Sentiment Analysis for Real-time Applications[*]

Análisis de sentimientos para aplicaciones en tiempo real

Javi Fernández
University of Alicante
javifm@ua.es

Abstract: In this paper we present a supervised hybrid approach for Sentiment Analysis in Real-time Applications. The main goal of this work is to design an approach which employs very few resources but obtains near state-of-the-art results.
Keywords: Sentiment Analysis, Real-time Applications, Lexicon, Machine Learning

Resumen: En este artículo presentamos una aproximación híbrida supervisada para el análisis de sentimientos para aplicaciones en tiempo real. El objetivo principal de nuestro trabajo es diseñar una aproximación que emplee muy pocos recursos pero que obtenga resultados cercanos al estado de la cuestión.
Palabras clave: Análisis de sentimientos, aplicaciones en tiempo real, lexicón, aprendizaje automático

1 Introduction

Recent years have seen the birth of Social Networks and Web 2.0. They have facilitated people to share aspects and opinions about their everyday life. This subjective information can be interesting for general users, brands and organisations. However, the vast amount of information (for example, over 500 million messages per day in Twitter[1]) complicates traditional sentiment analysis systems to process this subjective information in real-time. The performance of sentiment analysis tools has become increasingly critical.

The main goal of our work is to design a sentiment analysis approach oriented to real-time applications. An approach that balances efficiency and quality. It must employ very few resources, in order to be able to process as many texts as possible. This will also make sentiment analysis more accessible for everybody. In addition, the quality of the approach should be near the state-of-the-art results. In the following sections we explain our approach in detail. Section 2 briefly describes the related work in the field and introduce our work. In Section 3 we detail the approach we propose. Finally, Section 4 concludes the paper, and outlines the future work.

2 Related Work

Two main approaches can be followed: *machine learning* and *lexicon-based* (Taboada et al., 2011; Medhat, Hassan, y Korashy, 2014; Mohammad, 2015; Ravi y Ravi, 2015). *Machine learning* approaches treat polarity classification as a text categorisation problem. Texts are usually represented as vectors of features, and depending on the features used, the system can reach better results. If a labelled training set of documents is needed, the approach is defined as *supervised* learning; if not, it is defined as *unsupervised* learning. These approaches perform very well in the domain they are trained on, but their performance drops when the same classifier is used in a different domain (Pang y Lee, 2008; Tan et al., 2009). In addition, if the number of features is big, the efficiency drops dramatically. *Lexicon-based approaches* make use of dictionaries of opinionated words and phrases to discern the polarity of a text. In these approaches, each word in the dictionary is as-

[*] This research work has been partially funded by the *University of Alicante*, *Generalitat Valenciana*, *Spanish Government*, *Ministerio de Educación, Cultura y Deporte* and *Ayudas Fundación BBVA a equipos de investigación científica 2016* through the projects *TIN2015-65100-R*, *TIN2015-65136-C2-2-R*, *PROMETEOII/2014/001*, *GRE16-01: Plataforma inteligente para recuperación, análisis y representación de la información generada por usuarios en Internet* and *Análisis de Sentimientos Aplicado a la Prevención del Suicidio en las Redes Sociales* (*ASAP*).

[1] www.internetlivestats.com/twitter-statistics

signed a score for each sentiment (e.g. positivity and negativity). To detect the polarity of a text, the scores of its words are combined, and the polarity with the greatest score is chosen. These dictionaries can be generated manually (Tong, 2001), semiautomatically from an initial seed of opinionated words (Kim, Rey, y Hovy, 2004; Baccianella, Esuli, y Sebastiani, 2010), or automatically from a labelled dataset (Jijkoun, de Rijke, y Weerkamp, 2010; Cruz et al., 2013). The major disadvantage of these approaches is the incapability to find opinion words with domain and context specific orientations, while the last one helps to solve this problem (Medhat, Hassan, y Korashy, 2014). These approaches are usually faster than machine learning ones, as the combination of scores is normally a predefined mathematical function. We decided to use a hybrid approach, trying to take advantage of the *machine learning approach* categorisation quality and the *lexicon approach* speed.

Most of the current sentiment analysis approaches employ *words*, *n-grams* and *phrases* as information units for their models, either as features for machine learning approaches, or as dictionary entries in the lexicon-based approaches. However, words and n-grams have some problems to represent the flexibility and sequentiality of human language. This is the reason why we decided to use *skipgrams*. The use of skipgrams is a technique whereby n-grams are formed (bigrams, trigrams, etc.), but in addition to using adjacent sequences of words, it also allows some words to be *skipped* (Guthrie et al., 2006). In this way, skipgrams are new terms that retain part of the sequentiality of the terms, but in a more flexible way than n-grams (Fernández et al., 2014). Note that an n-gram can be defined as a 0-skip-n-gram, a skipgram where $k = 0$. For example, the sentence *"I love healthy food"* has two word level trigrams: *"I love healthy"* and *"love healthy food"*. However, there is one important trigram implied by the sentence that was not captured: *"I love food"*. The use of skipgrams allows the word *"health"* to be skipped, providing the mentioned trigram.

3 Methodology

Our contribution consists in a hybrid approach which creates a lexicon from a labelled dataset and builds a polarity classifier from the dataset and the generated lexicon with machine learning techniques. Its architecture can be seen in Figure 1. In the following subsections we explain the different parts of our approach in detail.

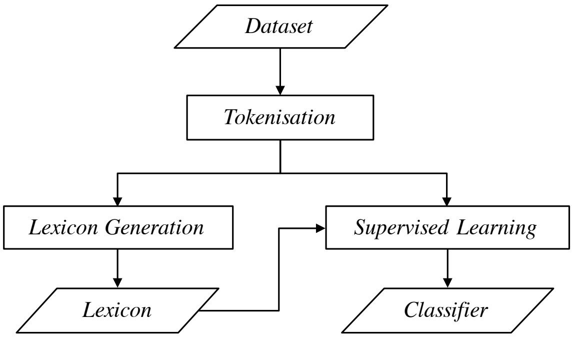

Figure 1: Approach architecture

3.1 Tokenisation

We tried to employ the minimum number of external linguistic tools, to minimise the possible propagation of external errors, in addition to the extra time they can consume. The tokenisation process starts obtaining all the words in the text. We only extract words containing alphabetic characters. Numbers, punctuation symbols, or emoticons, are not considered at this moment, but we are studying the best way to include them in the future. The only external resource we employ for the tokenisation process is a *stemmer* to obtain the most general form of the words we extracted. We preferred a stemmer over a *lemmatiser* because they are much faster (Balakrishnan y Lloyd-Yemoh, 2014) and require less resources, one of the goals of our approach. Specifically, we used the *Snowball*[2] implementation for each language.

Once we have the words in the text, we combine them using the skipgram modelling to obtain multiword terms. We will use two variables in this work: n will be the maximum number of words when building a new term with the skipgram modelling, and k will be the maximum number of skips. Note that $n = 3$ includes all the terms with 1, 2 and 3 words, and $k = 3$ includes 1, 2 and 3 skips.

3.2 Lexicon generation

In summary, our sentiment lexicon consists of a list of terms for each polarity, assigning a score indicating how strongly that term is

[2] snowball.tartarus.org

related to that polarity. To build this lexicon, we need a polarity labelled dataset, which will provide both the terms in the lexicon and their scores. There exist many term scoring techniques (Yang y Pedersen, 1997; Chandrashekar y Sahin, 2014), and the majority of them employ probabilities to calculate the scores. However, they take full advantage of the skipgram modelling, because they give the same importance to terms where words were adjacent, than to those where the words were not adjacent (we skipped some of them). Because of this, we created our custom scoring formula.

First, we will describe our *counting* formulas. In general, when we want to count the number of documents the term t occurs, we usually loop over the dataset and add 1 each time we find that term in a document. Instead, we add a value that is inversely proportional to the number of skips. This is what formulas in Equations 1 and 2 do, where D is the labelled dataset; $|D|$ is the number of documents in D, d is a document in D, D_p is the subset of documents in D labelled with polarity p, $|t|$ is the number of words in term t, and $\sigma(t,d)$ is the number of skips of term t in document d.

$$C(t) = \sum_{d \in D} [t \in d] \frac{|t|}{|t| + \sigma(t,d)} \qquad (1)$$

$$C(t,p) = \sum_{d \in D_p} [t \in d] \frac{|t|}{|t| + \sigma(t,d)} \qquad (2)$$

With this counting formulas, the number of skips is taken into account, and we can build our final scoring formula shown in Equation 3, where $s(t,p)$ is the score of term t for the polarity p, and θ is a factor that gives more relevance to terms that appear a largest number of times. This factor depends on the size and the domain of the dataset.

$$s(t,p) = \frac{C(t,p)}{C(t)} \cdot \frac{C(t,p)}{C(t,p) + \theta} \qquad (3)$$

At the end of this process we have a list of skipgrams with a score for each polarity: our sentiment lexicon. Table 1 shows an example of a dictionary built using the *Movie Reviews* dataset (Pang, Lee, y Vaithyanathan, 2002), with $n = 2$ and $k = 10$. In this example, we show only the best five terms for each polarity.

Negative	Score
this mess	.871
worst movie	.863
is terrible	.852
ludicrous	.833
waste	.818

Positive	Score
outstanding	.862
is terrific	.826
finest	.823
breathtaking	.803
is excellent	.795

Table 1: Best five terms of the dictionary built using the *Movie Reviews* dataset.

3.3 Supervised learning

We use machine learning techniques to create a model able to classify the polarity of new texts. The documents in the dataset are employed as *training instances*, and the labelled polarities are used as *categories*. However, in contrast with text classification approaches, we do not create one *feature* per term, we create a *feature* per polarity. In other words, we have the same number of features and categories. Our hypothesis is that this number of features is enough to obtain a decent system quality with a low latency. The weight of each feature is calculated as specified in Equation 4, where $w(d,p)$ is the weight of the feature for polarity p in document d.

$$w(d,p) = \sum_{t \in d} s(t,p) \cdot \frac{|t|}{|t| + \sigma(t,d)} \qquad (4)$$

Table 2 shows an example of feature weighting for the text *"worst movie ever"* using again the scores of a dictionary built using the *Movie Reviews* dataset, with $n = 2$ and $k = 10$. The final weights (positive = 1.48, negative = 3.40) will be employed as feature weights for the machine learning process.

To build our model we employed *Support Vector Machines* (SVM), as it has been proved to be effective on text categorisation tasks (Sebastiani, 2002; Mohammad, Kiritchenko, y Zhu, 2013). Specifically, we used the *Weka*[3] (Hall et al., 2009) default implementation with the default parameters (*linear kernel*, $C = 1$, $\epsilon = 0.1$).

[3] www.cs.waikato.ac.nz/ml/weka

	Positive	Negative
worst	$0.00 \cdot 1.00$	$0.79 \cdot 1.00$
movie	$0.48 \cdot 1.00$	$0.51 \cdot 1.00$
ever	$0.52 \cdot 1.00$	$0.45 \cdot 1.00$
worst movie	$0.00 \cdot 1.00$	$0.86 \cdot 1.00$
worst ever	$0.00 \cdot 1.00$	$0.59 \cdot 0.67$
movie ever	$0.47 \cdot 1.00$	$0.37 \cdot 1.00$
weight(w)	1.48	3.40

Table 2: Example of features weights for the sentence *"worst movie ever"* using the scores of a dictionary built using the *Movie Reviews* dataset with $n = 2$ and $k = 10$

4 Discussion

In this paper we presented a supervised hybrid approach for Sentiment Analysis in Twitter. We built a sentiment lexicon from a polarity dataset using statistical measures. We employed skipgrams as information units, to enrich the sentiment lexicon with combinations of words that do not appear explicitly in the text. The lexicon created was used in conjunction with machine learning techniques to create a polarity classifier.

Preliminary performance experiments have shown an acceptable speed to be employed in real-time applications[4]. Processing speeds go from $1,000$ documents per second in the worst cases (long texts, great values for n and k) to $10,000$ in the best cases (short texts, low values for n and k). These numbers are good enough to work with extensively used platforms like Twitter, where users generate over 500 million tweets per day (this is almost 6,000 tweets per second)[5].

Moreover, experiments with different datasets have also obtained promising results (Fernández et al., 2013; Fernández, Gómez, y Martínez-Barco, 2014; Fernández et al., 2014; Gutierrez, Tomas, y Fernandez, 2015; Fernández et al., 2015). Experiments with the *Movie Reviews* dataset (Pang, Lee, y Vaithyanathan, 2002) obtained an accuracy of 86.7%, with long texts in English and 2-level polarity, and 64.7% with the *TASS 2012* dataset (Villena-Román y García-Morera, 2013) for Spanish tweets and 6-level polarity.

As future work, we plan to study new methods to calculate and combine the weight of the skipgrams. We also want to add more features to the machine learning algorithm, but always trying to maintain a small number of them, in order to avoid increasing the latency. In addition, we want to include external resources and tools, such as knowledge from existing sentiment lexicons, but always focused in real-time applications. We will also extend our study to different corpora and domains, to confirm the robustness of the approach.

References

Baccianella, S., A. Esuli, y F. Sebastiani. 2010. Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. En *LREC*, volumen 10, páginas 2200–2204.

Balakrishnan, V. y E. Lloyd-Yemoh. 2014. Stemming and lemmatization: a comparison of retrieval performances. *Lecture Notes on Software Engineering*, 2(3):262.

Chandrashekar, G. y F. Sahin. 2014. A survey on feature selection methods. *Computers & Electrical Engineering*, 40(1):16–28.

Cruz, F. L., J. A. Troyano, F. Enríquez, F. J. Ortega, y C. G. Vallejo. 2013. Long autonomy or long delay? the importance of domain in opinion mining. *Expert Systems with Applications*, 40(8):3174–3184.

Fernández, J., J. M. Gómez, y P. Martínez-Barco. 2014. A supervised approach for sentiment analysis using skipgrams. En *11th International Workshop on Natural Language Processing and Cognitive Science (NAACL)*.

Fernández, J., Y. Gutiérrez, J. M. Gómez, y P. Martınez-Barco. 2014. Gplsi: Supervised sentiment analysis in twitter using skipgrams. En *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, páginas 294–299.

Fernández, J., Y. Gutiérrez, J. M. Gómez, P. Martínez-Barco, A. Montoyo, y R. Munoz. 2013. Sentiment analysis of spanish tweets using a ranking algorithm and skipgrams. En *XXIX Congreso de la Sociedad Espanola de Procesamiento de Lenguaje Natural (SEPLN 2013)*, páginas 133–142.

[4] Using a Macbook Pro 2.4 GHz i5 with 8GB RAM

[5] www.internetlivestats.com/twitter-statistics

Fernández, J., Y. Gutiérrez, J. M. Gómez, y P. Martínez-Barco. 2014. GPLSI: Supervised Sentiment Analysis in Twitter using Skipgrams. En *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, numero SemEval, páginas 294–299.

Fernández, J., Y. Gutiérrez, D. Tomás, J. M. Gómez, y P. Martínez-Barco. 2015. Evaluating a sentiment analysis approach from a business point of view.

Guthrie, D., B. Allison, W. Liu, L. Guthrie, y Y. Wilks. 2006. A Closer Look at Skip-gram Modelling. En *5th international Conference on Language Resources and Evaluation (LREC 2006)*, páginas 1–4.

Gutierrez, Y., D. Tomas, y J. Fernandez. 2015. Benefits of using ranking skip-gram techniques for opinion mining approaches. En *eChallenges e-2015 Conference, 2015*, páginas 1–10. IEEE.

Hall, M., E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, y I. H. Witten. 2009. The weka data mining software: an update. *ACM SIGKDD explorations newsletter*, 11(1):10–18.

Jijkoun, V., M. de Rijke, y W. Weerkamp. 2010. Generating focused topic-specific sentiment lexicons. En *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, páginas 585–594. Association for Computational Linguistics.

Kim, S.-m., M. Rey, y E. Hovy. 2004. Determining the Sentiment of Opinions. En *Proceedings of the 20th International Conference on Computational Linguistics (COLING 2004)*, página 1367.

Medhat, W., A. Hassan, y H. Korashy. 2014. Sentiment Analysis Algorithms and Applications: a Survey. *Ain Shams Engineering Journal*.

Mohammad, S. M. 2015. Sentiment analysis: Detecting valence, emotions, and other affectual states from text. *Emotion measurement*, páginas 201–238.

Mohammad, S. M., S. Kiritchenko, y X. Zhu. 2013. NRC-Canada: Building the State-of-the-Art in Sentiment Analysis of Tweets. En *Proceedings of the International Workshop on Semantic Evaluation (SemEval-2013)*.

Pang, B. y L. Lee. 2008. Opinion Mining and Sentiment Analysis. *Foundations and Trends in Information Retrieval*, 2(1–2):1–135.

Pang, B., L. Lee, y S. Vaithyanathan. 2002. Thumbs up? Sentiment Classification using Machine Learning Techniques. En *Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)*, numero July, páginas 79–86.

Ravi, K. y V. Ravi. 2015. A survey on opinion mining and sentiment analysis: tasks, approaches and applications. *Knowledge-Based Systems*, 89:14–46.

Sebastiani, F. 2002. Machine Learning in Automated Text Categorization. *ACM Computing Surveys (CSUR)*, 34(1):1–47, 3.

Taboada, M., J. Brooke, M. Tofiloski, K. Voll, y M. Stede. 2011. Lexicon-based methods for sentiment analysis. *Computational Linguistics*, 37(2):267–307.

Tan, S., X. Cheng, Y. Wang, y H. Xu. 2009. Adapting Naive Bayes to Domain Adaptation for Sentiment Analysis. *Advances in Information Retrieval*, páginas 337–349.

Tong, R. M. 2001. An operational system for detecting and tracking opinions in on-line discussion. En *Working Notes of the ACM SIGIR 2001 Workshop on Operational Text Classification*, volumen 1, página 6.

Villena-Román, J. y J. García-Morera. 2013. TASS 2013-Workshop on Sentiment Analysis at SEPLN 2013: An overview. En *XXIX Congreso de la Sociedad Española de Procesamiento de Lenguaje Natural (SEPLN 2013)*.

Yang, Y. y J. O. Pedersen. 1997. A comparative study on feature selection in text categorization. En *Icml*, volumen 97, páginas 412–420.