todo en los estudios referentes al léxico, esto gracias a que los especialistas en la materia han entendido que se trata de un fenómeno transversal que abarca todas las partes de la oración [20], lo que implica todas las dimensiones del lenguaje: lingüísticas, pragmáticas, culturales, entre otras [21]. La mayoría de los estudiosos de la lengua concuerdan en decir que la fijación es una propiedad inherente a las lenguas naturales, lo que le permite ocupar un lugar central dentro de la descripción de éstas, y es por tanto tomada en cuenta en todos los trabajos que tengan una perspectiva práctica (la traducción, la constitución de diccionarios, la enseñanza de idiomas y el tratamiento automático del lenguaje en tareas como la clasificación automática).

Al hablar de tratamiento automático del lenguaje, nos percatamos que la fijación impide contar con una modelización adaptada a las exigencias de los sistemas informáticos, y de ahí, surge la necesidad de contar con recursos lingüísticos suficientemente vastos y claramente estructurados para ser automatizados.

La segmentación de una oración en palabras es tal vez la primera operación efectuada en un tratamiento automático de la lengua. Pero el término palabra es lingüísticamente inapropiado porque corresponde en informática a una entidad,

llamada token, delimitada por separadores gráficos (blancos, retorno de línea, etc.). La noción de palabra es mucho más compleja, y cuando decimos complejo nos referimos a la dificultad posible que determina su polilexicalidad. En efecto, mientras que los informáticos nos concentramos sobre la palabra simple, los lingüistas se concentran en las palabras complejas que son también importantes en el tratamiento de las lenguas. Este artículo propone la unión de ambos conocimientos (informático, lingüístico) para el tratamiento automático de este tipo de secuencias, que debe de ser tratado correctamente porque la clasificación correcta de éstas secuencias dichas fijas es útil para numerosas aplicaciones como la traducción, la extracción de información, la clasificación, la constitución de diccionarios, la enseñanza de idiomas, entre otras.

El tratamiento automático de las secuencias fijas en un texto implica dos tareas de suma importancia:

1. La localización automática de las secuencias polilexicales.
2. El análisis sintáctico y semántico.

La primera tarea debe encarar la dificultad que se plantea en varios estudios realizados, la cual consiste en el hecho de que la mayoría de las secuencias fijas presentan una misma identidad formal tal como lo hacen las secuencias libres. La segunda tarea plantea la dificultad de la relación entre el sentido de la Secuencia Fija (SF) y su sintaxis [4]. Muy a menudo se ha dado preferencia, en cuanto a los estudios sobre la fijación, a las secuencias que rechazan cualquier tipo de variación sintáctica y cuya significación es opaca. Incluso los estudios que han unido la sintaxis con la semántica han privilegiado un aspecto semántico, la opacidad, asociándolo con el bloqueo sintáctico. Todo esto nos indica que se debe contar con una teoría que tome en cuenta, al momento de describir la SF, su entorno sintáctico.

Nuestra aproximación consiste en un recurso informático que localiza automáticamente las secuencias verbales fijas del Español Mexicano en un corpus de diferentes dominios; es decir, toma en cuenta la primer tarea implicada en el tratamiento automático de las secuencias fijas en un texto. Con respecto a la segunda tarea por el momento solo se han etiquetado las expresiones verbales fijas y analizado patrones morfosintácticos que sirven como base para identificar éstas construcciones en un texto.

El objetivo de este proyecto es el de reconocer secuencias verbales fijas, extraerlas y validarlas. La extracción no ha sido una tarea fácil pero hemos llegado a identificar expresiones verbales fijas en un texto escrito en lenguaje natural. La validación es una tarea más compleja, por tanto requiere de más tiempo y de un estudio profundo tanto sintáctico como semántico, por tal ese análisis lo tratamos como trabajo futuro pero partiendo de diferentes hipótesis. En el contexto de esta investigacion, cuando hablamos de “reconocer” nos referimos al hecho de identificar los limites inferior y superior que enmarcan una secuencia de palabras que tienen un alto grado de probabilidad de ser una expresión verbal fija.

2. Expresiones fijas

La comunicación humana depende, en gran medida, del conocimiento enciclopédico del mundo que tienen los hablantes. Todos tenemos en común universales humanos (escenarios, frames), que, sin embargo, vienen matizados por cada cultura, de manera que poseen valores específicos y diferenciados. Así, las sociedades poseen sus propias visiones del mundo, actitudes y conductas sobre diversos temas y circunstancias, distinguiéndose así las distintas comunidades culturales. Una de las disciplinas lingüísticas que mejor recoge éstas especificidades lingüísticas es la fraseología, pues pone de relieve numerosos aspectos socio-culturales [28].

Las Unidades Fraseológicas (UF), tambien llamadas fraseologismos, pertenecen a lo que Coseriu [6] denomina "discurso repetido", caracterizándose, sobre todo, por los tres rasgos siguientes:

1. Su caracter poliléxico, que las distingue de las palabras aisladas, simples o compuestas, de la lengua.
2. Su fijación, que implica su memorización como si constituyeran un todo inseparable, tal y como se almacenan las unidades simples.
3. Su idiomaticidad u opacidad léxica, rasgo que, sin embargo, puede faltar, como ocurre en las llamadas colocaciones, clasificación que aclararemos en seguida.

En efecto, las UF suelen clasificarse atendiendo a dos parámetros :

1. Que se trate de oraciones o de sintagmas.
2. Que sean idiomáticas o no.

Nuestro objeto de estudio concierne a las secuencias fijas, conocidas también con el nombre de de expresiones fijas, locuciones, expresiones idiomáticas, etc. Una secuencia fija es un grupo de palabras, no necesariamente contiguas, posedoras de una unidad semántica (sentido global), una fijación a la vez morfológica (bloqueo del número), lexical (bloqueo del paradigma conmutacional) y sintáctica (bloqueo de la pasivación, de la relativización para las secuencias verbales) [16]

En el estudio del 2011 de Mejri [23], se hace distinción entre las secuencias fijas y otros dos tipos de conceptos:

1. Las *secuencias totalmente fijas*, no aceptan ninguna modificación. El conjunto es un bloque inmutable y cuyo tratamiento necesita referente simple en un diccionario.
2. Las *colocaciones*: secuencias repetidas que aparecen frecuentemente en conjuntos. Ellas pueden ser propias de un dominio (colocación terminológica según [27]) o típicas de una lengua (como los verbos soporte o los verbos apropiados).

Como se menciona en [26], las locuciones verbales, colocaciones, construcciones con verbo soporte apropiados, entre otros, son denominaciones variadas

para designar un tipo particular de construcciones fijadas. Nosotros estudiamos más precisamente las *Secuencias Verbales Fijas (SVF)*, que en este trabajo se conciben como un grupo de palabras en las que al menos una es un verbo que funciona como núcleo del predicado, es decir, expresiones idiomáticas de significado no composicional. Son sintagmas fijos e idiomáticos cuya interpretación no se obtiene de la suma de sus partes, tomadas por separado [19].

La problemática de las secuencias fijas, y de manera más particular de las SVF, viene del hecho que no son totalmente fijas [1,13,12,15]. De hecho, las SVF permiten ciertas modificaciones de orden sintagmático y/o paradigmático creando también los grados de fijación [12]. Sin embargo, no es aparentemente posible definir a priori las transformaciones realizables de una secuencia. En el trabajo de Villada [24] remarca que "no hay presencia uniforme o ausencia de restricciones sintácticas en todas las expresiones fijas dado que no todas las expresiones fijas exhiben la misma versatilidad sintáctica". En [2] se postula que las secuencias de la misma estructura sintáctica no aceptan las mismas libertades tranformacionales.

Estos problemas desencadenan un reto importante en términos de la identificación y clasificación semántica. Por esta razón consideramos relevante la construcción de recursos, técnicas y herramientas para el tratamiento y análisis de expresiones verbales fijas.

2.1. Trabajo relacionado

Existen diversos trabajos que se centran principalmente en el estudio del grupo estable de dos o más palabras que funcionan como una unidad léxica con significado propio, no derivado de la sumas de sus componentes, es decir, su estudio se basa en las locuciones; una locución es una secuencia fija de palabras con un sentido unitario que en muchos casos no se puede deducir el significado de cada una de las palabras por separado. Si nos centramos en uno de los muchos trabajos que existen, tal como se menciona en [22] , las secuencias fijas se asemejan debido a su funcionamiento sintáctico-semántico, a predicados monolexicales, así, su descripción lingüística se realiza con la ayuda de las mismas herramientas utilizadas para describir las unidades léxicas simples. Las locuciones verbales ilustran perfectamente la saturación total, tal como se ha indicado en [22], y partiendo del hecho que remarca que las locuciones verbales tienen una ruptura paradigmática, concentramos nuestra atención en las locuciones verbales.

En el trabajo de Mogorron [14] se menciona que el significado de las locuciones verbales no puede deducirse de la suma de los significados individuales de cada uno de sus componentes. En el mismo artículo se menciona que las locuciones verbales tienen como principal propiedad a la idiomaticidad, centrándose en la elaboración de un diccionario bajo la forma de una base de datos que incorpor el mayor número de sentencias posibles de este tipo de expresiones. En general, la recolección de locuciones verbales que se ha hecho en este trabajo demuestra el interés de estudio de este campo lingüístico.

La pregunta ahora es: ¿cómo pueden extraerse este tipo de secuencias?, existen diferentes aproximaciones y en este artículo, lo abordamos desde el punto de vista sintáctico, estadístico, híbrido y basado en diccionarios.

La primera aproximación es puramente sintáctica. En el trabajo de Laporte et al. [10] se utilizan patrones sintácticos producidos de sustantivos compuestos y se los propone a un transductor (con la herramienta Unitex). Permitiendo ciertas transformaciones (inserción, coordinación y otras), recuperan así secuencias nominales que corresponden sintácticamente a sustantivos compuestos.

La segunda aproximación es puramente estadística. Estos métodos utilizan medidas estadísticas para determinar la relación entre los elementos de las secuencias. En este tipo de aproximación podemos citar a Caseli et al. [5] que en su trabajo se enfoca al uso estadístico basado en la alineación de la identificación de expresiones multipalabras en corpora. Utilizan varias fuentes de datos: incluido un corpus paralelo (inglés-portugués), corpus basado en dominio (pediátrico) y examinan con un segundo lenguaje que puede proporcionar pistas para resolver este tipo de tareas.

La aproximación más utilizada es la híbrida que involucra la sintaxis y la estadística. Algunos trabajos como [18,7,29] comienzan por un filtro lingüístico (selección de lexemas, patrones sintácticos) para enseguida tomar una decisión basada sobre el cálculo probabilístico (información mutua, logaritmo de máxima verosímilitud, entre otros). Otros a la inversa, generan el primer filtrado por criterios estadísticos para efectuar luego su elección sobre criterios lingüísticos [27]. Otro trabajo basado en este tipo de aproximación es el de Dias [9] en el cual se describe un sistema hibrido que extrae multipabras candidatas, la solución propuesta en este trabajo identifica automáticamente patrones sintácticos relevantes en el corpus, evalua el sistema con el *Brown Corpus* y propone una medida de asociación que llama GenLocalMax [8] la cual permite calcular el grado de fijación de una secuencia de más de dos palabras no necesariamente contiguas. Sin embargo el enfoque depende del tamaño del corpus.

Este tipo de métodos híbridos son más precisos, no obstante, permiten la extracción de datos terminológicos (a menudo nominales) más que la extracción de secuencias fijas; es decir, que pueden encontrarse en cualquier texto sea cual sea el dominio. Las posibles modificaciones integradas son del orden de la expansión de la secuencia.

Otra aproximación permite extraer unidades fraseológicas, basándose en el uso de diccionarios electrónicos, como son los trabajos de [17,11,3], por citar algunos. Existe también la aproximación basada en las gramáticas de adjunción de árboles, un método propuesto por [1], gracias al cual se pueden extraer secuencias fijas a pesar de su discontinuidad (inserción, modificación) y sus cambios sintácticos. Esto implica que la descripción transformacional sea completa.

El término expresiones multipalabra se ha hecho más polular a partir del 2000, probablemente por las diferentes iniciativas [25]. Sin embargo, el estudio de este tipo de estructuras es viejo para el campo de la lingüística. Al tratar de clasificar lingüísticamente este fenómeno involucramos conocimientos de léxicos, sintaxis, semántica, y muchas más áreas lingüísticas pero también observamos

que se necesita unir con la computación, es decir, el estudio de las secuencias fijas se encuentra entre los dos niveles. La consecuencia de ello, es que cualquier aproximación lingüística computacional incluye ambos niveles en los modelos con el fin de lograr más robustez.

3. Creación del recurso: Diccionario de Mexicanismos

El diccionario del cual se han extraído las secuencias verbales fijas con las que se han trabajado en este artículo es el Diccionario de Mexicanismos, el cual podríamos denominar base del conocimiento. Este diccionario es un resultado de una investigación realizada por la Academia Mexicana y tiene tres características esenciales : es sincrónico, contrastivo y descriptivo.

- Sincrónico. Representa lo actual, los elementos léxicos de uso en la segunda mitad del siglo XX, y principios del XXI.
- Contrastivo. Es diferencial ; se preparó el diccionario tratando de comparar lo que se dice en México con lo que se dice en otros países de habla española y sobre todo con el español de la Península Ibérica.
- Descriptivo. Indica la realidad del uso ; ya que no establece criterios normativos, no se excluyen prestamos de otras lenguas (que son principalmente del inglés), ni neologismos.

En nuestro caso, se obtuvieron 1,157 expresiones verbales fijas del Diccionario de Mexicanismos, que nosotros llamamos base del conocimiento de expresiones verbales fijas y las cuales forman la base del presente trabajo. Éstas expresiones son tomadas como punto de partida para la identificación de expresiones verbales fijas candidatas en el dominio noticiero. En la siguiente sección se describe la herramienta utilizada para la realización de dicha tarea.

4. Herramienta de idenficación de las expresiones verbales fijas

En este artículo se propone un método que considera el uso de dos recursos importantes: la base del conocimiento de expresiones verbales fijas y el corpus del dominio de noticias. Ambos recursos han sido etiquetados morfosintácticamente y lematizados por TreeTagger.

Con la lematización realizada se toman en cuenta diferentes transformaciones de las secuencias verbales fijas. Una de las primeras transformaciones realizada es la *conjugación* del verbo; es decir, dado que se ha lematizado se toma en cuenta las diferentes flexiones verbales que tiene. Continuando con las transformaciones tenemos la *flexión* que consiste en que la secuencia es modificada cambiando el número de sustantivos y su actualizador asociado; por ejemplo: *vender como pan caliente, vender como panes calientes*. Si bien sabemos que necesitan ser tomadas en cuenta otras transformaciones como la sustitución, inserción, supresión de elementos, negación/afirmación, inversión, entre otras. El trabajo sigue desarrollándose y estan considerándose este tipo de características.

4.1. Descripción del corpus

En esta sección se describe el corpus en español de noticias de los periódicos de la República Mexicana, el cual sirve como recurso para la identificación de expresiones verbales fijas candidatas en los diferentes dominios analizados.

Un corpus es un conjunto de textos recopilados, ya sea de un mismo tema o varios. El propósito de este corpus en particular, es convertirse en un conjunto de datos que proporcione ejemplos de uso (con sus respectivos contextos) de varias expresiones verbales fijas a fin de analizar su uso y frecuencia en diversos dominios. Adicionalmente, este corpus podría ser útil en algoritmos de aprendizaje automático para generar modelos que identifiquen automáticamente este tipo de estructuras lingüísticas. Dependiendo de la naturaleza de los algoritmos y de la tarea, las expresiones existentes en el corpus podrian estar previamente desambigüadas o no.

El material de trabajo es un corpus escrito –contiene solo el idioma español–, abierto –en constante crecimiento–, especializado –corresponde al género de noticias– y finalmente periódico, es decir, la colección de documentos (noticias) utilizada consta de relatos periodísticos ocurridos a partir del año 2007 y hasta el año 2013, y recopilados de una agencia mexicana de noticias.

Si bien, los documentos obtenidos presentan diferentes metadatos, para nuestro caso han resultado útiles los siguientes:

- Título de la noticia
- Dominio (Sociedad, Espectáculos, Política, Futbol, entre otros)
- Fecha de la noticia
- Noticia (el texto mismo de la noticia informativa)

Es cierto que de los documentos de noticias se pueden obtener más datos, sin embargo, dada la tarea presentada en este artículo, con la informacioń mencionada anteriormente es suficiente. El corpus utilizado para esta tarea consta de 378,890 noticias, un total de 4,579,284 oraciones y alrededor de 1,159,571 palabras. Cabe mencionar que solo se utilizó una parte del corpus total de 1,895,983 de noticias para poder balancear la base del conocimiento de expresiones verbales fijas con respecto al corpus.

4.2. Metodología propuesta

En esta sección se describe la metodología para la identificación automática de expresiones verbales fijas. La aproximación está basada en técnicas de aprendizaje automático (machine learning), una rama de la inteligencia artificial que se refiere a la construcción y estudio de los sistemas computacionales que pueden aprender a partir de datos. En este caso, necesitamos un corpus etiquetado por expertos que indique si un conjunto de palabras es o no una expresión verbal fija. El aprendizaje automático por lo general construye un modelo de clasificación que se utiliza para el etiquetado automático de nuevas muestras. Comenzamos este enfoque teniendo en cuenta que existe dependencia entre las palabras en

el contexto de las expresiones verbales fijas. Por lo tanto, para nuestros experimentos preliminares hemos utilizado las técnicas de aprendizaje automático que tienen en cuenta el orden de las palabras conocido como *Conditional Random Fields* que fue originalmente propuesto por la Universidad de Stanford.

Las técnicas de aprendizaje automático son capaces de aprender el proceso humano de la identificación de las expresiones verbales fijas basadas en características alimentados en el clasificador. Además, es capaz de detectar completamente nuevos tipos de expresiones verbales fijas que comparten propiedades con las almacenadas en el corpus de entrenamiento, la cual destaca que tan poderosa puede ser la Computación basada en métodos automáticos de aprendizaje.

Como mencionamos antes, los métodos de aprendizaje automático son alimentados por los datos etiquetados que necesitan ser construidos manualmente. Pero, tener un gran corpus de expresiones fijas marcadas con su respectivo contexto no es una tarea fácil. Por este motivo, se propone un mecanismo automático para la construcción de dicho recurso. Primero necesitamos la construcción de una base del conocimiento de expresiones verbales fijas que luego sea utilizada para identificar construcciones similares en un corpus de textos. Después utilizamos un sistema de recuperación de información para la búsqueda de contextos en el que se produce una estructura similar a una expresión verbal fija.

En resumen, la metodología propuesta para la identificación automática de expresiones verbales fijas candidatas del español mexicano es presentada de la siguiente manera:

1. Construir una base del conocimiento de Expresiones Verbales Fijas para el Español (SVFE).
2. Reunir un conjunto de documentos escritos en español en el que se espera encontrar SVFE.
3. Construir un gran corpus etiquetado de SVFE utilizando técnicas de recuperación de información.
4. Construir un modelo de clasificación para identificar SVFE candidatas usando técnicas de aprendizaje automático.
5. Identificar SVFE candidatas en textos no etiquetados.

En la figura 1 se puede observar la metodología que proponemos para la identificación automática de las expresiones verbales fijas; en la cual se puede observar que existe un proceso de aprendizaje automático y para el proceso de etiquetamiento automático utilizamos un sistema de recuperación de información para la búsqueda de contextos en los que se produce una estructura similar a una expresión verbal fija.

Partiendo de la metodología propuesta, a continuación presentamos una muestra de expresiones verbales fijas lo cual corresponde a la construcción de la base del conocimiento de SVFE.

poner a disposición, salir a relucir, dar a conocer, ganar terreno, llevar a cabo, estar detrás, hacer acto de presencia...

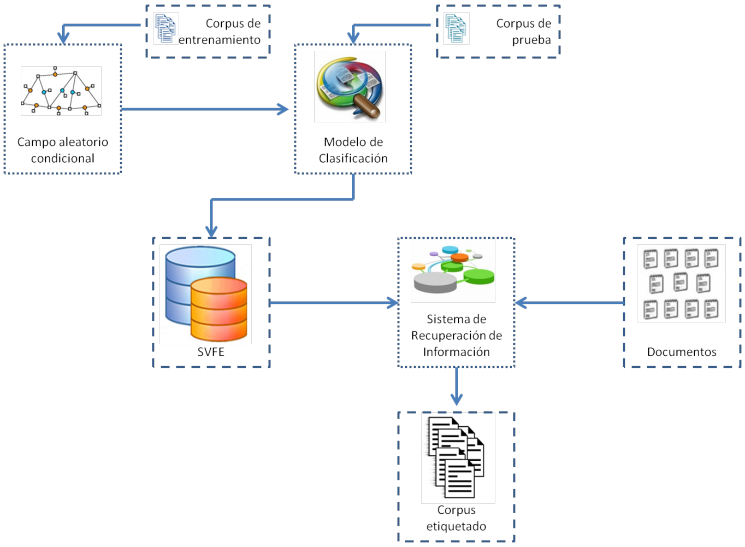

Fig. 1. Metodología propuesta para la identificación de expresiones verbales fijas.

Como se mencionó en la sección 4.1 para la realización de esta tarea hemos recolectado relatos periodísticos de una agencia mexicana de noticias y esa recopilación corresponde al punto 2 de la metodología.

Continuando con la metodología y aplicando los pasos 3, 4 y 5; damos un ejemplo del etiquetado que se realizó y lo que tenemos como resultado final en una noticia que contiene una expresión verbal fija, esto se visualiza en la imagen 2.

poner a disposición *-> Agotados los trámites legales en la dependencia policial el presunto delincuente junto el arma de fuego cartuchos y el vehículo asegurado fueron <EVF>**puestos a disposición**</EVF> del ministerio Público*

Fig. 2. Ejemplo de una noticia que contiene una expresión verbal fija.

Del total de documentos periodísticos que utilizamos para llevar a cabo nuestros experimentos, hemos encontrado que 222,857 noticias contienen expresiones verbales fijas, la etapa siguiente es validar éstas expresiones ya sea sintáctica o semánticamente e incluyendo más características.

5. Conclusiones y Perspectivas

En este trabajo se ha presentado una metodología para la identificación automática de expresiones verbales fijas candidatas para el español de México.

La metodología ha sido probada preliminarmente empleando recursos léxicos del mismo idioma. Como resultado hemos compilado 1,157 expresiones verbales fijas en solamente 378,890 de los 1,895,983 textos del dominio periodistico, de lo cual se obtuvo una muestra de textos anotados con las ocurrencias de las expresiones. Este último recurso debe ser ampliado y perfeccionado con el fin de tener un corpus de entrenamiento que pueda ser utilizado para la construcción de modelos de clasificación que permitirá identificar automáticamente la presencia de expresiones verbales fijas en texto plano y ampliar el experimento con todos los textos disponibles en el corpus.

La utilización de un método computacional que identifique ciertas expresiones de palabras en específico expresiones verbales fijas, es una tarea difícil y compleja, en este trabajo se presentó una aproximación computacional para tratar de resolver dicha tarea. Sin embargo, en el trabajo es necesario tomar en cuenta características semánticas y sintácticas, así como las diferentes transformaciones que tienen las expresiones. Las expresiones verbales fijas pueden ser categorizadas en numerosos tipos no solo por su grado de fijación si no por su literalidad, su duplicidad de sentido, su opacidad o según el dominio en el cual se encuentra, así que este tipo de características también podrían ser tomadas en cuenta para que la aproximación presentada tenga mejores resultados a la hora de identificar las expresiones. Consideramos que sería de gran utilidad tener un corpus de referencia, con el fin de realizar más pruebas y que los resultados puedan ser comparados con medidas estadísticas, así como con otros recursos.

Agradecimientos. Este trabajo ha sido parcialmente apoyado por el Consejo Nacional de Ciencia y Tecnología - CONACYT referencia 218862/314461.

Referencias

1. Abeillé, A., Schabes, Y.: Parsing idioms in lexicalized tags. In: Somers, H.L., Wood, M.M. (eds.) EACL. pp. 1–9. The Association for Computer Linguistics (1989), http://dblp.uni-trier.de/db/conf/eacl/eacl1989.html#AbeilleS89
2. Balibar-Mrabti, A.: Semi-figement et limites de la phrase figée. In: LINX. pp. 34–54 (2005)
3. Bungum, L., Gambäck, B., Lynum, A., Marsi, E.: Improving word translation disambiguation by capturing multiword expressions with dictionaries (2013)
4. Buvet, P.A.: Vers l'elaboration d'un dictionnaire unique des predicats du franais : Deesse. dictionnaire electronique syntactico-semantique. In: Description linguistique pour le traitement automatique du franais. pp. 23–42 (2008)
5. Caseli, H.d.M., Villavicencio, A., Machado, A., Finatto, M.J.: Statistically-driven alignment-based multiword expression identification for technical domains. In: Proceedings of the Workshop on Multiword Expressions: Identification, Interpretation, Disambiguation and Applications. pp. 1–8. MWE '09, Association for Computational Linguistics, Stroudsburg, PA, USA (2009), http://dl.acm.org/citation.cfm?id=1698239.1698241
6. Coseriu, E.: Principios de semántica estructural. In: Gredos. p. 113. Madrid (1977)
7. Daille, B.: Study and implementation of combined techniques for automatic extraction of terminology. In: Klavans et Resnik 1996. pp. 29–36 (1996)

8. Dias, G.: Extraction Automatique dAssociations Lexicales a partir de Corpora. Ph.D. thesis, New University of Lisbon (Portugal) and LIFO University of Orleans (France (2002)
9. Dias, G.: Multiword unit hybrid extraction. In: Proceedings of the ACL 2003 Workshop on Multiword Expressions: Analysis, Acquisition and Treatment - Volume 18. pp. 41–48. MWE '03, Association for Computational Linguistics, Stroudsburg, PA, USA (2003), http://dx.doi.org/10.3115/1119282.1119288
10. Eric Laporte, T.N., Voyatzi, S.: A french corpus annotated for multiword nouns. In: LREC Workshop Towards a Shared Task for Multiword Expressions (MWE 2008). pp. 27–30 (2008)
11. Grégoire, N.: Design and implementation of a lexicon of dutch multiword expressions. In: Proceedings of the ACL 2007 Workshop on A Broader Perspective on Multiword Expressions, Prague, Czech Republic. pp. 17–24 (2007)
12. Gross, G.: Les expressions figées en francais noms composés et autres locutions. In: Ophrys. Paris, France (1996)
13. Gross, M.: Une classification des phrases figées du francais. In: Revenue québécoise de linguistique (1982)
14. Huerta, P.M.: Estudio contrastivo lingüístico y semántico de las construcciones verbales fijas diatópicas mexicanas/espaola. In: Las construcciones verbo-nominales libres y fijas. pp. 179–198 (2010)
15. Lamiroy, B.: Le problème central du figement est le semi figement. In: LINX (2005)
16. Lamiroy, B.: Les expressions figées: à la recherche d'une définition. In: Blumental et Mejri 2008. pp. 85–98 (2008)
17. Laporte, E., Voyatzi, S.: An electronic dictionary of french multiword adverbs. In Language Resources and Evaluation Conference. Workshop Towards a Shared Task on Multiword Expressions pp. 31–34 (2008)
18. Manning, C.D., Schutze, H.: Foundations of Statistical Natural Language Processing. MIT Press, Cambridge, MA, USA (1999)
19. Martinez-Blasco, I.: Verbos soporte y fijacin lexica. In: Las construcciones verbo-nominales libres y fijas. pp. 47–59 (2008)
20. Mejri, S.: Le figement lexical. descriptions linguistiques et structuration semantique. In: Publications de la faculte des lettres de Manouba, Tunis (1997)
21. Mejri, S.: Categories linguistiques et etiquetage de corpus. In: Linformation grammaticale, Peeters, Paris (2007)
22. Mejri, S.: Constructions verbes supports, collocations et locutions verbales. In: La traduction des MEJRI Salah (2008)
23. Mejri, S.: Les dictionnaires électroniques sémantico-syntaxiques. In: Cardoro er al. 2011. pp. 159–188 (2011)
24. Moiron, M.V.: Data-driven identification pf fixed expressions and their modifiability. Ph.D. thesis, University of Groningen, Pays-Bas (2005)
25. Ramisch, C., Villavicencio, A., Kordoni, V.: Introduction to the special issue on multiword expressions: From theory to practice and use. TSLP 10(2), 3 (2013), http://doi.acm.org/10.1145/2483691.2483692
26. Sfar, I.: Polylexicalite et continuite predicative: le cas des locutions verbales figées. In: Las construcciones verbo-nominales libres y fijas. Aproximación contrastiva y traductológica. pp. 213–221 (2008)
27. Smadja, F.: Retrieving collocations from text: Xtract. Comput. Linguist. 19(1), 143–177 (Mar 1993), http://dl.acm.org/citation.cfm?id=972450.972458
28. Soler, N.P., Rodríguez, J.J.B.: Unidades fraseológicas y variación. In: Ogigia. Revista electrónica de estudios hispánicos. pp. 43–52 (2008)

29. Watrin, P.: Collocations et traitement automatique des langues. In: Lexis and Grammar, Bonifacio (2007)

Un modelo para detectar la similitud semántica entre textos de diferentes longitudes

Darnes Vilariño, Mireya Tovar, Beatriz Beltrán, Saúl León

Benemérita Universidad Autónoma de Puebla,
Facultad de Ciencias de la Computación,
Avenida San Claudio, 14 Sur, Ciudad Universitaria.
Puebla México.
\{darnes,mtovar,bbeltran,saul\}@cs.buap.mx
http://nlp.cs.buap.mx

Resumen En el presente trabajo se desarrolla un modelo para resolver el problema de similitud semántica entre textos de diferente longitud. Se propone extraer características léxicas, características basadas en conocimiento y características basadas en corpus, con el objetivo de desarrollar un modelo de aprendizaje supervisado. El modelo fue desarrollado utilizando regresión logística de la herramienta Weka. Los resultados obtenidos sobre los datos ofrecidos en el marco del Semeval 2014, han sido buenos para dos tipos de corpora.

Palabras clave: Similitud Semántica, Información mutua, Análisis semántico latente

1. Introducción

La similitud semántica textual tiene como objetivo capturar cuando el sentido de dos textos es similar. Este concepto difiere de encontrar el grado de similitud textual, que está solamente interesado en medir el número de componentes léxicas que comparten ambos textos, es decir, el valor de similitud textual puede variar cuando uno de los textos no está completo y no se logra con la poca información medir cuanto realmente comparten, pero no logra medir la similitud de ambos textos en cuanto al sentido que se desea expresar.

Encontrar la similitud semántica entre pares de textos se ha convertido en un gran reto para los especialistas en Procesamiento de Lenguaje Natural (PLN), ya que se puede aplicar en diferentes tareas de PLN, tales como máquinas de traducción, construcción automática de resúmenes, atribución de autoría, pruebas de lectura comprensivas, recuperación de información y muchas otras, que necesitan medir el grado de similitud entre dos textos dados.

Esta problemática se torna aún más compleja cuando se desea encontrar la similitud semántica textual entre textos de diferentes tamaños ya que ambos textos no ofrecen la misma cantidad de información y no se logra descubrir facilmente si el sentido del texto más pequeño es el mismo sentido del texto de mayor longitud. En particular esta problemática se ve reflejada cuando se desea

encontrar el grado de similitud entre un párrafo y una sentencia, una sentencia y una frase, una frase y una palabra y una palabra y un sentido. Esta tarea se presentó en el marco de la conferencia internacional Semeval 2014, como la tarea 3, que se denominó similitud semántica entre niveles [8]. El modelo que se desarrolla debe brindar un grado de similitud considerando los siguientes niveles:

- *4* Los dos textos tienen significados muy similares y las ideas más importantes, conceptos o acciones en el texto más grande están representadas en el texto más pequeño.
- *3* Ambos textos comparten muchas ideas importantes, conceptos o acciones, pero sin embargo lo que se expresa en el texto de menor longitud es similar, pero no idéntico a lo que se expresa en el texto de mayor longitud.
- *2* Ambos textos tienen significados diferentes, pero los conceptos, ideas o acciones en el texto de menor longitud tienen relación con el sentido que ofrece el texto de mayor longitud.
- *1* Los dos textos expresan ideas, conceptos y acciones completamente diferentes, pero es posible que se puedan encontrar juntos en un texto que hable del mismo tema.
- *0* Ambos textos no significan lo mismo, ni hablan del mismo tema.

El artículo está estructurado de la siguiente manera: en la sección 2 se discuten las diferentes metodologías desarrolladas para ofrecer el grado de similitud semántica entre textos, en la sección 3 se presentan las características seleccionadas, en la sección 4 se discute como se representa la información extraída de ambos textos y se presenta el modelo propuesto. La sección 5 muestra las características de las colecciones de datos, y el análisis de los resultados obtenidos. Por último en la sección 6 se presentan las conclusiones de la investigación y el trabajo a futuro.

2. Trabajo Relacionado

La búsqueda del grado de similitud semántica entre textos se ha propuesto como tarea dentro del marco de la conferencia internacional Ejercicios de Evaluación Semántica [1], [2] por lo que se le está dedicando bastante atención en los últimos años. Muchos de los modelos desarrollados han hecho énfasis fundamentalmente en la búsqueda de las características que comparten ambos textos, para lograr con ello detectar si ambos textos poseen un sentido similar. Se han propuesto modelos markovianos, se han propuesto modelos de traducción automática, buscando la alineación de ambos textos y se ha introducido en investigaciones la búsqueda de ciertos patrones sintácticos que comparten. A continuación se presentan alguno de los trabajos desarrollados.

En la investigación reportada en [15], se busca establecer un modelo semántico que permita identificar los argumentos implícitos en los textos. Se plantea que a pesar de ser una tarea fácil para un lector humano, resulta complicado para las computadoras, debido a que no hay una manera de indicarle a éstas, que un argumento puede ser inferido varias veces en el texto. En este trabajo se propone

una aproximación por inducción que explota la información complementaria obtenida de un par de textos comparables. Esto significa que se desarrollan modelos que permiten alinear ambos textos, esta es una técnica que se ha utilizado para el desarrollo de diccionarios estadísticos en las tareas de traducción automática. La metodología propuesta en este trabajo ha permitido encontrar la similitud entre textos de longitudes similares con un grado de precisión de un 83 %.

En el trabajo desarrollado por [4], se propone un modelo Markoviano para determinar la cantidad de información que comparten dos sentencias de una longitud aproximada. En el marco de esta investigación se desarrollan un conjunto de reglas que permiten inferir cuando dos textos poseen el mismo significado. La precisión reportada para los datos del test a partir de los modelos construidos fue de 73 %, con los datos de prueba que se ofrecieron en la tarea 6 de la conferencia [1].

De igual modo, se puede hacer mención al trabajo [18], en el cual se muestra un método computacional que identifica las metáforas en textos sin restricción respecto a su interpretación, es decir se convierte al problema de encontrar el parecido entre textos, como encontrar si un texto es el parafraseo del otro. Se define a la interpretación en metáforas como una tarea de encontrar una parárasis literal para una palabra usada metafóricamente, además de introducir el concepto de parafraseado inverso simétrico como un criterio para la identificación de metáforas. Esto lo logra haciendo experimentos en los que se manejan relaciones de verbo-sujeto y sujeto-objeto indirecto, sin embargo la precisión que logran usando los datos del test y training en la tarea 6 del SemEval 2013 es solamente del 66 %.

La investigación presentada por [16], plantea que para lograr medir el grado de similitud semántica textual entre textos estos deben ser representados no con términos unipalabras, sino con términos multipalabras. Para encontrar entonces la similitud en este tipo de representación acuden a traducciones automáticas, para lo que usan a la herramienta PanLex, la cual le permite la creación de un diccionario estadístico. Si la traducción es posible, esto quiere decir que es equivalente un término en un texto, con una expresión multipalabra en el otro. El sistema presentado ofreció una precisión del 66 % en el marco del Semeval 2013.

Otro trabajo al que se puede hacer mención es el desarrollado por [17], cuyo objetivo es caracterizar los errores que se producen al realizar conteo de palabras. Muestra que los conteos léxicos pueden ser significativamente mejorados mediante el empleo de medidas de ambigüedad. Es decir encontrar sobre el texto aquellas palabras ambigüas, eliminar aquellas que no están totalmente relacionadas con el contexto. Esta propuesta no brindó resultados significativos, ya que solamente alcanzó el 50 % de precisión.

Otra manera de abordar esta tarea es considerandola como un problema de Question Answering, donde uno de los textos es la pregunta y el otro es la respuesta dada, es este el sentido del trabajo desarrollado por [3] donde se propone un modelo que mide el grado de similitud en funcin de que si la

respuesta logra responder a la pregunta. El modelo propuesto con los datos de entrenamiento y prueba dados en el Semeval 2013, ofreció una precisión del 73 % .

La propuesta desarrollada en la presente investigación tiene como objetivo fundamental medir el comportamiento de diferentes características y un modelo para expandir los textos de menor longitud. A continuación se discuten las características empleadas para representar a ambos textos.

3. Extracción de Características

Básicamente se han usado tres tipos diferentes de características: léxicas, basadas en conocimiento y basadas en Corpus. La primera de ellas cuenta la frecuencia de ocurrencia de los n-gramas de caracteres, *skip*-gramas[1], palabras y algunas relaciones léxicas como sinónimos e hiperónimos. Adicionalmente se han incluído otras dos características: El coeficiente de similitud de Jaccard entre dos textos, expandiendo cada término con el conjunto de sinónimos tomados de WordReference [5], y la similitud coseno entre los dos textos representados cada uno por la bolsa de n-gramas y *skip*-gramas de caracteres. En esta propuesta no se incluye ningún proceso de desambiguación después de expandir con los sinónimos de cada una de las palabras que conforman los textos.

Tabla 1. Características usadas para encontrar la similitud semántica textual

Característica	**Tipo**
n-gramas de caracteres ($n = 2, \cdots, 5$)	Léxica
skip-gramas de caracteres($skip = 2, \cdots, 5$)	Léxica
Número de palabras que comparten	Léxica
Número de sinónimos que comparten	Léxica
Número de hiperónimos que comparten	Léxica
Coeficiente de Jaccard con expansión de sinónimos	Léxica
Similitud Coseno con n-gramas y *skip*-gramas de caracteres	Léxica
Similitud de palabra de Leacock & Chodorow	Basada en conocimiento
Similitud de palabra de Lesk	Basada en conocimiento
Similitud de palabra de Wu & Palmer	Basada en Conocimiento
Similitud de palabra de Resnik's	Basada en conocimiento
Similitud de palabra de Lin	Basada en conocimiento
Similitud de palabra de Jiang & Conrath	Basada en conocimiento
Métrica de Rada Mihalcea usando Información mutua	Basada en Corpus
Métrica de Rada Mihalcea usando LSA	Basada en Corpus

El segundo conjunto de características considera las 6 medidas de similitud de palabras ofrecidas por la herramienta NLTK: Leacock & Chodorow [10],

[1] Son conocidos como n-gramas dispersos porque consideran un salto en cierto número de caracteres.

Lesk [11], Wu & Palmer [20], Resnik [14], Lin [12], y Jiang & Conrath[2] [7]. En este caso, se determina la similitud semántica entre dos textos como el máximo valor de similaridad obtenido entre los pares de palabras. El tercer conjunto de características considera dos medidas basadas en corpus, ambas medidas utilizan la métrica de similitud semántica textual ofrecida por Rada Mihalcea [13]. La primera usa Información mutua (PMI) [19] para el cálculo de la similitud entre pares de palabras, mientras que la segunda utiliza análisis semántico latente(LSA) [9] (implementado en el entorno estadístico R). Para esta investigación los valores de PMI y LSA fueron obtenidos en base a un corpus construído,con Europarl, el proyecto Gutenberg y el thesauro de OpenOffice. Todas las características utilizadas se pueden ver en la Tabla 1.

4. Metodología propuesta

Se extraen las características descritas anteriormente de los datos de entrenamiento, con el objetivo de desarrollar un modelo de clasificación. Se construye un vector para cada par de textos. Este vector es introducido en Weka para construir un modelo de clasificación basado en regresión logística.[6]

Dos de los corpus ofrecidos poseen textos de diferentes longitudes. Encontrar la similitud semántica en este caso se vuelve una tarea complicada, es por ello que se decide utilizar algún mecanismo de expansión de los textos de menor longitud. Para encontrar la similitud semántica entre frases y palabras, se expanden las palabras con los términos relacionados obtenidos de Flickr. Cuando se desea encontrar la similitud entre palabra y sentido, se expande el sentido utilizando la taxonomía de Wordnet, en ninguno de los casos se introduce un proceso de desambiguación, para eliminar aquellos términos no relativos al contexto con el que se está trabajando.

5. Resultados experimentales

A continuación se describen los datos que permitieron validar el modelo desarrollado.

5.1. Conjunto de Datos

Se dispone de un corpus conformado por 2,000 pares de textos para entrenamiento y 2,000 pares de texto para prueba. El conjunto de datos considera 500 pares para cada nivel, es decir 500 pares para párrafo-sentencia, 500 pares para frase-sentencia y así sucesivamente. Una descripción completa del conjunto de datos empleado se encuentra en el artículo que describe esta tarea en el marco del Semeval 2014. [8]

[2] Herramienta de Python para el procesamiento de Lenguaje Natural; http://www.nltk.org/

5.2. Resultados Obtenidos

Los resultados obtenidos se muestran en la Tabla 2. Nuestra aproximación, denominada *BUAP*, obtuvo un rendimiento por arriba del promedio reportado en la conferencia (*Promedio*). Cabe aclarar que dicho promedio se obtiene sobre el conjunto completo de ejecuciones reportadas en SemEval 2014, a pesar de que en esta tabla solamente se muestran los tres primeros resultados. El porcentaje de mejora de nuestro sistema con respecto al promedio se muestra en el último renglón de la tabla. Como puede apreciarse se ha obtenido un buen comportamiento del modelo desarrollado, cuando se calcula la similitud semántica entre párrafos y sentencias y entre sentencias y frases. Sin embargo la expansión de las palabras utilizando Flickr sobre el corpus de frase-palabra no ofreció resultados buenos, se considera que se debe a que la expansión realizada no está relacionada con el dominio del corpus, se introducen términos de caracter muy general. Cuando se expanden los sentidos al trabajar con los pares palabra- sentido, nuevamente se considera que se introduce un grado fuerte de ambieguedad, lo que provoca que el comportamiento también sea inapropiado.

Tabla 2. Resultados obtenidos en la Tarea 3 del Semeval 2014

Equipo	**Sistema**	**Párr-Sentencia**	**Sentencia-frase**	**Frase-Palabra**	**Palabra-Sentido**	**Rango**
SimCompass	run1	0.811	0.742	0.415	0.356	1
ECNU	run1	0.834	0.771	0.315	0.269	2
UNAL-NLP	run2	0.837	0.738	0.274	0.256	3
BUAP	**run**	**0.805**	**0.714**	**0.142**	**0.194**	**10**
Promedio	-	0.728	0.651	0.198	0.192	11-12
run - Promedio		8 %	6 %	-6 %	0 %	-

6. Conclusiones y Recomendaciones

En este artículo se presenta el modelo para resolver el problema planteado en la tarea 3 del Semeval 2014. Las características utilizadas ofrecen un buen comportamiento para detectar el grado de similitud semántica entre párrafo- sentencia y sentencia-frase, sin embargo la metodología de expansión propuesta para detectar el grado de similitud semántica entre los pares frase a palabra y palabra a sentido no fue correcta. Los resultados obtenidos fueron extremadamente bajos, lo que nos indica que deben utilizarse mecanismos de expansión diferentes. Se está trabajando en extraer de la web documentos donde aparezcan las palabras, para construir vectores representativos de cada una de ellas considerando 5 términos a la derecha y 5 términos a la izquierda, con esto se pretende detectar aquellos términos relacionados con la palabra que se desea expandir.

Referencias

1. Eneko Agirre, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, and Weiwei Guo. *sem 2013 shared task: Semantic textual similarity. In *2nd Joint Conference on*

*Lexical and Computational Semantics (*SEM)*, pages 32–43, Atlanta, Georgia, USA, 2013.

2. Eneko Agirre, Mona Diab, Daniel Cer, and Aitor Gonzalez-Agirre. Semeval-2012 task 6: A pilot on semantic textual similarity. In *Proceedings of the 6th International Workshop on Semantic Evaluation (SemEval-2012)*, pages 385–393, 2012.
3. Alexis Palmer Andrea Horbach and Manfred Pinkal. Using the text to evaluate short answers for reading comprehension exercises. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 520–524, Atlanta, Georgia, USA, June 2013. Association for Computational Linguistics.
4. Islam Beltagy, Cuong Chau, Gemma Boleda, Dan Garrette, Katrin Erk, and Raymond Mooney. Montague meets markov: Deep semantics with probabilistic logical form. In *Proceedings of the Second Joint Conference on Lexical and Computational Semantics (*Sem-2013)*, Atlanta, GA, June 2013.
5. Maya Carrillo, Darnes Vilariño, David Pinto, Mireya Tovar, Saul León, and Esteban Castillo. Fcc: Three approaches for semantic textual similarity. In *Proceedings of the 1st Joint Conference on Lexical and Computational Semantics (SemEval 2012)*, pages 631–634, Montréal, Canada, 2012.
6. Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. The weka data mining software: an update. *SIGKDD Explor. Newsl.*, 11(1):10–18, 2009.
7. Jay J. Jiang and David W. Conrath. Semantic similarity based on corpus statistics and lexical taxonomy. In *Proc of 10th International Conference on Research in Computational Linguistics, ROCLING'97*, pages 19–33, 1997.
8. David Jurgens, Mohammad Taher Pilehvar, and Roberto Navigli. Semeval-2014 task 3: Cross-level semantic similarity. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval-2014)*, Dublin, Ireland, 2014.
9. Thomas K. Landauer, Peter W. Foltz, and Darrell Laham. An Introduction to Latent Semantic Analysis. *Discourse Processes*, (25):259–284, 1998.
10. C. Leacock and M. Chodorow. Combining local context and wordnet similarity for word sense identification. In Christiane Fellbaum, editor, *MIT Press*, pages 265–283, 1998.
11. Michael Lesk. Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone. In *Proceedings of the 5th Annual International Conference on Systems Documentation*, pages 24–26. ACM, 1986.
12. Dekang Lin. An information-theoretic definition of similarity. In *Proceedings of the Fifteenth International Conference on Machine Learning*, ICML '98, pages 296–304, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc.
13. Rada Mihalcea, Courtney Corley, and Carlo Strapparava. Corpus-based and knowledge-based measures of text semantic similarity. In *Proceedings of the 21st National Conference on Artificial Intelligence*, pages 775–780, 2006.
14. Philip Resnik. Using information content to evaluate semantic similarity in a taxonomy. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, IJCAI'95, pages 448–453, San Francisco, CA, USA, 1995.
15. Michael Roth and Anette Frank. Automatically identifying implicit arguments to improve argument linking and coherence modeling. In *Proceedings of the Second Joint Conference on Lexical and Computational Semantics (*Sem-2013)*, Atlanta, GA, June 2013.

16. Bahar Salehi and Paul Cook. Predicting the compositionality of multiword expressions using translations in multiple languages. In *Proceedings of the Second Joint Conference on Lexical and Computational Semantics (*Sem-2013)*, Atlanta, GA, June 2013.
17. H. Andrew Schwartz, Johannes Eichstaedt, Lukasz Dziurzynski, Eduardo Blanco, Margaret L. Kern, Stephanie Ramones, Martin Seligman, and Lyle Ungar. Choosing the right words: Characterizing and reducing error of the word count approach. In *Proceedings of the Second Joint Conference on Lexical and Computational Semantics (*Sem-2013)*, Atlanta, GA, June 2013.
18. Ekaterina Shutova. Metaphor identication as interpretation. In *Proceedings of the Second Joint Conference on Lexical and Computational Semantics (*Sem-2013)*, Atlanta, GA, June 2013.
19. Peter D. Turney. Mining the web for synonyms: Pmi-ir versus lsa on toefl. In *Proceedings of the 12th European Conference on Machine Learning*, pages 491–502. Springer-Verlag, 2001.
20. Zhibiao Wu and Martha Stone Palmer. Verb semantics and lexical selection. In James Pustejovsky, editor, *ACL*, pages 133–138. Morgan Kaufmann Publishers / ACL, 1994.

Clasificación de frases obscenas o vulgares dentro de tweets

Estefania Guzmán, Beatriz Beltrán, Mireya Tovar,

Andrés Vázquez, Rodolfo Martínez

Benemérita Universidad Autónoma de Puebla
Facultad de Ciencias de la Computación
Avenida San Claudio, 14 Sur, Ciudad Universitaria
Puebla, México
fanyguzm@gmail.com, {bbeltran, mtovar, andrex}@cs.buap.mx,
beetho@gmail.com
http://nlp.cs.buap.mx

Resumen. Se propone una metodología para la detección frases obscenas y vulgares en los tweets, debido a que México es uno países donde se usa mucho el doble sentido para comunicarse. La metodología propuesta se apoya en un diccionario de mexicanismos etiquetado manualmente por expertos. Se pudo detectar que las palabras obscenas y vulgares son las que más se emplean y los estados del país que más las utilizan. Además en base al diccionario, se logra clasificar un conjunto de tweets, dichos tweet fueron tomados por zonas geográficas de México, consideramos que dichas personas son mexicanas y por lo tanto pueden llegar a manejar el doble sentido. La intención de esto es poder tener un sistema que ayude a encontrar cuando una persona está utilizando el doble sentido dentro de algún texto corto, como pueden ser los tweets y además lograr hacer mapas de México que nos proporcione información los lugares donde se utilizan con mayor frecuencia el doble sentido.

Palabras clave: Palabras obscenas, palabras vulgares, albur, diccionarios, textos cortos.

1. Introducción

El uso del doble sentido en el habla del mexicano es algo común en todas las esferas sociales en mayor o menor grado. En algunas ocasiones es percibido sin intención por la misma vaguedad intrínseca del lenguaje; pero en otras, se utiliza de manera premeditada para provocar un efecto en el interlocutor, como un recurso humorístico o hasta ofensivo.

Además, en México el albur está en todas partes, en la comida, en los oficios, esto es, cualquier palabra cobra doble sentido y la combinación de verbos y sustantivos, acompañados de gestos y sonrisas, se convierten en un albur. Para los expertos, el uso

del doble sentido puede ser un buen método educativo, porque requiere de un amplio vocabulario, atención y destreza mental en pocos segundos [9].

Algunas personas no entienden un albur, y esto puede llevarlas a una confusión, lo cual en particular suele pasar en redes sociales, donde hay que saber distinguir que es lo que realmente las personas quieren expresar. Hoy en día uno de los principales modos de comunicación sobre todo entre los más jóvenes, es a través de twitter. Para los especialistas en procesamiento de Lenguaje Natural se hace interesante estudiar si en este tipo de mensajes se utiliza palabras consideradas como obscenas o vulgares y detectar cuales palabras son las más empleadas.

Según la RAE[1], se define como **vulgar** a frases, palabras o expresiones; las cuales son groseras u ordinarias, siendo catalogadas como impropias para personas cultas o educadas. Y se define como **obsceno** a frases, palabras o expresiones; las cuales son impúdicas u ofensivas al pudor y en la mayoría de los casos con relación al sexo.

Ambas se denotan como groserías y que no son apropiadas para comunicarse en un ámbito formal, y su aplicación es en el contexto informal para expresar dichos, albures, chistes, etc. Pero en resumen para diferenciar si una frase es vulgar u obscena se deberá identificar primero si es un tipo de ofensa al pudor, para descartar que sea obsceno y si no cuenta con ello se cataloga como vulgar

En la presente investigación se desarrollan dos diccionarios de frases vulgares y obscenidades, los cuales permitieron etiquetar un gran número de tweets identificando el doble sentido, para desarrollar posteriormente modelos de clasificación.

La estructura del artículo, se tiene de la siguiente manera, en la sección 2, se presenta el trabajo que se ha realizado en el área de identificación del doble sentido o humor. En la sección 3 se describe la propuesta de la metodología para la identificación de obscenidades o vulgaridades dentro de textos cortos. En la sección 4 se muestran algunos resultados obtenidos y finalmente las conclusiones que se obtienen hasta el momento.

2. Trabajo Relacionado

México se caracteriza por su riqueza lingüística y en ese aspecto los mexicanos saben aprovecharla para expresarse de muchas formas, desde la manera más inocente hasta la más vulgar, teniendo así muchas características de humor en cada situación que se le presenta. Sin embargo hay veces que es difícil identificar si es un cumplido o una burla. Hasta hace unos años identificar si se estaba usando el albur en textos a través de herramientas computacionales no existía, era más fácil la generación que la detección.

El albur es un acto de someter a prueba la agilidad mental de alguien. Por definición éste nunca puede ser inocente, pero sí involuntario, pues se define como un juego de palabras que lleva implícito o explícito una connotación sexual que se puede desprender de cualquier conversación [10]. El albur es un juego de palabras con doble sentido. Este juego de palabras emplea principalmente dos elementos verbales. El

[1] http://www.rae.es/

primer elemento es la asociación de palabras y expresiones aparentemente inocuas con términos sexuales y/o escatológicos. El segundo elemento que se emplea generalmente en el albur es la deconstrucción de palabras inocuas para crear palabras o expresiones completamente distintas, pero que implican un mensaje sexual o escatológico [3].

A través de los años han surgido varios trabajos utilizando Procesamiento de Lenguaje Natural (PLN), para clasificar y recuperar información, sin embargo no se ha realizado ningún sistema el cual sea capaz de detectar el albur de un texto en español. Existen trabajos que han utilizado herramientas como WEKA[2] para definir si textos cortos contienen albur o no [2].

Ocampo en [2], utiliza un diccionario que contiene alrededor de 300 palabras para detectar albur y humor en textos como chistes y rimas, en el idioma español. Para la clasificación de su modelo desarrollado utiliza la herramienta WEKA.

La mayoría de los trabajos han sido desarrollados por Rada Mihalcea, para el idioma inglés [1], [4], [6]. Por ejemplo, Rada y Attardo en [1] hacen uso de características de estilo de textos humorísticos tales como la aliteración, la antonimia y el contenido adulto; así como de características basadas en contenido y de una combinación de ambas características, para hacer una clasificación.

Los resultados obtenidos en los experimentos de clasificación automática muestran que los enfoques computacionales representan una solución viable para la tarea de reconocimiento de humor, que es posible utilizar estas técnicas para distinguir entre textos humorísticos y no humorísticos, y que se alcanza buen desempeño usando técnicas de clasificación basadas en características de contenido y de estilo.

Así mismo, Mihalcea y Pulman en [7] analizan de forma detallada otras dos características que, según teorías psicológicas, están presentes de manera muy frecuente en el humor: centrado en el humano y la polaridad negativa.

Pero uno de los trabajos más innovadores es el de Rada Mihalcea *et al*. [5], [8], en donde exploran diversos modelos computacionales para la resolución de incongruencias, que es una de las teorías del humor más ampliamente aceptadas; la cual sugiere que el humor se debe a la mezcla de dos cuadros opuestos de interpretación posible para un enunciado.

Hasta el momento se han obtenido resultados significativos en el desarrollo de herramientas computacionales para el idioma inglés, sin embargo para el idioma español, el avance ha sido muy lento. En el presente trabajo solamente se intenta detectar si en ciertos tipos de mensajes en redes sociales se utiliza el doble sentido, para en una etapa posterior desarrollar modelos de clasificación.

3. Confección del corpus

Se formó un corpus de 548,243 tweets a través de un programa realizado en Python y usando el API del Twitter. El proceso de generación del corpus consideró tomar todos aquellos tweets publicados durante un cierto dia y que estuviesen localizados en un

[2] http://www.cs.waikato.ac.nz/ml/weka/

radio no mayor a 10 kilómetros de la capital de cada estado de la república mexicana (geolocalizados). Un análisis del corpus indica que se tiene un promedio de 14.340 palabras por tweet, con un vocabulario total de 581,109 palabras y en el cual interactuaron 173,339 usuarios con un promedio de 3.167 tweets por usuario, esto se puede ver en la Tabla 1.

Tabla 1. Información del corpus de tweets por estado y total

Estado	Tweets	Mínimo de palabras	Máximo de palabras	Promedio de palabras	Vocabulario	Tokens
AGU	14,426	1	32	11.452	23,351	165,200
BCN	13,644	1	33	11.064	23,563	150,961
BCS	11,145	1	30	10.897	18,813	121,443
CAM	15,496	1	36	11.201	22,671	173,569
CHH	12,652	1	39	11.576	21,293	146,458
CHP	12,440	1	31	10.980	19,656	136,590
COA	13,581	1	34	11.667	21,898	158,445
COL	14,498	1	32	12.281	24,728	178,054
DIF	13,452	1	32	11.857	26,305	159,500
DUR	15,837	1	33	11.643	24,401	184,397
GRO	11,012	1	32	10.748	18,502	118,353
GUA	15,021	1	32	11.136	24,385	167,271
HID	14,304	1	33	10.874	21,349	155,546
JAL	14,201	1	31	11.693	25,394	166,056
MEX	13,208	1	36	11.767	25,311	155,413
MIC	14,643	1	31	11.186	23,104	163,797
MOR	12,288	1	32	11.439	21,242	140,560
NAY	12,892	1	33	11.932	22,854	153,828
NLE	14,354	1	33	10.455	22,999	150,072
OAX	11,159	1	36	12.333	19,221	137,623
PUE	13,415	1	33	11.092	21,143	148,797
QRO	16,274	1	32	10.372	23,390	168,798
ROO	14,369	1	33	11.974	21,975	172,055
SIN	16,169	1	32	10.814	23,518	174,858
SLP	16,473	1	32	11.676	26,194	192,341
SON	12,856	1	31	9.210	17,653	118,405
TAB	15,459	1	31	9.969	21,906	154,117
TAM	16,183	1	36	10.156	21,567	164,360
TLA	12,225	1	33	11.168	20,478	136,525
YUC	12,976	1	31	10.924	22,326	141,755
ZAC	15,394	1	32	11.798	24,138	181,614
TOTAL	548,243	1	39	12.340	581,109	6,765,487

A partir del diccionario de mexicanismos, el cual viene con una clasificación de uso, se tomaron los que tienen la categoría de VULG (vulgaridades) y OBSC (obscenidades), refiriéndose a palabras vulgares y obscenas respectivamente, creando así 2

diccionarios, uno de vulgaridades y el otro de obscenidades, con 409 y 361 frases respectivamente. Ejemplos de ambos diccionarios se muestran en la Tabla 2.

Tabla 2. Ejemplos de obscenidades y vulgaridades

Obscenidades	@Maracucho_Serio tweeted: RT @Manuelovsky: Dejarse **coger** no duele, lo que duele es enamorarse. @sswwar tweeted: Upsss **se me para** la verga http://t.co/I7WCUifjaO @boulalaAlex tweeted: Dejarme el bigote es lo más cercano a una **panocha** que he tenido en la cara. @Wilherreraleon tweeted: RT @weezysanchez: Vine a **calzonear** a las embajadoras!!! Raroooooooooo @ChicoCheVive @jorgedelacruz99 tweeted: "@Javii_Noble: Una Princesa bien amada, bien valorada y bien **cogida**, jamás te será infiel. #YoDigo #JaviTip"
Vulgaridades	@ChechoV tweeted: Ya saquen la caguama ps que está **de la chingada** el calor @lVICOCENTENO tweeted: Cuando es amor de verdad, no hay ni puta, ni **pendejo** que los separe... @raulcamangulo tweeted: Que gustarles que las anden manoseando entre el **chingamadral** de gente en el carnaval @iJosee_ tweeted: Pues dicen que no te sirve de nada estar **mamado** si estas feo... #NomasTeInformo @patycarretell0 tweeted: Me tienen hasta la madre con sus selfies

4. Análisis del corpus desarrollado

Se realizó un análisis estadístico sobre los tweets y en base a los diccionarios de vulgaridades y obscenidades se obtuvieron los siguientes resultados: Se realiza análisis por estado para revisar dentro de los tweets que estados tienen dentro de su vocabulario mayor cantidad de vulgaridades, resultando así el estado de Querétaro (QRO), seguido de Tabasco (TAB) y Nuevo León (NLE), y en último lugar se posiciona Sinaloa (SIN), como se puede observar en la Tabla 3, además en la tercer columna de la Tabla 4 se observa que el estado que resulta con mayor ocurrencias, es uno de los que utilizan las palabras con mayor frecuencia.

Tabla 3. Frecuencia de ocurrencia de vulgaridades por estado

Estado	Ocurrencias	Estado	Ocurrencias
QRO	408	MOR	247
TAB	350	MIC	240
NLE	347	ZAC	230
PUE	345	MEX	225
HID	325	YUC	224
TAM	319	GRO	220

CHP	311	DF	217
VER	306	BCN	211
GUA	305	ROO	203
AGU	305	COA	182
CHH	303	COL	181
TLA	284	NAY	179
CAM	275	OAX	171
SON	264	BCS	167
SLP	260	DUR	166
JAL	254	SIN	102

Además se pueden obtener las cinco primeras palabras o frases vulgares que dentro de todos los tweets obtuvieron mayor ocurrencia, las cuales se pueden mostrar en la Fig. 1, y que fueron tomadas a partir de la Tabla 4, donde se muestra los estados que más utilizan dicha palabra o frase.

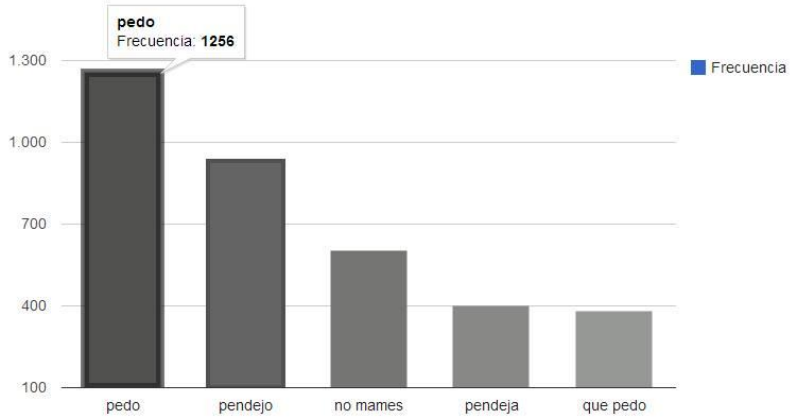

Fig. 1. Palabras o frases vulgares con mayor frecuencia.

Tabla 4. Palabras vulgares con mayor ocurrencia en total.

Frase	Ocurrencias	Estados
pedo	1256	**QRO**, TAB, PUE
pendejo	935	**QRO**, CHH, NLE
no mames	601	HID, TAB, TAM
pendeja	401	CAM, ZAC, HID
que pedo	382	PUE, **QRO**, CHH

Posteriormente, se realiza un análisis para cada estado, con el objetivo de obtener dentro de los tweets que estados manejan dentro de su vocabulario mayor cantidad de obscenidades, en las tres primeras posiciones se encuentra Chihuahua (CHH), Zacatecas (ZAC) y Guanajuato (GUA), respectivamente, y en último lugar se posiciona Sonora (SON).

Se obtienen las cinco primeras palabras o frases obscenas que dentro de todos los tweets obtuvieron mayor ocurrencia las cuales se muestran en la Fig. 2, éstas fueron tomadas a partir de la Tabla 5 y donde se pueden observar los estados que más utilizan dicha palabra o frase.

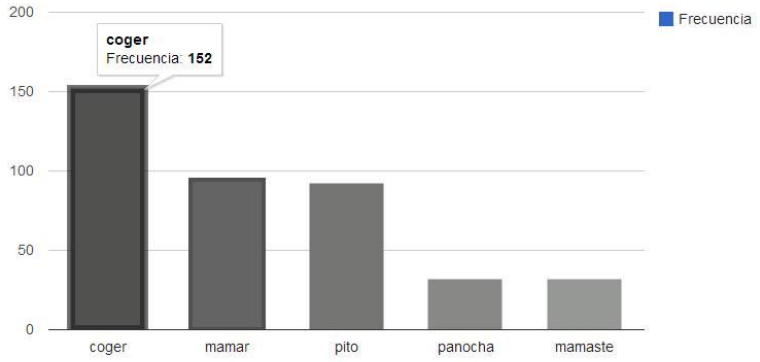

Fig. 2. Palabras o frases obscenidades con mayor frecuencia.

Tabla 5. Palabras obscenas con mayor ocurrencia en total.

Frase	Ocurrencias	Estados
coger	152	MEX, **CHH**, JAL
mamar	95	DF, QRO, VER
pito	92	MIC, **CHH**, NAY
panocha	32	**CHH**, GUA, COL
mamaste	32	**CHH**, TAM,COA

5. Modelo de clasificación

El análisis del corpus y la detección de las palabras obscenas y vulgares más utilizadas en las diferentes regiones, ayudó a categorizar los tweets que forman parte del corpus de entrenamiento. Es decir si el tweet contiene palabras obscenas, dicha frase es etiquetada como **obscenidad**, si el tweet contiene palabras vulgares, se etiqueta como **vulgaridad**, en el caso del que tweet contenga tanto palabras obscenas como vulgares se etiqueta como **mezclado**, y si no contiene ninguna palabra obscena o vulgar se etiqueta como **ninguno**. El conjunto de datos de entrenamiento quedó completamente desbalanceado.

Se realizó un proceso de balanceo quedando la misma cantidad de tweets correspondiente a la clase menos representativa en dicho conjunto. La selección de los tweets que forman parte del conjunto de entrenamiento se realizó de manera aleatoria, quedando al final 130 tweet por cada categoría.

Una vez desarrollado el corpus de entrenamiento, se utilizaron para el desarrollo del modelo de clasificación todas las palabras que aparecen en cada tweets. No se aplicó ningún preprocesamiento al corpus de entrenamiento construido.

Para el desarrollo de los modelos de clasificación se utilizó la herramienta Weka. En particular los algoritmos de clasificación empleados fueron: Vecino más cercano (IBK), Máquina de soporte vectorial (SMO), Bayes Multinomial y Árbol de decisión (J48).

En la Tabla 6 se pueden ver los resultados obtenidos, aplicando validación cruzada con 10 pliegues. Como se puede observar el algoritmo J48 es el que nos brinda mejores resultados con un 76.35%.

Tabla 6. Clasificación de tweets con cuatro clases

Algoritmo	**Correctamente clasificadas**	**Incorrectamente clasificadas**
IBK	50.77%	49.23%
SMO	75%	25%
Bayes multinomial	69.81%	30.19%
J48	**76.35%**	23.65%

Como el número de muestras positivas de cada clase es pequeño, se puede apreciar que los resultados en el modelo de clasificación no fueron muy altos, por este motivo se decidió desarrollar entonces dos corpus de entrenamiento, considerando solamente dos categorías: Obscenidad y Ninguna y Vulgaridad y Ninguna.

Las Tablas 7 y 8, muestran los resultados obtenidos aplicando validación cruzada con 10 pliegues.

Tabla 7. Clasificación con las clases Obscenidad y Ninguna

Algoritmo	**Correctamente clasificadas**	**Incorrectamente clasificadas**
IBK	70.53%	29.47%
SMO	**91.07%**	8.93%
Bayes multinomial	85.14%	14.86%
J48	84.28%	15.72%

Tabla 8. Clasificación con las clases Vulgaridad y Ninguna

Algoritmo	**Correctamente clasificadas**	**Incorrectamente clasificadas**
IBK	75.04%	24.96%
SMO	**98.90%**	1.10%
Bayes multinomial	88.54%	11.46%
J48	97.46%	2.54%

Como puede apreciarse los mejores resultados fueron ofrecidos por el algoritmo de clasificación máquina de soporte vectorial, con los datos por defecto que ofrece la herramienta Weka.

El hecho de que haya mejorado la precisión del modelo de clasificación se debe en gran medida al número de muestras positivas en cada clase, y por el hecho de que es más simple para el algoritmo de clasificación cuando se trabaja con 2 clases, que con 4 clases.

6. Conclusiones

En este trabajo se presenta una primera aproximación para la identificación de frases obscenas y vulgares en mensajes de twitter. Se puede observar que los mensajes de este tipo se emplean más en los estados de: Guanajuato, Estado de México y Jalisco para obscenidades y en el caso de vulgaridades los estados de la republica detectados que usan este tipo de frases son: Querétaro, Puebla e Hidalgo.

Desarrollando los diccionarios de vulgaridades y obscenidades se pudo de alguna manera construir un corpus de entrenamiento para posteriormente desarrollar un modelo de clasificación. A pesar de que el corpus no quedó en primera instancia balanceado, y el proceso de balanceo fue muy simple, los resultados son relativamente alentadores. En una segunda etapa se va a realizar un proceso de balanceo más adecuado.

En estos momentos se está preparando un corpus de prueba, para validar realmente el modelo de clasificación construido.

Como ha sido la primera aproximación, no se ha realizado ningún pre procesamiento a los tweets, aspecto que se está considerando, buscando otro modelo de representación de los mismos, ya que solamente se utilizó bolsa de palabras.

El corpus y el modelo de clasificación desarrollado es nuestro primer acercamiento para intentar educar a las nuevas generaciones en el no empleo desmedido de esta forma de comunicación, así mismo puede servir de apoyo en los sitios web, para que no permitan comentarios de usuarios sin previa revisión.

Referencias

1. Mihalcea, R., Attardo, S.: Making Computers Laugh. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing* (HLT/EMNLP), pages 531–538, Vancouver, October 2005.
2. Ocampo, R.: Detección automática de humor en textos cortos en español. *Tesis*: Instituto Politécnico Nacional (2010).
3. Hernández, V.: Antología del Albur. *BookSurge Publishing*, Charleston SC. (2006).
4. Mihalcea, R., Strapparava, C.: Making Computers Laugh: Investigations in Automatic Humor Recognition. In *Human Language Technology Conference and Con-*

ference on Empirical Methods in Natural Language Processing. Vancouver, October 2005. Association for Computational Linguistics. 531–538, (2005).
5. Mihalcea, R., Strapparava, C.: Laughter Abounds in the Mouths of Computers: Investigations in Automatic Humor Recognition. In INTETAIN 2005, LNAI 3814, pp. 84–93, Springer-Verlag Berlin Heidelberg (2005).
6. Mihalcea, R., Strapparava, C.: Learning to Laugh (Automatically): Computational Models for Humor Recognition. Computational Intelligence. 22. 126-142 (2006).
7. Mihalcea, R., Pulman, S.: Characterizing Humor: An exploration of Features in Humorous Texts. : CICLing 2007, LNCS 4394, pp. 337–347. Springer-Verlag Berlin Heidelberg, (2007).
8. Mihalcea, R., Strapparava, C., Pulman, S.: Computational Models for Incongruity Detection in Humour. : CICLing 2010, LNCS 6008, pp. 364–374. Springer-Verlag Berlin Heidelberg (2010).
9. García Torres Carmen C.: El doble sentido en la conversación de llamadas telefónicas a programa de radio: análisis semiótico-discursivo. Proyecto de Tesis Universidad Autónoma de Nuevo León (2010).
10. Espinosa Luz: El albur: el mete y saca de la palabra. http://culturacolectiva.com/el-albur-el-mete-y-saca-de-la-palabra/ (visitada: 25/10/2014)

Identificación automática del canal de aprendizaje en mensajes cortos

Rocío Galaviz, Beatriz Beltrán, Andrés Vázquez, Mireya Tovar

Benemérita Universidad Autónoma de Puebla
Facultad de Ciencias de la Computación
Avenida San Claudio, 14 Sur, Ciudad Universitaria
Puebla, México
chio_gus@hotmail.com,{bbeltran, andrex, mtovar}@cs.buap.mx
http://nlp.cs.buap.mx

Resumen. En el presente trabajo se proponen varios modelos de clasificación para detectar el tipo de aprendizaje de un individuo a partir de los tweet que escribe. Este modelo se apoya en los tres tipos de canal de aprendizaje propuesto en la programación neurolingüística (visual, auditivo y kinestésico). 4 expertos desarrollaron un corpus de entrenamiento, clasificando de manera manual a 1000 tweets. Se desarrollaron diversos experimentos considerando diferentes modelos de representación y varios clasificadores. Se realizó validación cruzada de 10 pliegues y los mejores resultados fueron ofrecidos por el clasificador máquina de soporte vectorial.

Palabras clave: Canal de Aprendizaje, modelos de clasificación, representación por medio de *n*-gramas, Programación Neurolingüística.

1. Introducción

Durante años se ha trabajado para mejorar el nivel educativo en México, se han implementado diversas propuestas metodológicas para mejorar el proceso enseñanza-aprendizaje y los resultados obtenidos aun en la actualidad no son los esperados, sería conveniente que se ofrezcan herramientas automáticas que permitan detectar la forma en que los alumnos aprenden.

La Programación Neurolingüística (PNL) constituye un modelo, formal y dinámico de cómo funciona la mente y la percepción humana, cómo se procesa la información y la experiencia, y las diversas implicaciones que esto tiene para el éxito personal. Con base a este conocimiento es posible identificar las estrategias internas que utilizan las personas de éxito, aprenderlas y enseñarlas a otros (modelar); para facilitar un cambio evolutivo y positivo. La Programación Neurolingüística, por analogía con los sistemas de cómputo, utiliza los patrones universales de comunicación y percepción que tenemos para reconocer e intervenir en procesos diversos (aprendizaje, terapia, afrontamiento del estrés, negociación, gestión de conflictos, superación de fobias, etc.). El campo de trabajo es tan amplio como el de las relaciones interpersonales [1].

Dentro del área de la educación, los tipos de aprendizaje que la PNL maneja son 3:

- **Visual**, son aquellos que comprenden y recuerdan más fácilmente la información si se les da por escrito. Para estudiar, a estas personas, les funciona muy bien hacer mapas conceptuales, buscar imágenes con que relacionar la información, escribir y leer apuntes.
- **Auditivo**, les es más fácil escuchar la información que leerla. Por lo general si toman apuntes se pierden, prefieren poner mucha atención a lo que escuchan y tienen una gran habilidad para recordar conversaciones. Su método de estudio puede implicar estudiar en voz alta, pedir a alguien que les explique de manera verbal o bien inventar rimas, canciones o juegos de palabras para memorizar.
- **Kinestésico**, aprenden a través del contacto directo con los contenidos y las emociones. Pueden fácilmente relacionar los contenidos con el estado de ánimo que les producen, estudian mejor en ambientes confortables y conocidos, muchos de ellos estudian mejor en compañía y optan por visitar museos y experimentar los contenidos [2].

En [6] se plantea que la PNL es un instrumento poderoso para reprogramar nuevamente la mente e introducirle información acerca de todos los principios que rigen una mala experiencia, para así desbloquear esas falsas creencias que se arraigaron en la mente y que no permiten tomar las acciones que conlleven a los resultados que deseamos.

La PNL parte de la teoría constructivista, que define la realidad como una invención y no como un descubrimiento. Se basa en el hecho de que el ser humano no opera directamente sobre el mundo real en que vive, sino que lo hace a través de representaciones mentales del mismo, que determinan la forma en que cada individuo percibe el mundo. Proporciona herramientas y habilidades para el desarrollo en comunicación y reprogramación de actitudes. Promueve la flexibilidad del comportamiento, la creatividad y la comunicación, el pensamiento trascendental y una comprensión de los procesos mentales, tanto para el desarrollo individual como para la optimización de cada grupo humano.

Existen diversos test en la web que normalmente manejan los psicólogos para detectar el canal de aprendizaje, sin embargo no son tan confiables, pues estos pueden ser alterados por los alumnos al responder. Es más conveniente detectar el canal de aprendizaje dentro de una plática común o incluso cuando el estudiante redacta algo que sale de su imaginación, como un cuento o la explicación de algún texto. Por lo que, se propone desarrollar una metodología que se pueda aplicarse al sector educativo y que le brinde al profesor las herramientas tecnológicas donde él pueda clasificar a los alumnos según su estilo de aprendizaje, así sabría cómo dirigirse de una forma más asertiva y qué métodos puede aplicar para mejorar el aprovechamiento del mismo.

La estructura del artículo, se tiene de la siguiente manera, en la sección 2, se presenta los fundamentos teóricos de la programación neurolingüística; en la sección 3, se presenta el trabajo que se ha realizado para la detección del canal de aprendizaje. En la sección 3 se describe la propuesta de metodología para la identificación del canal de aprendizaje dentro de textos cortos. En la sección 4 se muestran algunos resultados obtenidos y finalmente las conclusiones que se obtienen hasta el momento.

2. Trabajo Relacionado

Dentro de la investigación realizada no se encontró ningún sistema que mediante el análisis de un texto de manera automática detecte el estilo de aprendizaje y que esté basado en la PNL, sin embargo se encontraron algunos sistemas con temas relacionados, los cuales se discuten a continuación:

Dentro del trabajo de tesis desarrollado en [3] se propone un sistema de apoyo para el aprendizaje de lecto-escritura para niños, su objetivo fue la realización de un módulo de dictado y corrección de ortografía, programado de una manera flexible para su integración a la arquitectura Galaxy como parte de un sistema tutorial de lecto-escritura.

Por otro lado, el trabajo desarrollado en [4] se propone un sistema para el fortalecimiento del pensamiento matemático dirigido a niños de preescolar. El software propuesto se conforma de dos juegos que cubren distintas áreas del pensamiento matemático. Se utilizan los bloques mágicos para abordar principalmente el tema de número y el bloque denominado Circolar que se enfoca al tema de forma y figuras geométricas. Con la utilización del sistema en el proceso de enseñanza se pudo validar que los alumnos lograron de manera correcta apropiarse de los conceptos matemáticos involucrados.

Otra tendencia actual ha sido introducir las plataformas de aprendizaje en las diferentes materias que forman parte de la curricula a nivel básico y medio superior, sin embargo esto se ha visto limitado por las condiciones particulares de cada escuela. Otra tendencia ha sido el desarrollo de entornos virtuales de enseñanza-aprendizaje para la enseñanza en el nivel superior. En particular en [5] se desarrolló una plataforma de este tipo a través de la Web y que maneja 3 perspectivas:

- Perspectiva social y pedagógica, trabajando sobre aprendizaje colaborativo
- Perspectiva tecnológica, diseñar módulos para la plataforma educativa que permitan la configuración y la realización de experiencias con grupos de alumnos con tecnología basada en Internet y WWW.
- Perspectiva experimental, diseñar, configurar y realizar experiencias a distancia en entornos reales.

Como se comentó al inicio hasta el momento no se ha desarrollado ninguna herramienta tecnológica de apoyo al docente que le permita detectar de una manera adecuada la forma en que su grupo de trabajo aprende, en este sentido está encaminada la presente investigación.

A continuación se discute el modelo de clasificación desarrollado para detectar el canal de aprendizaje de un individuo.

3. Fundamentos teóricos de PNL

La PNL parte de los fundamentos de la teoría constructivista, la cual define la realidad como una invención y no como un descubrimiento. Es un constructo psíquico de Grinder y Bandler con base en el hecho de que el ser humano no opera directamente

sobre el mundo real en el que se vive, sino que lo hace a través de mapas, representaciones, modelos a partir de los cuales se genera y guía su conducta. Estas representaciones, que además determinan el cómo se percibirá el mundo y qué elecciones se entenderán como disponibles, difieren necesariamente de la realidad de la cual representan. Esto se debe a que el ser humano, al transmitir su representación del mundo, tiene ciertas limitaciones, que se derivan de las condiciones neurológicas del individuo, de la situación social en que se vive y de sus características personales.

Estas limitaciones determinan que el lenguaje, como toda expresión humana, esté también sometido a ciertos procesos que empobrecen su modelo. Para el abordaje de estos modelos, la PNL se apoya en el modelo de la gramática transformacional que se basa en la creación de nuevos mensajes; considera que las personas son capaces de interpretar y producir mensajes nuevos, de manera que puede afirmarse que no se produce por repetición ni por recuerdo [7].

Se considera la PNL como:

- Una herramienta que ayuda a mejorar el aprendizaje
- Un esquema general práctico cuya finalidad es sernos útil, aumentar las opciones y conocernos mejor.

La PNL ayuda a entender como los seres humanos estructuramos las experiencias individuales de la vida; representa una actitud de la mente y una forma de ser en este mundo.

Desde 1976 los autores originales llaman PNL a su trabajo original [8]:

- **Programación** porque se refiere a un proceso organizador de componentes de un sistema para obtener un resultado esperado; a los modos que podemos escoger para organizar nuestras ideas (pensamientos-emociones) y acciones.
- **Neuro** porque se parte de todo comportamiento proviene de procesos del sistema nervioso (los sentidos y demás expresiones neurológicas) mediante los cuales hacemos contacto con el mundo y damos sentido a la información, actuando en consecuencia.
- **Lingüística** porque usamos el lenguaje para ordenar nuestras ideas y comunicarnos con los demás, como un recurso que tiene una representación, orden y secuencia que manifestamos.

La comunicación está formada por un conjunto de factores (gestos, posturas, palabras, tono de voz). Esta manera de comunicarnos viene determinada por nuestro mapa del mundo; es decir, nuestra manera de percibir y organizar la realidad. Y este proceso se da a través de nuestros sentidos. En la PNL las maneras como recogemos, almacenamos y codificamos la información en nuestra mente –ver, oír, gustar y oler- se conoce con el nombre de sistemas representativos o canales de aprendizaje:

- Visual (imágenes)
- Auditivo (sonidos, ritmo)
- Cinestésico o kinestésico (sensaciones, gusto y olfato)

Cuando una persona tiende a utilizar, de manera habitual, un sentido de forma interna, se dice que éste es su sistema preferido o primario, el cual se refleja tanto en el vocabulario que utilizamos como en las posturas, tono de voz, gestos con los que acompañamos las palabras [7].

4. Descripción de la propuesta

Se confeccionó un corpus de entrenamiento conformado por 1,000 tweets considerando una muestra de 676 usuarios diferentes con un promedio de 1.479 tweets cada uno. Se desarrolló un diccionario de emoticones para normalizar los tweets, además cada usuario es sustituido por la etiqueta USER_TAG y los *hash-tag* por una etiqueta similar. Las características del corpus se muestran en la Tabla 1 y Tabla 2.

Tabla 1. Información del corpus de tweets

Tweets	Mínimo de palabras	Máximo de palabras	Promedio de palabras	Vocabulario	Tokens
1,000	1	32	11.757	4,215	11,757

Tabla 2. Información de usuarios dentro del corpus

Mínimo de tweets por usuario	Máximo de tweets por usuario	Promedio de tweets por usuario	Total de usuarios
1	16	1.479	676

En el proceso manual de clasificación se propone utilizar un total de 4 clases. La categoría **neutro**; es asignada si en el tweet no existen los elementos suficientes para identificar el canal de aprendizaje. La categoría de **auditivo** es asignada cuando a través del texto se detecta que aprende por medio de lo que escucha. La categoría **visual** es detectada si en el tweets se determina que su canal de aprendizaje es a través de lo que observa, por último si aprende por medio de lo que toca se dice que ese individuo es **kinestésico**. La cantidad de tweets por clase se muestra a continuación en la Tabla 3.

Tabla 3. Tweets por clase

Clase	Tweets por clase
Neutro	577
Auditivo	64
Visual	144
Kinestésico	267

Tabla 4. Ejemplos de tweets de cada clase

Neutro	USER_TAG que tengas buen dia HAPPY_TAG
	USER_TAG eee gracias * le da, una buena propina
	I am at La Casa De Don Porfirio (Atlixco, Puebla) URL_TAG
	Una niña vale más que veinte niños HAPPY_TAG
	USER_TAG Visita la Capilla del Rosario en Recorrido Virtual HASH_TAG URL_TAG
Auditivo	Ya los acabo de escuchar, muy buenos USER_TAG USER_TAG
	Por qué HOY? Por que HOY? ya estaba decidido a que mañana hablaría las cosas por HOY pasa esto! PUTAAAAAAAMAAAADREEEEEE! La cagas
	escucha una canción o reconoce en un desconocido un gesto, y todo se vuelve a abrir, todo regresa, como una ola, como una bofetada
	HASH_TAG Ella dijo que no le hable y no le hablare
	Decir que un alto cociente intelectual no hace superior a nadie es aceptar que se tiene uno bajo
Visual	algo no me cuadra... CONOCISTE A OTRA PERSONA?
	RT USER_TAG El uso de zapatos blancos debería ser ilegal
	Acabo de publicar una foto URL_TAG
	Típico: hacer enojar a tu hermana, recordar que te puede acusar con tu mamá y la haces en contentar
	USER_TAG me deberías hacer así lindura
Kinestésico	RT USER_TAG En domingo siempre me llega la nostalgia. URL_TAG
	RT USER_TAG No es fácil olvidar a alguien que fue muy importante en tu vida
	USER_TAG jaja de hecho no, expresé una disculpa y amablemente expresé mi opinión sobre el tema. TONGUE-TAG
	No tengo celos, pero tengo un rifle
	Yo por ti dejo lo que tú me pidas yo te doy mi vida entera te la doy y te prometo tu serás la reyna de mi fortaleza y de mi corazón

Con estos datos de entrenamiento se desarrollan varios modelos de clasificación. En particular se utilizan los clasificadores Naive-Bayes Multinomial, k-vecinos más cercanos (IBK), máquina de soporte vectorial (SMO) y árbol de decisión (J48), ofrecidos por la herramienta WEKA[1] .Se realizó validación cruzada con 10 pliegues, los resultados obtenidos por los 4 clasificadores se muestran en la siguiente sección.

5. Resultados

Para la representación de los tweets se utilizó un modelo de *n*-gramas, donde *n* va del valor 1 hasta el valor 3. Se desarrollaron varios experimentos considerando la frecuencia de cada *n*-grama en el rango de 1 a 3. Los resultados obtenidos se muestran en la Tabla 5.

[1] http://www.cs.waikato.ac.nz/ml/weka/

Tabla 5. Resultados utilizando diferentes clasificadores

Clasificador	Mín frec	*n*-gramas	Instancias Correctas	Instancias Incorrectas	*n*-gramas	Instancias Correctas	Instancias Incorrectas
BAYES-MULTINO-MIAL	1	1-3	**60.9316%**	39.0684%	2-3	60.0760%	39.9240%
IBK	1	1-3	59.3156%	40.6844%	2-3	59.3156%	40.6844%
SMO	1	1-3	60.3612%	39.6388%	2-3	**60.4563%**	39.5437%
J48	1	1-3	55.6084%	44.3916%	2-3	57.6996%	42.3004%
BAYES-MULTINO-MIAL	2	1-3	60.3612%	39.6388%	2-3	57.4144%	42.5856%
IBK	2	1-3	55.038%	44.962%	2-3	58.2700%	41.7300%
SMO	2	1-3	**61.0266%**	38.9734%	2-3	59.9810%	40.0190%
J48	2	1-3	56.0837%	43.9163%	2-3	57.0342%	42.9658%
BAYES-MULTINO-MIAL	3	1-3	58.9354%	41.0646%	2-3	56.3688%	43.6312%
IBK	3	1-3	51.711%	48.289%	2-3	54.6578%	45.3422%
SMO	3	1-3	**60.7414%**	39.2586%	2-3	**58.4601%**	41.5399%
J48	3	1-3	55.9886%	44.0114%	2-3	57.5095%	42.4905%

Los mejores resultados con un 61.0266% de instancias correctamente clasificadas, fueron obtenidos utilizando el clasificador máquina de soporte vectorial, donde tanto para los unigramas, los bigramas y los trigramas se tiene una frecuencia mínima de 2. La *F1* muestra su valor máximo para la clase neutra, y su mínimo valor para la clase auditiva, eso se muestra en la Tabla 6. Este resultado se corresponde totalmente con las características del corpus de entrenamiento, ya que está totalmente desbalanceado.

Tabla 6. Precisión, evocación y F1 para el clasificador SMO

	TP Rate	FP Rate	Precision	Recall	F1	ROC Area	Class
	0.828	0.366	0.733	0.828	0.778	0.751	NEUTRO
	0.125	0.031	0.205	0.125	0.155	0.652	AUDITIVO
	0.25	0.087	0.313	0.25	0.278	0.61	VISUAL
	0.449	0.161	0.488	0.449	0.468	0.644	KINESTESICO
Weighted Avg.	0.61	0.255	0.581	0.61	0.593	0.699	

Los resultados obtenidos en la Tabla 7 muestran que la mayor cantidad de tweets clasificados incorrectamente, son aquellos que corresponden a tweets asociados a individuos con aprendizaje kinestésico.

Tabla 7. Matriz de confusión para el clasficador SMO

A	B	C	d	Clasificado como		
478	7	29	63	a	=	NEUTRO
25	8	11	20	b	=	AUDITIVO
57	8	36	43	c	=	VISUAL
92	16	39	120	d	=	KINESTESICO

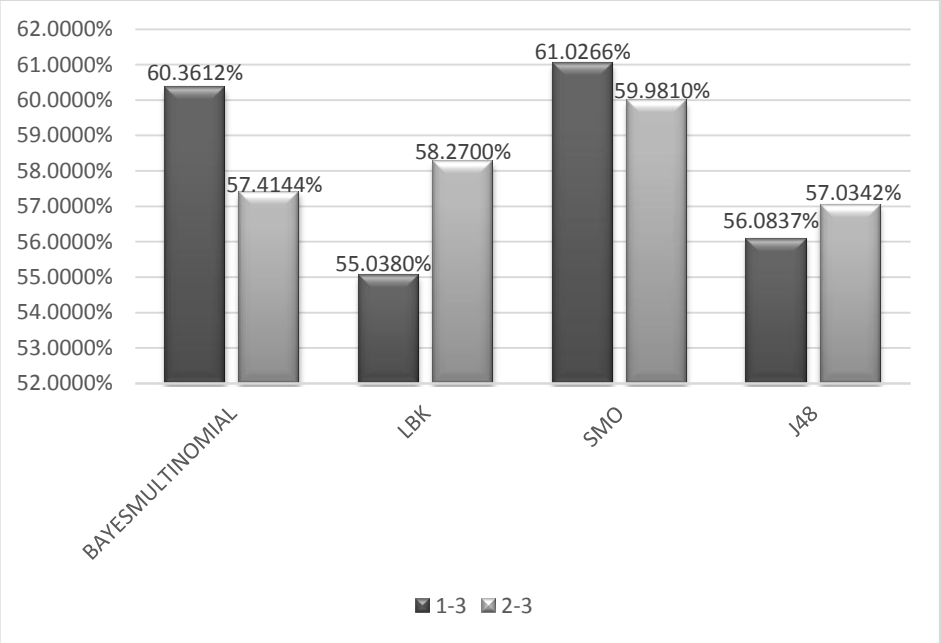

Fig. 1. Porcentajes para una mínima frecuencia de 2, *n*-gramas de 1-3 y de 2-3

En la Fig. 1 se muestran los porcentajes obtenidos por cada uno de los clasificadores, con una frecuencia mínima de 2 y para *n*-gramas de 1 a 3 y de 2 a 3.

6. Conclusiones

Como resultado de este trabajo de investigación se ha desarrollado un corpus de entrenamiento en el que han participado 4 expertos en el área de programación neurolingüística, lo que nos permitirá proponer nuevos modelos de aprendizaje automático. Los resultados obtenidos son completamente lógicos, ya que el corpus no está balanceado, y se sabe que esto afecta en el proceso de clasificación. Se pudo comprobar que de los 4 clasificadores utilizados el mejor comportamiento lo ofreció la máquina de soporte vectorial utilizando bolsa de *n*-gramas, sin embargo se considera adecuado utilizar **tf*idf**.

En estos momentos se está extendiendo el corpus de entrenamiento, buscando obtener tweets representativos de cada clase. Se están desarrollando modelos de entrenamientos considerando hasta quintigramas.

Referencias

1. Programación Neuroligüistica, http://es.wikipedia.org/wiki/Programación_neurolingüística (Visitada: 14/10/2014)
2. Canales del Aprendizaje, http://elpsicoasesor.com/canales-del-aprendizaje/ (Visitada: 14/10/2014)
3. Gutiérrez F.L.: Módulo de dictado para un sistema de apoyo para el aprendizaje de lecto-escritura para niños. Tesis Universidad de las Américas Puebla. (2002)
4. Peralta H.I.: KidMatex: Software Educativo para el fortalecimiento del pensamiento matemático dirigido a niños de preescolar. Tesis Universidad de las Américas Puebla. (2010)
5. Gil, G., Arias, D., Sanchez, E., Gimson,L. Romero, R., Ramirez, J., E., Silvera, J., Massé, M., Fernández, E.: Desarrollo de un entorno virtual de enseñanza-aprendizaje para la Universidad Nacional de Salta. In XII Workshop de Investigadores en Ciencias de la Computación. WICC 2010. Santa Cruz, Argentina (2010)
6. Cuscó M., Solanich, R.: Actividades de programación neurolingüística para educación primaria. Aula de innovación educativa (2012)
7. Peterson D.: PNL conoce cómo te puede apoyar. Editores Emu Mexicanos Unidos, México (2013).
8. O'Connor, J. & Seymour, J.: Introducción a la Programación Neurolingüística. Editorial Urano, Barcelona (1995)

Detección del lenguaje figurativo e ironía en textos cortos

Patricia Espinoza, Darnes Vilariño, Mireya Tovar, and Beatríz Beltrán

Benemérita Universidad Autónoma de Puebla,
Facultad de Ciencias de la Computación,
Avenida San Claudio, 14 Sur, Ciudad Universitaria.
Puebla México.
patricia.efong@gmail.mx
darnes,mtovar,bbeltran@cs.buap.mx
http://nlp.cs.buap.mx

Resumen En la presente investigación se propone un modelo para resolver la Tarea 11 de la Competencia Semeval 2015. El modelo propuesto utiliza características léxicas extraídas de los textos, asi como la polaridad de las palabras obtenidas utilizando diferentes herramientas. El modelo fue validado con 1 corpus de Twitter y se compara el desempeño de dos de los algorítmos más usados para clasificación: Naïve Bayes Multinomial y Máquinas de soporte vectorial.

Palabras clave: Análisis de sentimientos, twitter, polaridad

1. Introducción

Analizar la polaridad de los textos es una tarea importante para varias aplicaciones de la vida real. Existe un gran número de tareas que se ven beneficiadas por el desarrollo de métodos computacionales que permitan detectar la intención con que ha sido escrito un cierto mensaje, es decir, si el mensaje es positivo, negativo, neutro u objetivo. Una de las aplicaciones directas de ésto se aprecia cuando se desea construir la reputación de un político, de una celebridad o de un hombre de negocios que se puede hacer a partir del análisis de la polaridad de los mensajes en Internet, sin embargo esta tarea ha evolucionado del análisis de mensajes donde se usa un lenguaje literal, sin doble sentido, hasta detectar ese lenguaje figurativo o ironía en el texto, ya que si una opinión es clasificada como positiva existe la posibilidad de que sea irónica. En particular en este trabajo se presenta una comparación de los resultados obtenidos al desarrollar un modelo para la Tarea 11 del semeval 2015 [1]. La cual consiste en crear un sistema de análisis de sentimientos que nos ayude a identificar el grado de lenguaje figurativo e ironía implementada dentro de un mensaje. Dicho modelo fue probado con dos de los clasificadores más usados actualmente implementados en Weka[1]: Naïve Bayes Multinomial[2] y Máquinas de soporte vectorial (SMO)[3].

[1] http://www.cs.waikato.ac.nz/ml/weka/
[2] http://weka.sourceforge.net/doc.dev/weka/classifiers/bayes/NaiveBayesMultinomial.html
[3] http://weka.sourceforge.net/doc.dev/weka/classifiers/functions/SMO.html

La estructura del artículo es la siguiente. En la sección 2 se presentan los trabajos desarrollados en la literatura con respecto a la identificación de emociones en textos cortos. La sección 3 presenta la descripción de las características y el modelo de clasificación usando los experimentos desarrollados. La discusión acerca de los resultados obtenidos se presenta en la sección 4. Finalmente la conclusión del presente trabajo de investigación se realiza en la sección 5.

2. Trabajo Relacionado

En general la mayoría de los trabajos reportados hasta el momento para darle solución a esta tarea se han encaminado a la extracción de características estadísticas, el uso de diccionarios de expansión y la generación de modelos supervisados de clasificación. A continuación se comentan los trabajos más relevantes reportados en el Semeval 2013. En el trabajo desarrollado por [4] se extraen como características los unigramas y bigramas de palabras, se toma en cuenta una colección de hashtags positivos y negativos. Para las entradas se generaron parejas de unigrama−unigrama, de unigrama−bigrama y de bigrama−bigrama, se eliminó el caracter "@ ". Con las caraterísticas extraídas se desarrolló un modelo de clasificación utilizando la Máquina de Soporte Vectorial(SVM), a través del paquete LibSVM, utilizando un kernel lineal. Este modelo incluye además como características el número de emoticones, los signos de puntuación, el número de palabras con mayúsculas, número de signos de negación y los nombres de usuarios en twitter. Con las caraterísticas seleccionadas reportan una precisión que supera el 60 %.

En la investigación realizada en [3] se propone el desarrollo de un motor de detección[4] que está diseñado específicamente para detectar el sentimiento positivo, negativo o neutral en tweets, para lograr esto se realiza la detección de rasgos semánticos del tweet como emoticones y onomatopeyas, la evaluación de los hashtags y la descomposición del tweet en una lista de n−gramas, se realiza una comparación de cada n-grama con los términos léxicos. Se plantea que dado que los emoticones y onomatopeyas tienen fuertes indicios de sentimiento, pero también tienen una gran variedad ortográfica, se tiene una lista con las exclamaciones más comunes y se utilizan expresiones regulares para capturar la variedad de formas que pueden asumir. Para la evaluación de los hashtags se aplica una serie de heurísticas para dividirlos (en caso de que el hashtag sea compuesto) . El tweet es descompuesto en una lista de unigramas, bigramas, trigramas y tetragramas, se recorren todos los n-gramas del tweet y se realizan comprobaciones de su presencia en diccionarios, finalmente para el análisis de sentimientos se utilizaron cuatro glosarios: tono positivo, tono negativo, fuerza del sentimiento y negaciones; los cuales fueron creados manualmente. En este trabajo se reporta una precisión mayor al 60 %.

Otro trabajo que es importante destacar es el desarrollado por [7]. Para detectar la polaridad de un tweet, se establece que esto es posible si se aplican los

[4] Disponible en www.umigon.com

siguientes pasos de normalización: todas las palabras se convierten a minúscula, se sustituyen @ y # por las notaciones de [usuario], y de [tag] respectivamente, los emoticones se clasifican en positivos y negativos, se remueven caracteres innecesarios, en el caso de palabras que contienen repeticiones de caracteres se reduce la longitud, sólo teniendo en cuenta una secuencia de tres caracteres, con el fin de unificar estas repeticiones. Después de la normalización de los mensajes se determinó la polaridad de cada palabra utilizando el diccionario de sentimientos SentiWordNet, se considera una palabra como positiva si el valor positivo relacionado es mayor a 0.3; como negativa, si el valor negativo relacionado es mayor a 0.2 y como neutral si el valor relacionado es mayor a 0.8. Una vez calculada la polaridad se contemplan tres características para cada tweet que son: el número de palabras positivas, negativas y objetivas respectivamente; se verifica si a una palabra positiva le precede una negación, si es así la polaridad se invierte; se utilizó un diccionario de siglas y para cada sigla se utilizó una polaridad. Esta propuesta supera el 50 % de precisión.

En el trabajo desarrollado por [2], se propone que cada mensaje debe ser preprocesado considerando que los mensajes deben ser tokenizados utilizando una expresión regular, que coincide con direcciones URL, secuencias de caracteres alfanuméricos o secuencias de caracteres sin espacio, de esta forma las secuencias de puntuación como los emoticones se conservan, se separan las palabras en caso de que falte un espacio; los hashtags pueden ser procesados por separado o como n-gramas. Para la clasificación de los mensajes en las clases positiva, negativa y neutra, se utilizaron tres modelos lineales y fueron entrenados uno contra todos, en tiempo de predicción se eligió el marcador con puntuación más alta. Los modelos lineales fueron capacitados mediante el uso del método de optimización gradiente estocástico . El término estocástico se refiere al hecho de que los pesos del modelo se actualizan para cada modelo de entrenamiento.

A pesar de que se han desarrollado diversas investigaciones, los resultados obtenidos hasta el momento en esta propuesta no superan una precisión superior al 60 %.

En [6] proponen un modelo para detectar la ironía en textos obtenidos de Twitter, el cual es organizado alrededor de 4 tipos de características conceptuales:

1. Firmas: engloba elementos tipográficos como signos de puntuación, emoticones y palabras en mayúscula.
2. Lo inesperado: aqui se estudian contradiciones que pudieran ocurrir en la oración, midiendo la similitud semántica de los conceptos del texto.
3. Estilo: está representado obteniendo las frecuencias de estos 3 tipos de secuencias, los n-grams, los skip-grams y los polarity-grams.
4. Escenario emocional: se intenta capturar información emocional de las palabras a través de un diccionario.

Ellos reportan resutados de 73 % para un corpus balanceado y 62 % para un corpus no balanceado.

Otro trabajo relacionado a la detección de ironía es[5], en donde se estudian opiniones de productos de un sitio de ventas en linea. Se usan como carac-

terísticas los n-grams de caracteres para encontrar términos frecuentes, POS n-grams cuyo objetivo es encontrar secuencias de patrones morfosintácticos, ya que la ironía usa aspectos divertidos para crear su efecto, entonces ellos buscan características humorísticas que describan a un documento. También se calcula la relación entre elementos positivos y negativos en el texto y algunas otras emociones. Los resultados reportados por ellos son del 87 % utilizando el algoritmo de Naïve Bayes y del 89.03 % con Máquinas de soporte vectorial.

3. Descripción del enfoque propuesto

En el enfoque propuesto se emplea un modelo supervisado basado en máquinas de aprendizaje, para el cual se construye un modelo de clasificación usando la metodología que se muestra en la figura 1.

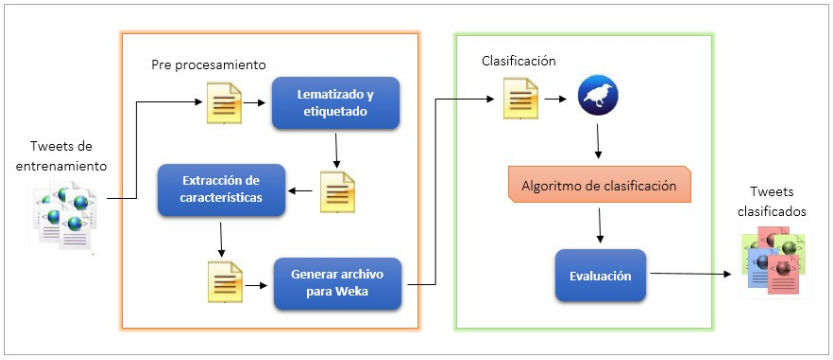

Fig. 1. Metodología empleada.

Como se puede observar, la metodología propuesta consta de dos fases las cuales se explican con mas detalle a continuación:

3.1. FASE 1 - Lematizado y etiquetado

Para realizar este proceso se utilizó la herramienta de Clips llamada pattern.en[5], la cual cuenta con un método llamado "parse.[el] cual identifica las oraciones, las palabras y la categoría a la que pertenece una palabra dentro de una oración. Este parser utiliza un lexicon de 100,000 palabras conocidas y su correspontiende categoría. Un ejemplo de la implementación de esta función se muestra en la figura 1.

[5] http://www.clips.ua.ac.be/pages/pattern-en

```
>>> from pattern.en import parse
>>> print parse('I eat pizza with a fork.')

I/PRP/B-NP/O eat/VBD/B-VP/O pizza/NN/B-NP/O with/IN/B-PP/B-PNP a/DT/B-NP/I-PNP
fork/NN/I-NP/I-PNP ././O/O
```

Fig. 2. Ejemplo de uso de la función parse de pattern.

3.2. FASE 1 - Extracción de características

Las características generales obtenidas del corpus de Twitter proporcionado para esta tarea son las siguientes:

1. Conteo de caracteristícas léxicas.
2. Conteo de categorías gramaticales.
3. Polaridad y objetividad de una oración (Clips).
4. Polaridad de las palabras (SentiWordNet).
5. Emociones de las palabras (NRC Emotion Lexicon).
6. Bolsa de palabras.

La descripción de cómo se calcula cada característica, con el objetivo de construir un vector representativo de cada mensaje, se explica a continuación.

Las caracteristícas léxicas están compuestas por las siguientes 15 carácteristicas:

1. Número de slangs.
2. Número de contracciones.
3. Número de prefijos.
4. Número de signos.
5. Conteo de las 100 palabras más utilizadas.
6. Número de palabras mal escritas.
7. Longitud de la oración.
8. Número de números.
9. Número de palabras que empiezan con mayúscula.
10. Número de palabras escritas en mayúscula.
11. Longitud de la palabra mas larga.
12. Número de palabras de longitud 1,2,10 y 15.

Para estos conteos se desarrollaron diferentes recursos léxicos como son: un diccionario de slangs, un diccionario de signos, diccionario de contracciones, diccionario de prefijos, diccionario de palabras más frecuente y un diccionario que nos permite detectar si la palabra ha sido mal escrita . Las categorías gramaticales generan un vector de 39 catacterísticas ya que esas son las que nos proporciona Clips.

Para detectar el grado de ironía y el lenguaje figurativo se parte de la hipótesis descrita en[6] que plantea que en un texto irónico hay mas elementos positivos que negativos.

Para intentar probar esta hipótesis se utilizan diferentes recursos léxicos para realizar un conteo de cargas positivas y negativas en la oración. Uno de ellos es la función de análisis de sentimientos de Clips con la cual podemos obtener la polaridad y la objetividad del texto, basado en los adjetivos que este contenga. Se sumaron todos los valores de polaridad y objetividad de cada palabra de la oración y se agregaron estas dos como características al vector. Otro recurso léxico que se utilizó y que está disponible para su descarga de forma gratuita, es SentiWordNet[6], del cual se obtuvieron los valores positivos y negativos de cada palabra del texto, ambos valores se agregan al vector como 2 catacterísticas separadas. Para calcular el valor positivo y negativo de cada oración, se sigue el mismo procedimiento que con el anterior. También se utilizó el recurso léxico NRC Emotion Lexicon[7], este describe las palabras con valores binarios a partir de 10 emociones (enojo, anticipación, asco, miedo, alegría, negativo, positivo, tristeza, sorpresa, confianza) , como por ejemplo la representación de la palabra "happy"se puede ver en la tabla 1.

Tabla 1. Ejemplo de la representación de las palabras en NRC

Palabra	Emoción	Valor
happy	anger	0
happy	anticipation	1
happy	disgust	0
happy	fear	0
happy	joy	1
happy	negative	0
happy	positive	1
happy	sadness	0
happy	surprise	0
happy	trust	1

Se agregan 10 caraterísticas que suman el valor de las 10 emociones de cada tweet. Por último se agrega el mensaje como una bolsa de palabras, haciendo un total de 69 características.

3.3. FASE 1 - Generar archivo para weka

Con este vector finalmente se construye un archivo el cual sirve para crear un modelo de clasificación , donde el atributo clasificador varía desde -5.0 hasta 5.0, obteniéndose un total de 11 posibles clases -5, -4, -3, -2, -1, 0, 1, 2 , 3, 4, 5, siendo -5 un tweet totalmente irónico o de doble sentido y 5 un tweet con lenguaje literal.

3.4. FASE 2 - Clasificación

Utilizando el archivo creado en la fase anterior, se utilizan los algoritmos de clasificación Naïve Bayes Multinomial y Máquinas de soporte vectorial (SMO) implementados en Weka.

[6] http://sentiwordnet.isti.cnr.it/
[7] http://www.saifmohammad.com/WebPages/ResearchInterests.html

4. Resultados Experimentales

El corpus de entrenamiento se compone de textos breves (mensajes), una descripción completa del conjunto de datos de entrenamiento se encuentra en el artículo de descripción de la tarea 11[1].

En la tabla 2 se pueden apreciar los resultados obtenidos, se realizaron diferentes experimentos para analizar el impacto de cada característica sobre el conjunto de entrenamiento. Se puede observar el desempeño de ambos algoritmos. Se aplicó validación cruzada con 10 pliegues.

El mejor comportamiento se presentó al utilizar el clasificador máquina de soporte vectorial, con una precisión del 43.12 %. Se considera que este resultado se obtuvo como consecuencia de que el corpus está totalmente desbalanceado y el tamaño de los tweets difiere mucho, lo que provoca que ciertas características no puedan ser obtenidas. Las investigaciones que han obtenido mejores resultados reportan que los corpus evaluados con su metodología tienen características muy particulares y sobre todo como son opiniones de usuarios de amazon el tamaño de cada opinión es mucho más grande.

Tabla 2. Resultados obtenidos

System	**NaïveBayesMultinomial**	**SMO**
Léxicas	36.95	37.33
Léxicas+Gramáticales	38.57	38.28
Léxicas+Gramáticales+Clips	38.51	38.82
Léxicas+Gramáticales+Clips+SentiWordNet	38.37	38.44
Léxicas+Gramáticales+Clips+SentiWordNet+Bolsa	41	**43.12**
Léxicas+Gramáticales+Clips+SentiWordNet+Bolsa+NRC	41.6	42.71

5. Conclusiones

Se ha presentado una aproximación para detectar el grado de lenguaje figurativo o ironía de un mensaje usando básicamente tres tipos de características: polaridad y emoción de las palabras, conteos de diferentes elementos y palabras del mensaje como tal. Los resultados obtenidos muestran que estas características no son suficientes para detectar correctamente la clase de un mensaje con alta precisión. Se considera que el desbalanceo de los datos de entrenamiento, así como el número de clases que se tienen en total, influyó significativamente en los resultados que se han obtenido. Sin embargo, se está analizando los conjuntos de datos para lograr encontrar nuevas características, que permitan desarrollar un mejor modelo de clasificación .

Referencias

1. John Barnden, Antonio Reyes, Ekaterina Shutova, Paolo Rosso, and Tony Veale. Semeval−2015 task 11: Sentiment analysis of figurative language in twitter. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval−2015)*, Dublin, Ireland, 2015.

2. Tobias Gunther and Lenz Furrer. Gu−mlt−lt: Sentiment analysis of short messages using linguistic features and stochastic gradient descent. In *Proceedings of the seventh international workshop on Semantic Evaluation Exercises (SemEval-2013)*, Atlanta, Georgia, USA, June 2013.
3. Clement Levallois. Umigon: Sentiment analysis for tweets based on lexicons and heuristics. In *Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 520–524, Atlanta, Georgia, USA, June 2013. Association for Computational Linguistics.
4. Saif M. Mohammad, Svetlana Kiritchenko, and Xiaodan Zhu. Nrc−canada: Building the state-of-the-art in sentiment analysis of tweets. In *Proceedings of the seventh international workshop on Semantic Evaluation Exercises (SemEval-2013)*, Atlanta, Georgia, USA, June 2013.
5. Antonio Reyes and Paolo Rosso. Making objective decisions from subjective data: Detecting irony in customer reviews. *Decision Support Systems*, 53(4), november 2012.
6. Antonio Reyes, Paolo Rosso, and Tony Veale. A multidimensional approach for detecting irony in twitter. *Language Resources and Evaluation*, 47(1), march 2013.
7. Rich ard Farkas Viktor Hangya, Gabor Berend. Szte−nlp: Sentiment detection on twitter messages. Atlanta, Georgia, USA, June 2013. Association for Computational Linguistics.

Una metodología para la detección del perfil de un autor

Yuridiana Alemán, Darnes Vilariño, David Pinto
yuridiana.aleman@gmail.com, darnes@cs.buap.mx, and dpinto@cs.buap.mx

Facultad de Ciencias de la Computación-BUAP
Av. San Claudio y 14 sur, CP: 72570, Puebla, Mexico
http://www.cs.buap.mx/

Resumen En este artículo se presenta una metodología para la detección del perfil de un autor usando los siguientes dos elementos: género y edad. Para esto se parte de un conjunto de diálogos, escritos en dos idiomas: inglés y español, proporcionado para la competencia Author Profiling dentro del foro de evaluación "Uncovering Plagiarism, Authorship, and Social Software Misuse" (PAN2013). Se utiliza conteos de características léxicas, semánticas y sintácticas para generar un sistema de clasificación de dos fases, el cual clasifica primero el género y posteriormente la edad. Los resultados obtenidos muestran que, con la cantidad de datos disponibles, es posible caracterizar tanto la edad como el género de un autor con una exactitud mayor al 50 %. Sin embargo, estos valores podrían mejorarse al tener una mayor evidencia de información en los datos de entrenamiento.

Palabras clave: Clasificación supervisada, PAN 2013, Género, Edad, Bosque aleatorio

1. Introducción

La detección del perfil de un autor es un problema de creciente importancia en varios campos del conocimiento como medicina forense, seguridad y marketing. Por ejemplo, desde la perspectiva de la lingüística forense, sería importante conocer el perfil lingüístico del autor de un mensaje de acoso. Del mismo modo, desde el punto de vista del marketing, las empresas pueden estar interesadas en saber, mediante el análisis de los blogs y reseñas de productos en línea, que tipo de personas comentan sobre sus productos, y de esta manera encaminar sus campañas publicitarias hacia determinado género o rango de edad.

En este artículo se propone una metodología para detectar dos aspectos del perfil de los autores en los chats o blogs: género y edad. Esta metodología ha permitido crear un sistema que, a partir de un conjunto de diálogos escritos por diferentes personas (cada diálogo contiene el género y grupo de edad de la persona que lo escribió), catalogue un nuevo conjunto de diálogos (conjunto de evaluación) y determine los dos aspectos del perfil antes mencionados. Los textos utilizados se presentan en los idiomas inglés y español.

El artículo está estructurado de la siguiente manera. En la sección II se detalla el estado del arte referente a este tema de investigación. La sección III muestra la metodología planteada y el preprocesamiento dado a los conjuntos de datos. La sección IV muestra los resultados obtenidos en cada corpus. Finalmente, la sección V muestra las conclusiones obtenidas y el trabajo futuro para esta investigación.

2. Estado del arte

Gran parte de las investigaciones sobre blogs y redes sociales van encaminadas a descubrir al autor del texto, sin tomar en cuenta el perfil de dicho autor. En la rama de detección de perfil, específicamente género y edad, se han logrado algunos avances. En [1] se aplican técnicas de aprendizaje automático para la categorización de textos. Se utilizan dos tipos básicos de categorías: basadas en contenido y basadas en estilo, para determinar dos aspectos del perfil del autor (género y edad). Se realizaron tres tipos de experimentos, con los conjuntos de características tratados de forma separada y por último la unión de ambos conjuntos, utilizando taxonomías para determinar la relación de las palabras con la parte del discurso en cada texto escrito.

Esta misma metodología se utiliza en [2] donde el corpus analizado contiene 4 aspectos a determinar: género, edad, personalidad y lengua nativa. Los mejores resultados en género y edad fueron obtenidos utilizando la unión de las características de estilo y contenido; la lengua materna obtiene mejores resultados con el conjunto de características de contenido, mientras que la personalidad obtiene mejores resultados con las categorías de estilo. Las evaluaciones se realizaron con validación cruzada de 10 pliegues. En [3], se presenta un estudio donde se utilizan técnicas de categorización de texto para la predicción de género y edad en un corpus de mensajes escrito en el idioma holandés. Para los experimentos se utiliza el análisis de X^2, n-gramas (palabras y caracteres) y bolsa de palabras. Se realizan varios experimentos con diferentes números de instancias y características para clasificar dos tipos de categorías: adolescentes y adultos, utilizando el clasificador conocido como máquinas de soporte vectorial. Los mejores resultados se obtuvieron usando unigramas de palabras como esquema de representación de los textos.

En investigaciones más recientes como en [4], se estudia la relación entre la edad y el uso del lenguaje en la red social de Twitter en tres diferentes tipos de experimentos. La clasificación de los usuarios por categorías de edades, por etapas de la vida y la predicción de su edad exacta. Se concluye con que un sistema automático logra mejor rendimiento que los seres humanos en estas tareas, y que los seres humanos y los sistemas automáticos tienen dificultades en la predicción de la edad de las personas mayores. Por otra parte, se presenta un análisis detallado de las variables que cambian con la edad. En particular, se encuentran fuertes patrones de cambio mayoritariamente en personas más jóvenes. En [5] se presenta una propuesta para la detección automática del género y de la edad utilizando ciertas características estilísticas, tales como:

emoticonos, signos de puntuación y frecuencia de palabras. Los experimentos fueron realizados utilizando máquinas de soporte vectorial obteniendo alrededor de 0.57 de exactitud en la detección de género y cerca de 0.65 para la detección de la edad. La conclusión obtenida a través de estos resultados es que el estilo de escritura está correlacionado mayormente con la edad de las personas y menos con el género. Es importante mencionar que aunque se han logrado avances en la detección del perfil del autor, estos resultados no han sido validados en otras colecciones de datos, y por tanto, se trata de valores que se encuentran ajustados a ciertos tipos de corpora. En este sentido, en el año 2013 se propuso una competencia internacional (PAN 2013), con la finalidad de proponer un corpus estándar y balanceado, así como evaluar el rendimiento de diversos sistemas a nivel mundial.

3. Metodología

La metodología propuesta se basa en la extracción de diversos conjuntos de características, ésta se muestra en la figura 1.

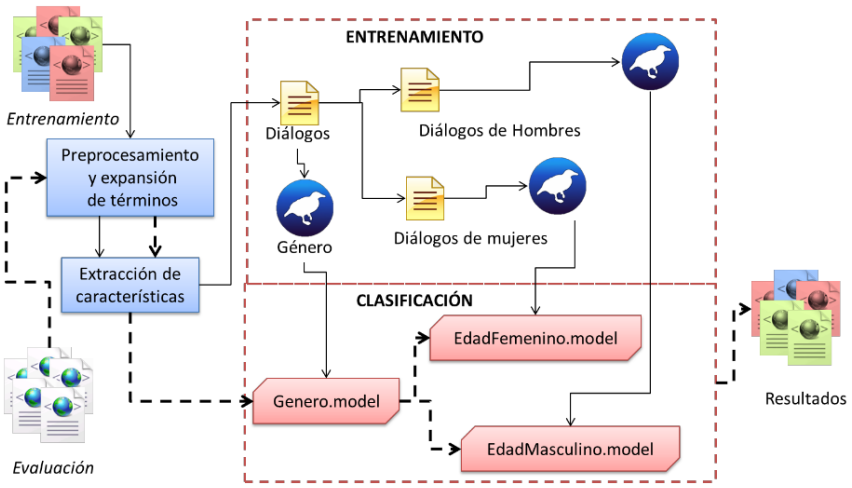

Fig. 1. Metodología propuesta para la investigación

El conjunto de entrenamiento consiste en documentos de tipo XML, que contienen conversaciones sobre diferentes temas agrupados por autor y etiquetados con su idioma (inglés o español), género (masculino y femenino) y grupo de edad. Existen tres grupos de edad:

1. 10s: Personas de 13 a 17 años.
2. 20s: Personas de 23 a 27 años.

3. 30s: Personas de 33 a 47 años.

El corpus del inglés contiene 236,600 autores, mientras que el corpus en español contiene 75,900. En la tabla 1 se muestran algunas estadísticas de los diálogos, separados por idioma y género.

Tabla 1. Conjunto de entrenamiento para la tarea *Author profiling*

	Inglés		Español	
	Femenino	**Masculino**	**Femenino**	**Masculino**
Total de diálogos	118,300	118,300	37,950	37,950
Diálogos de "10s"	8,600	8,600	1,250	1,250
Diálogos de "20s"	42,900	42,900	21,300	21,300
Diálogos de "30s"	66,800	66,800	15,400	15,400
Vocabulario	1,228,711	1,219,020	533,873	592,605
Promedio de palabras	798	705	249	273
Diálogo con más palabras	10,648	12,917	11,806	11,714

El vocabulario es muy extenso, sobre todo para el idioma inglés, esto es lógico debido al número de diálogos para este idioma. Para el caso del idioma español es más extenso el vocabulario en la categoría de masculino, mientras que para el inglés es más extenso el vocabulario en la categoría de femenino. Además, se observa que en todas las categorías de edad, el número de textos es el mismo para ambos géneros (en este sentido es un corpus balanceado por género para cada rango de edad).

Analizando los conjuntos de textos, se llegó a la conclusión de que éstos presentan muchas palabras mal escritas, truncadas, emoticones y vocabulario propio de blogs y chats. Por lo tanto, se realiza una expansión de los textos construyendo algunos recursos léxicos para ambos idiomas,como son: un diccionario de emoticones, un diccionario de abreviaturas, un diccionario de palabras comunes en SMS y de las contracciones mas utilizadas. Usando los recursos léxicos anteriormente mencionados, se fue capaz de sustituir en el corpus de entrenamiento cada ocurrencia de emoticones y contracciones por su correspondiente significado. Se eliminaron además los signos de puntuación y los caracteres no imprimibles. Después de aplicar este pre-procesamiento, el vocabulario en ambos idiomas se redujo drásticamente.

En la fase de extracción de características la mayoría de las investigaciones realizadas emplean todo el vocabulario, sin embargo, en este caso se observa que este tipo de enfoque consumiría demasiados recursos de espacio y tiempo de cómputo, e incluso ciertas herramientas de aprendizaje automático no serían capaces de soportar tal volumen de información, es por ello que se propone utilizar los siguientes conteos:

1. **Categorías Gramaticales:** Se obtuvo la categoría gramatical de cada palabra dentro de los textos, para posteriormente realizar el conteo de cada una de ellas. Para esto, se utilizó la herramienta *Tree-Tagger* [6] para el idioma español y el *Stanford POS-tagger* [7] para el inglés, obteniendo 102 características para el idioma español y 52 para el inglés.

2. **Palabras Cerradas:** Dentro de esta categoría se clasifican grupos de palabras como preposiciones, conjunciones y determinantes. Para los experimentos se realiza un conteo de palabras cerradas en todas las conversaciones; así, cada palabra cerrada representa una característica, y el valor de ésta en cada instancia está dado por las veces que aparece en dicha conversación. Se obtienen 195 palabras para el idioma inglés y 178 para el español.
3. **Sufijos:** Se tomaron como características los sufijos existentes para ambos idiomas. Al igual que en los conjuntos anteriores, cada sufijo representa una característica, y las veces que aparece en cada conversación es el valor para dicho atributo. En este conjunto se obtuvieron 131 características para el idioma inglés y 172 para el español.
4. **Signos:** Se contabilizan todos los signos de puntuación existentes.

Tal y como se ha mencionado anteriormente, en este artículo se utiliza el enfoque de aprendizaje automático para detectar el perfil de los autores. Este enfoque parte de la premisa de la existencia de un corpus supervisado que se usa para entrenar un modelo de clasificación, que es usado posteriormente para calcular la clase asociada a un texto de entrada cuya clase es desconocida. Utilizando las características antes mencionadas, se han generado los siguientes modelos de clasificación mediante el método de *Random Forest* [8]:

1. *Género*: Se usan todos los textos del conjunto de entrenamiento usando el género (masculino, femenino), como atributo clasificador.
2. *EdadMasculino*: Se agrupan todos los textos escritos por hombres usando el rango de edad (10s, 20s y 30s), como atributo clasificador.
3. *EdadFemenino*: Se agrupan todos los textos escritos por mujeres usando el rango de edad (10s, 20s y 30s), como atributo clasificador.

Los modelos se crean de esta manera bajo la premisa de que en primera instancia, es más fácil detectar el género, que la edad en la forma de escribir de un autor. En la Figura 1, se puede observar el proceso de dos fases usado para determinar el perfil del autor. La clasificación de las instancias se realizó evaluando primeramente el modelo Género, y una vez obtenida la categoría (masculino o femenino), los textos etiquetados como "femenino" se evalúan con el modelo *EdadFemenino* para obtener la etiqueta de edad, mientras que los etiquetados como "masculino" son evaluados con el modelo *EdadMasculino*. Con la finalidad de medir objetivamente los resultados obtenidos, se han llevado a cabo dos tipos de evaluaciones:

1. **Validación cruzada a 10 pliegues:** Los datos se dividen en K subconjuntos, uno de los subconjuntos se utiliza como datos de prueba y el resto (K-*1*) como datos de entrenamiento. Este proceso se repite 10 veces.
2. ***Software submission*:** Se genera el modelo de clasificación usando la totalidad de los datos de entrenamiento para el conjunto de evaluación de la competencia "PAN 2013"[1]. Regularmente, los foros de evaluación proporcionan el conjunto de prueba para validar los modelos de clasificación, de

[1] http://pan.webis.de/

tal manera que se envían los resultados obtenidos, los cuales son evaluados y comparados contra otras aproximaciones. Sin embargo, en el marco del "PAN 2013", la metodología establecida fue que cada equipo debe enviar el software programado e instalarlo en una máquina virtual. Así, los organizadores son capaces de ejecutarlo usando un conjunto de datos prueba que no es accesible para los equipos que envían el software.

4. Resultados obtenidos

En esta sección se presentan los resultados obtenidos tanto para la evaluación por validación cruzada, como por *software submission*. Para el análisis de los resultados obtenidos se utilizan las métricas de precisión, recuerdo y la medida F_1, la cual se define como la media armónica entre la precisión y el recuerdo.

4.1. Validación cruzada a 10 pliegues

Como se mencionó anteriormente, en este tipo de evaluación se usa únicamente el conjunto de datos de entrenamiento. En la Tabla 2 se muestran los resultados obtenidos al clasificar solamente el género, mientras que en la Tabla 3 se muestran los valores obtenidos al clasificar únicamente la edad del autor. El valor de medida F_1 obtenido para el caso de la clasificación de género es de 0.54, para el idioma Español, mientras que es de 0.55 para el idioma Inglés. La clasificación de la edad arrojó un F_1 igual a 0.37 para el idioma Español y 0.44 para el idioma Inglés. Como puede observarse, sobre el conjunto de datos usado como entrenamiento, los valores promedio validan la hipótesis planteada, en la cual se indica que es más fácil detectar el género del autor, que su edad. El comportamiento observado fue similar para ambos idiomas (Español e Inglés).

Tabla 2. Evaluación por validación cruzada usando el género (masculino y femenino) como atributo clasificador

	Español			Inglés		
Categoría	**Precisión**	**Recuerdo**	F_1	**Precisión**	**Recuerdo**	F_1
Masculino	0.5350	0.6430	0.5840	0.5410	0.6480	0.5897
Femenino	0.5560	0.4440	0.4937	0.5640	0.4530	0.5024
Promedio	**0.5455**	**0.5435**	**0.5445**	**0.5525**	**0.5505**	**0.5515**

Sin embargo, este comportamiento podría estar afectado de alguna manera por el número de muestras para la clase de autores que se encuentran en el rango de los 10 años (10s), ya que al ser pocas pueden no representar correctamente la clase y generar un modelo sobreajustado, que no se comporte adecuadamente sobre el corpus de prueba. Esta clase es demasiado pequeña con relación a las otras dos (20s y 30s) y por tanto, el proceso de clasificación puede tender a favorecer a las clases con mayor número de muestras. Bajo este razonamiento, y considerando un proceso de clasificación basado en dos fases, se proponen dos

Tabla 3. Evaluación por validación cruzada usando la edad (10s, 20s y 30s) como atributo clasificador

	Español			Inglés		
Categoría	**Precisión**	**Recuerdo**	F_1	**Precisión**	**Recuerdo**	F_1
10s	0.0000	0.0000	0.0000	0.3000	0.0200	0.0375
20s	0.6100	0.7900	0.6900	0.5100	0.4700	0.4892
30s	0.5100	0.3300	0.4100	0.6400	0.7400	0.6864
Promedio	**0.3733**	**0.3733**	**0.3733**	**0.4833**	**0.4100**	**0.4437**

evaluaciones: primeramente clasificar el género del autor seguido de la clasificación de la edad (tal y como se propuso en la metodología inicial) y en una segunda evaluación se realiza el proceso inverso.

En la Tabla 4 se presentan los valores de precisión, recuerdo y F_1 para cada modelo de clasificación aplicado en la segunda fase de la metodología inicial, es decir, tomando los modelos de género para determinar la edad.

Tabla 4. Resultados de los modelos que usan la edad (10s, 20s y 30s) como atributo clasificador en la segunda fase de clasificación.

		Español			Inglés		
Modelo	Categoría	**Precisión**	**Recuerdo**	F_1	**Precisión**	**Recuerdo**	F_1
Edad Masculino	10s	0.1430	0.0040	0.0078	0.3100	0.0280	0.0514
	20s	0.6110	0.7820	0.6860	0.5180	0.4750	0.4956
	30s	0.5190	0.3550	0.4216	0.6240	0.7330	0.6741
	Promedio	0.4243	0.3803	0.4011	0.4840	0.4120	0.4451
Edad Femenino	10s	0.0000	0.0000	0.0000	0.3060	0.0300	0.0546
	20s	0.6040	0.7680	0.6762	0.5170	0.4580	0.4857
	30s	0.5050	0.3530	0.4155	0.6440	0.7610	0.6976
	Promedio	0.3697	0.3737	0.3717	0.4890	0.4163	0.4497
Promedio General		**0.3970**	**0.3770**	**0.3864**	**0.4865**	**0.4142**	**0.4474**

Se observa un mejor comportamiento en el rango de edad es de 20 años, para los diálogos escritos en Español (independientemente del género). En el caso de los diálogos escritos en Inglés, el mejor resultado se obtiene en el rango de edad de los 30 años. Desde nuestro particular punto de vista, estos resultados están co-relacionados con la cantidad de diálogos que se tienen por rango de edad en el conjunto de datos de entrenamiento. El experimento arroja un F_1 de 0.39 para el idioma Español y de 0.45 para el Inglés. Tal y como se esperaba, estos valores de F_1 son menores que aquellos obtenidos en la primera fase de clasificación. Los diálogos mal clasificados en la fase uno, están destinados a provocar un margen de error cercano al 45 % en la segunda fase de clasificación.

Para el caso de la segunda evaluación, se creó un modelo de clasificación en donde se utiliza únicamente la edad como atributo clasificador. La salida de este modelo puede ser “10s”, “20s” o “30s” (Resultados en la tabla 3). Una vez identificado el rango de edad del autor, se procede a identificar el género, de acuerdo a la edad identificada en la fase anterior. Así, se tienen ahora tres modelos de clasificación adicionales, uno que entrena sobre los diálogos escritos por personas en el rango de edad de los 10 años, otro sobre el rango de edad de los 20 años, y el último sobre el rango de edad de los 30 años.

Tabla 5. Resultados de los modelos que usan el género (masculino y femenino) como atributo clasificador en la segunda fase de clasificación

		Español			Inglés		
Modelo	Categoría	**Precisión**	**Recuerdo**	F_1	**Precisión**	**Recuerdo**	F_1
Genero10s	Masculino	0.0000	0.0000	0.0000	0.1912	0.2453	0.2149
	Femenino	0.0000	0.0000	0.0000	0.3069	0.2366	0.2672
Genero20s	Masculino	0.3218	0.3915	0.3532	0.3344	0.3927	0.3612
	Femenino	0.3405	0.2678	0.2998	0.2849	0.2240	0.2508
Genero30s	Masculino	0.2703	0.3372	0.3001	0.3193	0.3769	0.3457
	Femenino	0.2844	0.2128	0.2435	0.3656	0.3055	0.3329
Promedio General		**0.2028**	**0.2015**	**0.2022**	**0.3004**	**0.2968**	**0.2986**

La salida final indica el rango de edad y el género identificado al que pertenece un determinado autor. Si bien, este sistema tiene la misma entrada y la misma salida que el presentado anteriormente, los resultados que se observan en la Tabla 5 muestran el impacto de seleccionar una fase por encima de la otra. Se presentan de nuevo los valores de precisión, recuerdo y F_1 para cada modelo de clasificación aplicado en la segunda fase.

Al comparar los resultados obtenidos con respecto al esquema anterior, se observa una pérdida mayor de precisión en todos los rangos de edad, sin importar el idioma. En particular, para el caso del idioma Español, no se logró distinguir los diálogos escritos por los autores en el rango de edad de los 10 años. Este hecho sugiere de nuevo que los resultados están co-relacionados con la calidad del corpus de entrenamiento, de tal manera que los diálogos no son representativos de la clase y pueden haber generado un modelo sobreajustado, tal y como se mencionó anteriormente.

Al momento, no existe evidencia de que el proceso de clasificación se comporte diferente para cada idioma evaluado. Los resultados obtenidos se corresponden completamente con el número de muestras que tiene cada clase en el corpus de datos.

Finalmente, se decidió investigar el comportamiento del proceso de clasificación usando un modelo único, es decir, una sola fase en donde existen 6 clases distintas asociadas a las diferentes categorías que presenta el corpus: masculino en el rango de 10 años, masculino en el rango de 20 años, masculino en el rango de 30 años, femenino en el rango de 10 años, etc. Los resultados obtenidos y su comparación con los dos modelos presentados anteriormente (Género->Edad y Edad->Género) se muestran en la Figura 2.

Al utilizar las 6 clases se evita arrastrar errores de una fase a otra , sin embargo, existe el problema del incremento en el número de clases. El clasificador tiene mayor dificultad para discernir entre las diferentes posibilidades. Si a este problema, se le agrega el hecho de que las clases no están balanceadas, entonces se obtiene el comportamiento mostrado en la Figura 3. Básicamente, la mejor aproximación que se pudo tener en los experimentos presentados en este trabajo fue cuando se usó un proceso de clasificación en dos fases, identificando primero el género y después el rango de edad del autor. En función de estos resultados obtenidos, se ha decidido usar esta aproximación para la evaluación del corpus de prueba en la aproximación "software-submission".

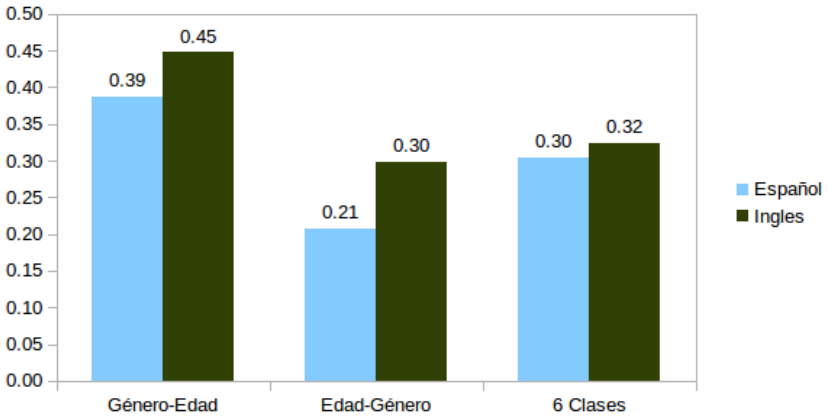

Fig. 2. Comparación entre las aproximaciones de dos fases presentadas y un sólo clasificador con 6 clases

4.2. Software submission

El sistema presentado para la competencia "PAN 2013"(Aleman) obtuvo el lugar 7 con una exactitud de 55.22 % para la detección del género, 59.23 % para la detección de la edad y una exactitud general de 32.92 %. Como se puede observar en la tabla 6.

Tabla 6. Resultados obtenidos por la aproximación de dos fases (Género-Edad) en la competencia PAN 2013 y comparados con otras aproximaciones

Team	Total	Gender	Age
Meina	0.3894	0.5921	0.6491
Pastor L.	0.3813	0.5690	0.6572
Seifeddine	0.3677	0.5816	0.5897
Santosh	0.3508	0.5652	0.6408
Yong Lim	0.3488	0.5671	0.6098
Ladra	0.3420	0.5608	0.6118
Aleman	**0.3292**	**0.5522**	**0.5923**
Gillam	0.3268	0.5410	0.6031
Kern	0.3115	0.5267	0.5690
Cruz	0.3114	0.5456	0.5966
Pavan	0.2843	0.5000	0.6055
Caurcel Diaz	0.2840	0.5000	0.5679
H. Farias	0.2816	0.5671	0.5061
Jankowska	0.2814	0.5381	0.4738
Flekova	0.2785	0.5343	0.5287
Weren	0.2564	0.5044	0.5099
Sapkota	0.2471	0.4781	0.5415
De-Arteaga	0.2450	0.4998	0.4885
Moreau	0.2395	0.4941	0.4824
baseline	0.1650	0.5000	0.3333
Gopal Patra	0.1574	0.5683	0.2895
Cagnina	0.0741	0.5040	0.1234

Estos resultados son bastante similares a los obtenidos en las pruebas con el conjunto de entrenamiento y aunque de manera global la exactitud no supera el 50 %, ningún equipo participante alcanzó este valor, quedando el primer lugar con 38.9 % de exactitud.

Esta misma aproximación fue evaluada en el marco de la competencia con la finalidad de detectar conversaciones en las que intervienen pederastras. Los resultados obtenidos nos colocan en el lugar 12, de 21 equipos participantes. De hecho, existen 8 equipos que detectan al 100 % el género de los depredadores, sin embargo, este resultado no es tan significativo dado que es bastante conocido que la mayoría de los depredadores sexuales son del sexo masculino, lo cual también se refleja en el conjunto de datos de la competencia. Nuestra aproximación obtuvo un 74 % de exactitud en la detección del género, lo cual consideramos es consistente con los resultados que se han reportado a lo largo de este trabajo. La conclusión es que se debe afinar el conjunto de características de acuerdo al tipo de atributo clasificador, ya que no es lo mismo clasificar género que edad. En el género influye el carácter de las personas y los hábitos de comunicación entre hombres y mujeres, pero en el caso de la edad, el tamaño del vocabulario, por ejemplo, debería ser significativo, puesto que las personas suelen incrementar y modificar el vocabulario a medida que crecen.

5. Conclusiones y trabajo futuro

En este artículo se presentó una metodología para la detección de género y edad en blogs y chats. Esta metodología usa conteos de características léxicas, sintácticas y semánticas para representar los diálogos de las personas con la finalidad de entrenar un modelo de clasificación supervisado que permita determinar el género y el rango de edad de un autor.

Los resultados obtenidos utilizando solamente el conjunto de entrenamiento muestran que existe una mejor clasificación en género que en edad, sin embargo, en ninguno de los dos casos se supera el 55 % de medida F_1, además, esta medida baja cuando se unen los dos clasificadores, llegando a un valor F_1 entre 40 % y 44 %.

Las métricas muestran valores más altos para el idioma inglés, que para el idioma español, lo cual parece estar directamente relacionado con el número de diálogos para cada idioma. En particular, el número de diálogos para los autores que se encuentran en el rango de edad de los 10 años es muy bajo y por tanto, existe el riesgo de que no sean representativos de la clase en la vida real. Consideramos así, que la calidad del corpus tuvo un efecto negativo en el proceso de clasificación, sobreajustando los datos de entrenamiento y generando un modelo que no puede reconocer adecuadamente los datos de prueba, especialmente para el conjunto de diálogos que fueron escritos por autores en el rango de edad de los 10 años.

Aunque los resultados de la competencia no sobrepasan el 50 %, a nivel mundial se logró una buena posición respecto al total de equipos participantes. Por lo tanto, aún falta mucho en investigaciones para lograr superar los resultados

aquí obtenidos. Con todos los experimentos realizados se puede concluir que es necesario construir un corpus representativo del género, pero principalmente de los rangos de edad, puesto que éstos se encuentran totalmente desbalanceados y, desde nuestro particular punto de vista, pocamente representados, principalmente para el caso de aquellos escritos por autores en el rango de edad de los 10 años.

Es preciso hacer un análisis lingüístico profundo para determinar las características representativas para cada modelo de clasificación. En este sentido, se considera firmemente que estas características serán diferentes cuando se busque identificar el género de una persona que cuando se intente determinar su rango de edad.

Referencias

1. Schler, J., Koppel, M., Argamon, S., Pennebaker, J.W.: Effects of age and gender on blogging. In: Computational Approaches to Analyzing Weblogs, Papers from the 2006 AAAI Spring Symposium, Technical Report SS-06-03, Stanford, California, USA, March 27-29, 2006. (2006) 199–205
2. Argamon, S., Koppel, M., Pennebaker, J.W., Schler, J.: Automatically profiling the author of an anonymous text. Commun. ACM **52**(2) (February 2009) 119–123
3. Peersman, C., Daelemans, W., Van Vaerenbergh, L.: Predicting age and gender in online social networks. In: Proceedings of the 3rd International Workshop on Search and Mining User-generated Contents. SMUC '11, New York, NY, USA, ACM (2011) 37–44
4. Nguyen, D., Gravel, R., Trieschnigg, D., Meder, T.: "how old do you think i am?": A study of language and age in twitter. In: Proceedings of the Seventh International AAAI Conference on Weblogs and Social Media. ICWSM 2013 (2013)
5. Rangel, F., Rosso, P.: Use of language and author profiling: Identification of gender and age. In: Proceedings of the 10th Workshop on Natural Language Processing and Cognitive Science (NLPCS-2013). (2013)
6. Schmid, H.: Probabilistic part-of-speech tagging using decision trees. In: Proceedings of the International Conference on New Methods in Language Processing, Manchester, UK (1994)
7. Toutanova, K., Klein, D., Manning, C., Singer, Y.: Feature-rich part-of-speech tagging with a cyclic dependency network. In: Human Language Technology Conference (HLT-NAACL 2003). (2003)
8. Breiman, L.: Random forests. Mach. Learn. **45**(1) (October 2001) 5–32

Extracción y clasificación de características de imágenes de incendios forestales

Belém Priego Sánchez[1], Esteban Castillo[2], Arturo Bustamante[3], Iván Olmos[3]

[1]Laboratoire LDI (Lexique, Dictionnaires, Informatique)
Université Paris 13, Sorbonne Paris Cité, Villetaneuse, France
belemps@gmail.com

[2]Universidad de las Américas Puebla
Departamento de Computación, Electrónica y Mecatrónica, México
esteban.castillojz@udlap.mx

[3]FCC, Benemérita Universidad Autónoma de Puebla, México
{dvartus, ivanoprkl}@gmail.com

Resumen Este artículo se enfoca en el análisis y extracción de características de imágenes de incendios forestales (usando procesamiento digital de imágenes), las cuales son utilizadas como entrada en un proceso de clasificación para la construcción de modelos que permitan a futuro identificar si una imagen tiene o no un incendio forestal. Las imágenes se procesaron a través de un proceso de particionamiento, donde de cada fragmento se obtuvieron un conjunto de características basadas en la estadística. Para el proceso de clasificación, se utilizaron dos técnicas de aprendizaje automático: árboles de decisión y redes neuronales. A partir de los resultados obtenidos, se logró obtener una clasificación global del 84 %, mostrando que el proceso de particionamiento de la imagen, junto con los clasificadores seleccionados, son un medio viable para identificar incendios forestales a partir de fotografías digitales.

Palabras clave: Incendios forestales, minería de datos, árboles de decisión, red neuronal.

1. Introducción

Los incendios forestales son uno de los fenómenos que afectan negativamente tanto a la naturaleza como a la calidad de vida de los humanos. Desde el punto de vista ecológico, un incendio forestal altera un ecosistema, desde el tipo de vegetación presente en la zona siniestrada, hasta la fauna que habita en ella.

En México, existen esfuerzos tanto a nivel federal como estatal para detectar de forma temprana un incendio forestal. Por ejemplo, en el estado de Puebla, desde el año 2009 se instaló una red de videocámaras que permiten monitorear regiones del estado de Puebla propensas a estos fenómenos. El sistema es operado de forma remota por personal de la Secretaría de Medio Ambiente y Recursos Naturales (SMNR) a través de un centro de control ubicado en la reserva

ecológica Flor del Bosque. A pesar de ser un importante avance en el combate de los incendios forestales, este sistema tiene algunos puntos de oportunidad que pueden ser mejorados. Uno de ellos es la manipulación manual para la detección de los incendios forestales, ya que requiere que el personal manipule manualmente cada videocámara en busca de un posible incendio forestal, lo cual puede llegar a ser cansado y tedioso.

Una estrategia que puede resolver este problema planteado es el desarrollo de un sistema que permita procesar imágenes de incendios forestales y que determine de forma automática la presencia de estos eventos en una región. En este sentido, en el presente trabajo se propone un método para el procesamiento de las imágenes, que permita clasificar a nuevas imágenes como dos clases: presencia de incendio forestal o ausencia del mismo.

La propuesta se basa en el proceso de Descubrimiento de Conocimiento en Bases de Datos (Proceso KDD), que contempla desde la construcción de una base de datos (en este caso, imágenes con incendios forestales), selección de las muestras más representativas, proceso de transformación, técnicas de aprendizaje automático para aprender las características de los patrones (los incendios forestales), hasta la interpretación de los resultados. En este trabajo, la parte de aprendizaje automático se centró en árboles de clasificación (decisión) y redes neuronales, que son técnicas que permiten construir modelos descriptivos para la clasificación de instancias, en donde es posible analizar y construir reglas para entender que características influyen en el proceso de clasificación [1]. Para esto se propone utilizar un proceso de análisis de las imágenes basado en particiones, con el fin de medir con mayor precisión las caracterícticas propias de un incendio forestal.

Un incendio forestal (IF) es la propagación libre y no programada del fuego sobre la vegetación, causado en forma natural, accidental o intencional debido a lo cual se afectan combustibles naturales situados en áreas boscosas [2]. Existen varios tipos de incendios, los más comunes en México son los superficiales en un 90 % [3]. Esta clase de desastre se caracteriza porque el fuego consume únicamente lo que los especialistas llaman combustibles forestales, es decir: hierbas, sácate, hojarasca y leña suelta; sin dañar los árboles en su totalidad en algunos casos.

En investigaciones sobre incendios forestales, la propagación del fuego depende de la húmedad, de la acumulación y combinación de los atributos de los combustibles. En las hierbas, arbustos, árboles vivos la propagación depende de su humedad. Por otro lado, se encuentran las ramas, hojas y arbustos secos que tienen alta inflamabilidad dependiendo de su abundancia y posición (generalmente en el suelo) por lo que también se les llama combustibles rápidos.

A partir de 1998, en México el Servicio Meteorológico Nacional empezó a intentar reducir los problemas causados por los IF [4], debido a que el número de incendios se incremento de manera alarmante, la medida que se tomó es la creación del Programa para Detección de Puntos de Calor Mediante Técnicas de Percepción Remota [5], cuyo propósito principal es ubicar aquellos sitios donde se registran temperaturas mucho más altas de lo normal, a fin de alertar

sobre posibles incendios [5]. En este programa se uso imágenes de satélite y en particular, las generadas por el sensor AVHRR (Advanced Very High Resolution Radiometer) de la Nacional Oceanic and Atmospheric Administration (NOAA).

El Servicio Meteorológico Nacional cuenta con un centro de monitoreo, en el que reciben imágenes de todo el territorio mexicano, generadas pos dos satélites estadounidenses: Terra y Aqua, mismos que proporcionan diariamente 8 reportes, 4 en el día y 4 en la noche [6], la detección se realiza principalmente a través de imágenes obtenidas del sensor NOAA/16-AVHRR que proporciona 5 bandas para cada pixel, los algoritmos para el procesamiento de imágenes fueron desarrollados por la NASA y la universidad de Winsconsin-CIMSS.

Cabe mencionar que los satélites de órbita polar como el NOAA tienen una amplia resolución espacial [4], pero tienen el inconveniente de proporcionar ciclos limitados con sólo 1 o 2 pasadas por sobre el mismo lugar [4], la alternativa a este tipo de satélites es la utilización de estos como el Geostationary Operational Environmental Satellite (GOES), que proporciona un ciclo con mayores recorridos sobre el mismo lugar con el inconveniente de que la resolución espacial es de mucho menor tamaño (apenas 4 km.) [4].

Si bien los modelos mencionados anteriormente permiten en gran medida la detección de IF, aún presentan grandes desventajas como son: la dependencia de satélites estadounidenses, esto representa la desventaja de que solamente 8 imágenes son proporcionadas cada 24 horas, aunque el trabajo con satélites GOES [4] permite reducir los tiempos de espera entre cada muestra sobre un mismo punto, aun se presenta la desventaja de que los resultados son confiables solo si se cuenta con la ausencia de nubes además para imágenes nocturnas se debe de tener en cuenta el no confundir la iluminación de las ciudades con un incendio forestal.

Las desventajas anteriormente mencionadas han motivado a la utilización de otros tipos de sistemas que no utilicen imágenes provenientes de sensores de satélites, como es el caso del Sistema de Información para el Monitoreo de Incendios Forestales (SIMIF) en el estado de Puebla. Éste sistema se basa en el uso de cámaras para el monitoreo de bosques, permitiendo que estas zonas estén vigiladas las 24 horas del día. Con este tipo de sistemas se pueden obtener imágenes a intervalos de tiempo que no dependan del clima.

Como parte de la búsqueda de conocimiento dentro del área de minería de datos así como la teoría relacionada a la prevención de incendios, podemos encontrar múltiples referencias a trabajos relacionados al planteado en este artículo, dichos trabajos muestran una fuerte relación a los objetivos buscados, los cuales son encontrar características que nos permitan reducir o eliminar la posibilidad de incendios forestales, por lo que a continuación se presentan algunos trabajos relevantes. OIlero [7] presenta un sistema multi-sensorial para la detección temprana de incendios forestales, se emplean principalmente imágenes infrarrojas. Yasar [8] utiliza animales como sensores biológicos móviles para hacer la detección temprana de incendios forestales. Florent [9] propone un método totalmente automático de detección de incendios forestales a partir de imágenes satélitales sobre la base de la TIR (Thermical InfraRed). González, Hernández

y Jiménez [10], realizan un sistemas que incorpora autómatas celulares para la detección del comportamiento de los incendios forestales en un área determinada. Seng, Wynne y Kim [11], presentan un enfoque para reducir la falsas alarmas en los puntos calientes de las regiones de los incendios forestales, emplea las coordenadas geográficas de calor asi como manchas en las regiones de los incendios forestales en la detección probable de los puntos de incendio. Con el fin de determinar los patrones regulares en los derivados puntos de acceso y los clasifique como las falsas alarmas en el supuesto de que los incendios por lo general no se propagan en patrones regulares, como una línea recta.

El resto de este trabajo está estructurado de la siguiente manera: En la Sección 2 se explica con detalle la propuesta que se llevó a cabo para la realización de este trabajo, obteniendo resultados que se muestran en la Sección 3 y finalmente reportando conclusiones en la Sección 4 y con base en estas conclusiones se introduce el posible trabajo a futuro en esta misma sección.

2. Metodología propuesta

El objetivo de este trabajo es el de identificar y clasificar imágenes de incendios forestales, los pasos para la construcción de un modelo de clasificación se muestran en la figura 1. Básicamente consiste en construir una base de datos, ésta contiene la mayor cantidad de datos posibles, una vez teniendo los datos es necesario seleccionar los que realmente servirán para el objetivo planteado, transformándolos como entrada al modelo de clasificación, en este caso: árboles de clasificación y redes neuronales, para posteriormente evaluar los resultados y descubrir conocimiento.

En las siguientes secciones se describe a detalle cada una de las etapas del proceso aplicado.

2.1. Selección de los datos

Las imágenes utilizadas tienen la finalidad de proporcionar una similitud con las imágenes provenientes de las cámaras del SIMIF, esto añade como ventaja poder aplicar esta metodología no únicamente con cámaras del SIMIF si no que podrá ser utilizada por otros sistemas que trabajen con imágenes de características similares.

Para la construcción de la base de datos, se hizo necesaria la ayuda del gran repositorio que es la Web. Es decir, se obtuvieron la mayor cantidad de imágenes de incendios forestales que estuvieran disponibles en la Web mediante una consulta a un buscador. Con la finalidad de obtener una amplia colección de distintos tipos de incendios y diferentes perspectivas, con el propósito de mejorar la detección de incendios.

Por el momento, no importaba si la imagen tenía un retoque, si tenía o no un incendio forestal o si simplemente el buscador nos la proporcionaba como resultado. En la tabla 1 se describen las características de las imágenes de incendios forestales recolectadas de la Web. El objetivo de la recolección era

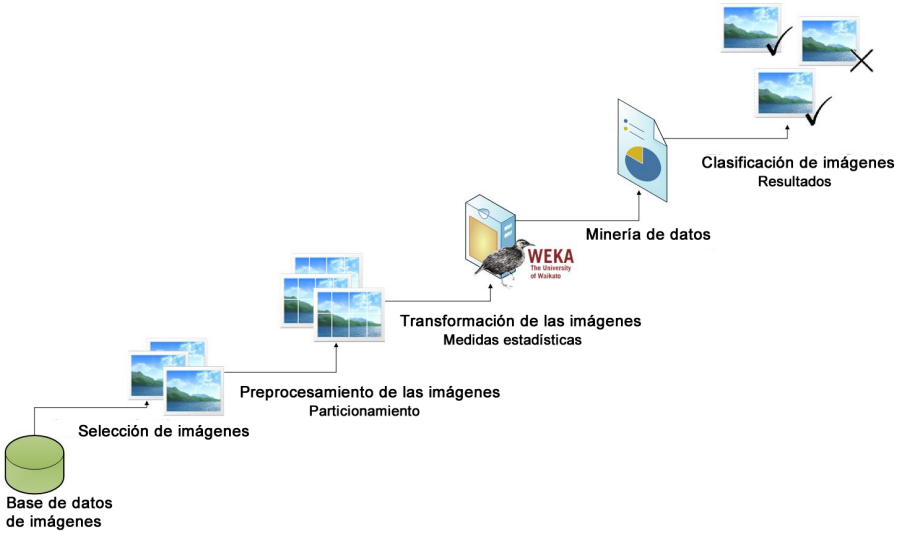

Fig. 1. Metodología para la construcción de un modelo de clasificación

tener una gran base de datos de imágenes donde estuviera inmerso un incendio forestal, que en una etapa posterior se seleccionaran las imágenes que realmente se ocuparán.

Tabla 1. Características de las imágenes de incendios forestales recolectadas de la Web

Característica	**Descripción**
Número de imágenes recolectadas	535
Imágenes con incendios de día	198
Imágenes con incendios de noche	120
Imágenes solamente con humo	100
Imágenes sin visualización de fuego o humo	117
Formato de la imagen	JEPG
Resolución promedio	400x400

2.2. Preprocesamiento de los datos

De las imágenes descargadas de la Web, se prosiguió a identificar manualmente las que realmente servirían para la tarea de identificación de incendios forestales ocupando para el propósito planteado una muestra de 98 imágenes que proporcionan información para la identificación de un incendio forestal y que constan de las características descritas en la tabla 2.

Una vez obtenida la muestra, se continuó con el particionamiento de las imágenes y este se realizó utilizando la propiedad Quadtree mediante un software extractor de características que fué implementado. El quadtree permite hacer operaciones sobre las imágenes, así como unirlas, modificar ciertos cuadrantes de la imagen, agregar/modificar colores, entre otras. Dentro de los tipos de quadtree podemos encontrar los siguientes: quadtree de puntos, quadtree de región y quadtree de cubo.

Tabla 2. Características de las imágenes de incendios forestales seleccionadas

Característica	**Descripción**
Número de imágenes seleccionadas	98
Imágenes con incendios de día	49
Imágenes con incendios de noche	49
Formato de la imagen	JEPG
Resolución promedio	400x400
Número de particiones de la imagen	8
Distribución de las particiones	100x200

De los cuales el quadtree de región es el más importante en el procesamiento digital de imágenes y es el tipo de quadtree que se utilizó para particionar la imagen. El quadtree de región representa una partición del espacio en dos dimensiones descomponiendo la región en cuatro cuadrantes iguales, subcuadrantes, y así sucesivamente con cada nodo de la hoja que contiene los datos que corresponden a una subregión específica.

Es importante mencionar que para cada partición se seleccionó de manera no automática, si dicha partición tiene fuego o no, esta característica es almacenada en la base de información como el atributo clasificador(C). Para obtener el atributo clasificador de la imagen que fue dividida en 8 particiones, se aplican las siguientes reglas:

1. Si cualquiera de las particiones obtiene un resultado positivo(por medio del software generado),entonces la imagen(las 8 particiones) se considerara como positiva
2. Si ninguna de las 8 particiones obtiene un resultado positivo, entonces la imagen real se considerara como negativa

En la figura 2 se muestra un ejemplo del proceso anteriormente descrito para la obtención de una muestra positiva y una muestra negativa.

2.3. Transformación de los datos

La obtención de resultados implica la creación de nuevo conocimiento por medio de información previamente adquirida en este caso, propiedades de imágenes

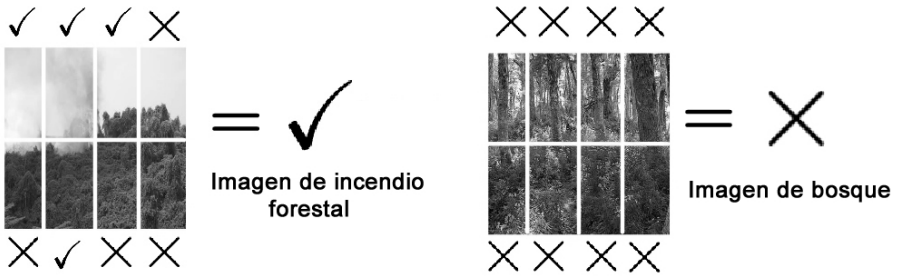

Fig. 2. Ejemplo de clasificación positiva y negativa.

de incendios forestales, para obtener dichas propiedades de manera automática, se creó un sistema extractor de características el cual permite abstraer de cada imagen, características de tipo estadísticas y de textura las cuales son usadas después por algoritmos de minería de datos, donde dicho software genera por cada imagen un quadtree de 8 elementos, de cada elemento de este quadtree se obtiene una tupla de información, abstrayendo la información de ésta y almacenándola en una base de información.

Las técnicas para la extracción de características de las imágenes, son de tipo estadístico, están basadas en el uso de histogramas $p_f(f)$ donde dejemos que f_k, $k = 1, \ldots, N$ sea los niveles de intensidad de la imagen. Las medidas de textura basadas en el histograma son las siguientes:

- Varianza de textura (σ): Medida de la dispersión de las intensidades de las regiones.

$$\sigma^2 = \sum_{k=1}^{N} (f_k - \mu)^2 p_f(f_k) \tag{1}$$

- Kurtosis (μ_4): Medidad de la cola del histograma, donde histogramas con una cola considerablemente larga corresponden a regiones con altos picos de concentración de información.

$$\mu_4 = \frac{1}{\sigma^4} \sum_{k=1}^{N} (f_k - \mu)^4 p_f(f_k) - 3 \tag{2}$$

- Media (μ): Es una estimación del nivel de intensidad en la región.

$$\mu = \sum_{k=1}^{N} f_k p_f(f_k) \tag{3}$$

- Oblicuidad(skewness) (μ_3): Medida de la simetría del histograma, la cual muestra el porcentaje de las regiones de pixeles que favorecen las intensidades.

$$\mu_3 = \frac{1}{\sigma^3} \sum_{k=1}^{N} (f_k - \mu)^3 p_f(f_k) \tag{4}$$

- Entropía (H): Medida que expresa el grado de incertidumbre que existe sobre un conjunto de datos de los cuales sólo interesa alguno en particular.

$$H = -\sum_{k=1}^{N} p_f(f_k) \ln p_f(f_k) \tag{5}$$

2.4. Minería de datos

Tomando la herramienta WEKA[1] como el software extractor de conocimientos, se definió como serán las tuplas (en la base de datos), usando como atributos a las funciones que se definieron en la sección 2.3 como se muestra en la figura 3.

$$(\underbrace{M1, M2, M3, I}_{Media}\ \underbrace{D1, D2, D3, I}_{Desviacion}\ \underbrace{E1, E2, E3, I}_{Entropia}\ \underbrace{V1, V2, V3, I}_{Varianza}\ \underbrace{S1, S2, S3, I}_{Skewness}\ \underbrace{K1, K2, K3, I}_{Kurtosis}, C)$$

Fig. 3. Representación vectorial utilizada.

De dicha formula se puede notar que cada medida se usa para cada canal del RGB, donde los valores van de 0 a 255, así como se puede denotar el valor I el cual se define como la intensidad o en otras palabras como la suma de los tres canales, éste nos muestra las características globales de la imagen. Finalmente se muestra el atributo C el cual es el atributo clasificador y puede tener los valores de positivo o negativo.

De todos los métodos de aprendizaje (o de obtención del conocimiento) los sistemas de aprendizaje basados en árboles de clasificasión son quizás el método más fácil de utilizar y de entender. Un árbol de clasificasión es un conjunto de condiciones organizadas en una estructura jerárquica, de tal manera que la decisión final se puede determinar siguiendo las condiciones que se cumplen desde la raíz hasta alguna de sus hojas. Una de las grandes ventajas de los árboles de clasificasión es que, en su forma más general, las opciones posibles a partir de una determinada condición son excluyentes. Ésto permite analizar una situación y, siguiendo el árbol de clasificasión apropiadamente, llegar a una sola acción o decisión a tomar [12].

En el caso del árbol de clasificasión se utilizó el algoritmo C4.5 genera un árbol de clasificación a partir de los datos mediante particiones realizadas recursivamente. El árbol se construye mediante la estrategia de profundidad-primero (depth-first) [12].

Las redes neuronales son modelos que intentan reproducir el comportamiento del cerebro. Como tal modelo, realiza una simplificación, averiguando cuáles son los elementos relevantes del sistema, bien porque la cantidad de información de

[1] Consultar: http://www.cs.waikato.ac.nz/ml/weka/

que se dispone es excesiva o bien porque es redundante. Una elección adecuada de sus características, más una estructura conveniente, es el procedimiento convencional utilizado para construir redes capaces de realizar una determinada tarea.

De manera concreta un perceptrón consiste en dos tipos de nodos: nodos de entrada, los cuales son usados para representar las entradas de los atributos, y los nodos de salida, los cuales son usados para representar el modelo de salida. Un perceptrón es conectado vía un enlace con peso a los nodos de salida. El enlace con peso es usado para emular la fuerza de una conexión sináptica entre neuronas. Así como en los sitemas biológicos de neuronas, el entrenamiento de un perceptrón le permite a éste adaptarse hasta que las salidas son consistentes con la verdadera salida de los ejemplos del set de entrenamiento [13].

3. Resultados

El proceso usado para obtener conocimiento de un conjunto de imágenes de incendio forestal, fué en principio particionar cada imagen en 8 Quadtree, donde de estas particiones se obtuvo un conjunto de tuplas(una por cada partición de imagen) por medio de un sistema extractor de características, teniendo dichas tuplas se experimentó primero con un árbol de decisión usando el algoritmo C 4.5, donde principalmente se usó una validación cruzada de 10 elementos y se cambiaron las propiedades de ejecución del algoritmo, cambiando las propiedades más importantes. En el caso de una red neuronal, se usó un perceptrón multicapa, ésto es, múltiples secciones o capas cada una con un conjunto de neuronas las cuales se conectan unas con otras para obtener un mejor aprendizaje, así como se eligió una función de transferencia sigmoidal, y al igual que con un árbol de decisión se hicieron pruebas cambiando el número de neuronas y de capas así como las opciones principales sobre el algoritmo.

En la tabla 3 se muestran los porcentajes de clasificación de instancias correctamente clasificadas (Instancias C.C) que se obtuvieron al aplicar árboles de decisión usando el algoritmo C4.5.

Tabla 3. Resultados al aplicar Árboles de Decisión

Instancias C.C	BinarySplits	MinNumObj	numFolds	ConfFactor	RedErrorPruning
78.699	true	2	3	0.25	false
77.2959	false	6	3	0.25	false
78.8265	**true**	**3**	**3**	**0.25**	**false**
77.9337	true	5	3	0.25	false
78.3163	true	4	3	0.25	true
75.6378	true	3	4	0.15	true
75.8929	false	2	3	0.25	false
78.8260	true	3	3	0.35	false
78.2092	true	2	3	0.40	false

En la tabla 4 se muestran los porcentajes de clasificación de Instancias C.C al usar una red neuronal tipo perceptrón.

Tabla 4. Resultados al aplicar Redes Neuronales

Instancias C.C	HiddenLayer	TrainingTime	LearningRate	Momentum	ValThreshold
84.1837	2,2	700	0.3	0.2	20
81.7602	1,1	500	0.3	0.2	20
82.9082	4,4	500	0.3	0.2	20
83.1633	4,4	1000	0.5	0.2	20
82.1429	4,4	1500	0.3	0.2	20
83.0357	15	500	0.3	0.5	20
83.4184	15,15	500	0.3	0.2	25
84.4388	**20**	**1000**	**0.3**	**0.2**	**20**
81.8878	30	500	0.3	0.2	25
83.2908	30,30	1000	0.4	0.2	20
60.7143	7,7,7,7	1000	0.1	0.2	20

Teniendo los resultados de ambas series de experimentos (red neuronal y árbol de clasificación) se obtuvieron las clasificaciones parciales, las cuales ayudaran a obtener las clasificaciones totales (la clasificación de la imagen verdadera) por medio de los pesos ajustados en el caso de la red neuronal y el árbol de clasificación obtenido en el caso del algoritmo C 4.5, en la tabla 5 se muestra el procentaje de instancias correctamente clasificadas al realizar la clasificación total.

Tabla 5. Comparativa de Clasificación verdadera

Tipo	Instancias C.C
Red Neuronal	89.795
Árbol Clasificación	87.755

Teniendo la clasificación real se hizo una evaluación de que método de aprendizaje tiene un mejor desempeño teniendo una evidencia dada. Al aplicar las técnicas de evaluación con un intervalo de confianza del 90 %, se obtiene como resultado que la red neuronal presenta una mejor clasificación con respecto al árbol de clasificaión y es estadísticamente nos acerca de manera más favorable al error verdadero.

4. Conclusiones y Trabajo a Futuro

Los incendios forestales desempeñan un papel fundamental en la transformación de los paisajes, la sucesión de la vegetación, la degradación del suelo y calidad de aire; por tal motivo este trabajo se centró principalmente en la identificación de este fenómeno. Actualmente tratar de detectar incendios forestales se realiza de diferentes maneras y se ha convertido en una tarea primordial para el cuidado y protección del medio ambiente; la técnica que se presenta en este trabajo ayudará a reconocer de una forma automática esta catastrofe que ocurre en el medio ambiente y que muchas de las veces se podria evitar si se tomaran las medidas necesarias para prevenirlos.

Por otro lado, se creó un software capaz de obtener información estadística y de textura de un set de imágenes de incendios forestales, observando que usar un particionamiento (Quadtree) ayuda a mejorar la precisión de los clasificadores que se utilizaron al realizar un modelo de clasificaión obteniendo un 89.795 % de instancias clasificadas correctamente para el caso de una red neuronal frente a un 87.755 % para el árbol de clasificación. Concluyendo que la red neuronal aportó mayor información y mayor conocimiento. De acuerdo a estos resultados se tiene que de acuerdo a los falsos positivos se consideró que es mejor que el clasificador diga que hubo un incendio forestal(positivo) cuando en realidad no lo hubo(negativo) porque de esta manera no se tiene un porcentaje de clasificación incorrecto, de acuerdo a los datos procesados.

El trabajo se centró en la detección de incendios forestales basandose en imágenes descargadas de la Web que no pertenecen a ningún sistema de monitoreo, sin embargo, en un futuro se pueden agregar imágenes a la base de datos provenientes del sistema que se vaya a utilizar para mejorar la detección de IF. Además de que se pueden utilizar otros clasificadores para ver su comportamiento en esta tarea.

Agradecimientos. Este trabajo ha sido parcialmente apoyado por el Consejo Nacional de Ciencia y Tecnología - CONACYT referencia 218862/314461.

Referencias

1. José Hernández Oralio, María José Ramírez Quintana, C.F.R.: Introducción a la Minería de Datos. Prentice Hall, Madrid (2004)
2. SEMARNAT: Secretaría de medio ambiente y recursos naturales (2010) Dependencia de Gobierno Federal.
3. INIFAP: Instituto nacional de investigaciones forestales, agrícolas y pecuarias (2010) Institución de excelencia científica y tecnologíca.
4. Viller Ruiz Lourdes, L.B.J.: Incendios forestales en México. Métodos de evaluación. Universidad Nacional Autónoma de México, México (2004)
5. SEMARNAT: Cuenta méxico con un programa de detección que permite eficientar el combate a incendios forestales (2007) Secretaría de Medio Ambiente y Recursos Naturales.

6. SMN: Detección de incendios forestales en méxico (2007) Servicio Meteorológico Nacional.
7. Ollero, J.M.D.D., B.C.: Arre: Integrated systems for early forest-rire detection. III International Conference on Forest Fire Research **2** (1998) 1977–1988
8. Sahin, Y.G.: Animals as mobile biological sensors for forest fire detection. Sensors **7** (2007) 3084–3099
9. Florent Lafarge, X.D.y.J.Z.: Forest fire detection based on gaussian field analysis. European Signal Processing Conference (EU-SIPCO) (2007)
10. González Becerra Diego Alexander, Hernández Velázquez Xochitl, J.R.I.: Simulador de incendio forestal(sif). Benemérita Universidad Autónoma de Puebla (2007)
11. Seng Chuan TAY, Wynne HSU, K.H.L.: Spatial data mining: Clustering of hot spots and pattern recognition. In: IEEE International Geo-science and Remote Sensing Symposium. Volume 6. (2003) 3685– 3687
12. Daniel Santin Gonzalez, C.P.L.: Mineria de datos: Tecnicas y Herramientas. Thompson, Madrid (2007)
13. Martin del Brío, Bonifacio; Sanz, A.: Redes Neuronales y Sistemas Borrosos. Alfa-Omega, México (2007)

Reviewing Committee

David Pinto
Ivo Humberto Pineda
Beatriz Beltran
Darnes Vilariño Ayala
Mireya Tovar
Manuel Martin
Josefina Guerrero Garcia
Juan Manuel González
Meliza Contreras González
Claudia Zepeda
Ivan Olmos
Mario Rossainz
María Josefa Somodevilla-Garcia

Impreso en los Talleres Gráficos
de la Dirección de Publicaciones
del Instituto Politécnico Nacional
Tresguerras 27, Centro Histórico, México, D.F.
Diciembre de 2014
Printing 500 / Edición 500 ejemplares