TASS 2016: Workshop on Sentiment Analysis at SEPLN, septiembre 2016, pág. 29-33

LABDA at the 2016 TASS challenge task: using word embeddings for the sentiment analysis task*

LABDA en la competición TASS 2016: utilizando vectores de palabras para la tarea de análisis de sentimiento

Antonio Quirós[1,2], Isabel Segura-Bedmar[1], and Paloma Martínez[1]

[1]Departamento de Informática, Universidad Calos III de Madrid
Avd. de la Universidad, 30, 28911, Leganés, Madrid, España
100342879@alumnos.uc3m.es, isegura,pmf@inf.uc3m.es

[2]Sngular Data&Analytics
Av. LLano Castellano 13, Planta 5, 28034 Madrid, España
antonio.quiros@sngular.team

Resumen: Este artículo describe la participación del grupo LABDA en la tarea 1 (Sentiment Analysis at global level) de la competición TASS 2016. En nuestro enfoque, los tweets son representados por medio de vectores de palabras y son clasificados utilizando algoritmos como SVM y regresión logística.
Palabras clave: Análisis de Sentimiento, Vectores de palabras

Abstract: This paper describes the participation of the LABDA group at the Task 1 (Sentiment Analysis at global level). Our approach exploits word embedding representations for tweets and machine learning algorithms such as SVM and logistics regression.
Keywords: Sentiment Analysis, Word embeddings

1 Introduction

Knowing the opinion of customers or users has become a priority for companies and organizations in order to improve the quality of their services and products. With the ongoing explosion of social media, it affords a significant opportunity to poll the opinion of many Internet users by processing their comments. However, it should be noted that sentiment analysis, which can be defined as the automatic analysis of opinion in texts (Pang and Lee, 2008), is a challenging task because it is not strange that different people assign different polarities to a given text. On Twitter, the task is even more difficult, because the texts are small (only 140 characters) and are charectized by their informal style language, many grammatical errors and spelling mistakes, slang and vulgar vocabulary and abbreviations.

Since their introduction in 2013, the TASS shared task editions have had as main goal to promote the development of methods and resources for sentiment analysis of tweets in Spanish. This paper describes the participation of the LABDA group at the Task 1 (Sentiment Analysis at global level). In this task, the participating systems have to determine the global polarity of each tweet in the test dataset. There are two different evaluations: one based on 6 different polarity labels (P+, P, NEU, N, N+, NONE) and another based on just 4 labels (P, N, NEU, NONE). A detailed description of the task can be found in the overview paper of TASS 2016 (García-Cumbreras et al., 2016). Our approach exploits word embedding representations for tweets and machine learning algorithms such as SVM and logistics regression. The word embedding model can yield significant dimensionality reduction compared to the classical Bag-Of-Word (BoW) model. The dimensionality redution can have several positive effects on our algorithms such as faster training, avoiding overfitting and better performance.

The paper is organized as follows. Section 2 describes our approach. The experimental results are presented and discussed in Section

* This work was supported by eGovernAbility-Access project (TIN2014-52665-C2-2-R).

3. We conclude in Section 4 with a summary of our findings and some directions for future work.

2 System

In this paper, we study the use of word embeddings (also known as word vectors) in order to represent tweets and then examine several machine learning algorithms to classify them. Word embeddings have shown promising results in NLP tasks, such as named entity recognition (Segura-Bedmar, Suárez-Paniagua, and Martınez, 2015), relation extraction (Alam et al., 2016), sentiment analysis (Socher et al., 2013b) or parsing (Socher et al., 2013a). A word embedding is a function to map words to low dimensional vectors, which are learned from a large collection of texts. At present, Neural Network is one of the most used learning techniques for generating word embeddings (Mikolov and Dean, 2013). The essential assumption of this model is that semantically close words will have similar vectors (in terms of cosine similarity). Word embeddings can help to capture semantic and syntactic relationships of the corresponding words.

While the well-known Bag-of-Words (BoW) model involves a very large number of features (as many as the number of non-stopwords words with at least a minimum number of occurrences in the training data), the word embedding representation allows a significant reduction in the feature set size (in our case, from million to just 300). The dimensionality reduction is a desirable goal, because it helps in avoiding overfitting and leads to a reduction of the training and classification times, without any performance loss.

As a preprocessing step, tweets must be cleaned. First, we remove all links and urls. We then remove usernames which can be easily recognized because their first character is the symbol @. We then transform the hashtags to words by removing its first character (that is, the symbol #). Taking advantage of regular expressions, the emoticons are detected and classified in order to count the number of positive and negative emoticons in each tweet and then we remove them from the text. Table 1 shows the list of positive and negative emoticons, which were taken from the wikipedia page `https://en.wikipedia.org/wiki/List\_of\_emoticons`. We convert the tweets to lowercase and replace misspelled accented letters with the correct one (for instance "à" with "á"). We also treat elongations (that is, the repetition of a character) by removing the repetition of a character after its second occurrence (for example, "hoooolaaaa" would be translated to "hola"). We then decided to take into account laughs (for instance "jajaja") which turned out to be challenging because of the diverse ways they are expressed (i.e. expressions like "jajajaja" or "jejeje" and even misspelled ones like "jajjajaaj") We addressed this using regular expressions to standardize the different forms (i.e. "jajjjaaj" to "jajaja") and then replace them with the word "risas". Finally we remove all non-letters characters and all stopwords present in tweets[1].

Orientation	Emoticons
Positive	:-), :), :D, :o), :], D:3, :c), :>, =], 8), =), :}, :ˆ), :-D, 8-D, 8D, x-D, xD, X-D, XD, =-D, =D, =-3, =3, BˆD, :'), :'), :*, :-*, :ˆ*, ;-), ;), *-), *), ;-], ;], ;D, ;ˆ), >:P, :-P, :P, X-P, x-p, xp, XP, :-p, :p, =p, :-b, :b
Negative	>:[, :-(, :(, :-c, :-<, :<, :-[, :[, :{, ;(, :-||, >:(, :'-(, :'(, D:<, D=, v.v

Table 1: List of positive and negative emoticons

Once the tweets are preprocessed, they are tokenized using the NLKT toolkit (a Python package for NLP); we also performed experimentation by lemmatizing each tweet using MeaningCloud[2] Text Analytic software to compare both approaches. Then, for each token, we search its vector in the word embedding model. We use a pretrained model (Cardellino, 2016), which was generated by using the word2vec algorithm (Mikolov and Dean, 2013) from a collection of Spanish texts with approximately 1.5 billion words. The dimension of the word embedding is 300. It

[1] http://snowball.tartarus.org/algorithms/spanish/stop.txt
[2] `https://www.meaningcloud.com/`

should be noted that these texts were taken from different resources such as Spanish Wikipedia, WikiSource and Wikibooks, but none of them contains tweets. Therefore, it is possible that the main characteristics of the social media texts (such as informal style language, noisy, plenty of grammatical errors and spelling mistakes, slang and vulgar vocabulary, abbreviations, etc) are not correctly represented in this model. One of the main problems is that there is a significant number of words (almost a 13 % of the vocabulary, representing the 6 % of words occurrences) that are not found in the model. We perform a review of a small sample of these words, showing that most of them were mainly hashtags.

In our approach, a tweet of n tokens ($T = w_1, w_2, ..., w_n$) is represented as the centroid of the word vectors $\vec{w_i}$ of its tokens, as shown in the following equation:

$$\vec{T} = \frac{1}{n}\sum_{i=1}^{n}\vec{w_i} = \frac{\sum_{j=1}^{N}\vec{w_j}.TF(w_j,t)}{\sum_{j=1}^{N}TF(w_j,t)} \quad (1)$$

where N is the vocabulary size, that is, the total number of distinct words, while $TF(w_j, t)$ refers to the number of occurrences of the j-th vocabulary word in the tweet T.

We also explore the effect of including the inverse document frequencies IDF to represent tweets (see Equation 2). This helps to increase the weight of words that occur often, but only in a few documents, while it reduces the relevance of words that occur very frequently in a larger number of texts.

$$\vec{T} = \frac{1}{n}\sum_{i=1}^{n}\vec{w_i} = \frac{\sum_{j=1}^{N}\vec{w_j}.TF(w_j,t).IDF(w_j)}{\sum_{j=1}^{N}TF(w_j,t).IDF(w_j)} \quad (2)$$

having $IDF(w_j) = \frac{log|D|}{|tw \in D : w_j \in tw|}$ where $|D|$ refers to the number of tweets.

In addition to using the centroid, we assess the impact of complementing the tweet model with the following additional features:

- posWords: number of positive words present in the tweet.
- negWords: number of negative words present in the tweet.
- posEmo: number of positive emoticons present in the tweet.
- negEmo: number of negative emoticons present in the tweet.

For the posWords and negWords features we used the iSOL lexicon(Molina-González et al., 2013), a list composed by 2,509 positive words and 5,626 negative words. As described before, for the emoticons we used the listed in Table 1, but also added to the positive ones the number of laughs detected; and also, we included the number of recommendations present in the form of a "Follow Friday" hashtag (#FF), due to its ease of detection and its positive bias.

Classification is performed using scikit-learn, a Python module for machine learning. This package provides many algorithms such as Random Forest, Support Vector Machine (SVM) and so on. One of its main advantages is that it is supported by extensive documentation. Moreover, it is robust, fast and easy to use.

As stated before, we have two main training models: Averaged centroids and the averaged centroids including the inverted document frequency, for both the lemmatized and not-lemmatized texts. We performed experiments using three different classifiers: Random Forests, Support Vector Machines and Logistic Regression because these classifiers often achieved the best results for text classification and sentiment analysis.

Also we evaluated the impact of applying a set of emoticon's rules as a pre-classification stage, similar to (Chikersal et al., 2015), in which we determine a first stage polarity for each tweet as follows:

- If posEmo is greater than zero and negEmo is equal to zero, the tweet is marked as "P".
- If negEmo is greater than zero and posEmo is equal to zero, the tweet is marked as "N".
- If both posEmo and negEmo are greater than zero, the tweet is marked as "NEU".
- If both posEmo and negEmo are equal to zero, the tweet is marked as "NONE".

Then, after the classification takes place we made three tests: i) Applying no rule, ii) honoring the polarity defined by the rule, which means, we keep the predefined polarity

if the tweet was marked as "P" or "N", otherwise we take the value estimated by the classifier, and iii) a mixed approach where we give each polarity a value (N+: -2; N: -1; NEU,NONE: 0; P: 1; P+: 2) and performed an arithmetic sum of both the predefined and estimated polarity if and only if they are not equal; with that for instance, if the classifier marked a tweet as "N" and the rules marked it as "P" the tweet will be classified as "NEU".

3 Results

In order to choose the best-performing classifiers, we use 10-fold cross-validation because there is no development dataset and this strategy has become the standard method in practical terms. Our experiments showed that, although the results were similar[3], the best settings for the 5-levels task are:

- RUN-1: Support Vector Machine, over the averaged centroids without applying any rules for pre-defining polarities.
- RUN-2: Support Vector Machine, over the averaged centroids and applying the mixed rules approach.
- RUN-3: Logistic Regression, over the centroids with inverted document frequency and applying the mixed rules approach.

and for the 3-levels task are:

- RUN-1: Support Vector Machine, over the averaged centroids and applying the mixed rules approach.
- RUN-2: Logistic Regression, over the centroids with inverted document frequency and applying the mixed rules approach.
- RUN-3: Logistic Regression, over the averaged centroids and applying the mixed rules approach.

Tables 2 and 3 show the results for these settings provided by the TASS submission system. For each run, accuracy is provided as well as the macro-averaged precision, recall and F1-measure. As expected, the results for 3 levels are higher than for 5 levels because the training dataset is larger.

[3]Experiments showed that not-lemmatized text performed better in all settings, hence the best settings reported here is using not-lematized model

Run	P	R	F1	Acc
RUN-1	0.411	0.449	0.429	0.527
RUN-2	0.412	0.448	0.429	0.527
RUN-3	0.402	0.436	0.418	0.549

Table 2: Results for Sentiment Analysis at global level (5 levels, Full test corpus)

Run	P	R	F1	Acc
RUN-1	0.506	0.510	0.508	0.652
RUN-2	0.508	0.508	0.508	0.652
RUN-3	0.512	0.511	0.511	0.653

Table 3: Results for Sentiment Analysis at global level (3 levels, Full test corpus)

With the settings mentioned above, the obtained results are extremely similar, but we can state that, in terms of Accuracy, Logistic Regression report the best results; and, even it's not measured in this work, is worth mentioning that Logistic Regression's performance was observably faster.

4 Conclusions and future work

This paper explores the use of word embeddings for the task of sentiment analysis. Instead of using, the bag-of-words model to represent tweets, these are represented as word vectors taken from a pre-trained model of word embeddings. An important advantage of word embedding model compared to the technique of bag-of-words representation is that it achieves a significant dimensional reduction of the feature set needed to represent tweets and leads, therefore, to a reduction of training and testing time of the algorithms.

In order to use word embedding models properly, a preprocessing stage had to be completed before training a classifier. Due to the unstructured nature of the tweets, this preprocessing proved to be a very important step in order to standardize at some degree the input data. The experimentation showed that the three tested classifiers obtained very similar results, with Random Forest having slight worse performance and Logistic Regression being slightly better and much more faster.

One of the main drawback of our approach is that many words do not have a word vector in the word embedding model used for our experiments. An analysis showed that many

of these words come from hashtags, which are usually short phrases. Therefore, we should apply a more sophisticated method in order to extract the words forming hashtag.

As future work, we also plan to use a word embedding model trained on a collection of text from Spanish social media. We think that this will have a positive effect of the performance of our system to identify the polarity of tweets because this model will be generated from documents characterized by the main features that describe social media texts (for example, informal style language, plenty of grammatical errors and spelling mistakes, slang and vulgar vocabulary).

Acknowledgments

This work was supported by eGovernAbility-Access project (TIN2014-52665-C2-2-R).

References

Alam, F., A. Corazza, A. Lavelli, and R. Zanoli. 2016. A knowledge-poor approach to chemical-disease relation extraction. *Database*, 2016:baw071.

Cardellino, C. 2016. Spanish Billion Words Corpus and Embeddings, March.

Chikersal, P., S. Poria, E. Cambria, A. Gelbukh, and C. E. Siong. 2015. Modelling public sentiment in twitter: using linguistic patterns to enhance supervised learning. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 49–65. Springer.

García-Cumbreras, M. A., J. Villena-Román, E. Martínez-Cámara, M. C. Díaz-Galiano, M. T. Martín-Valdivia, and L. A. U. na López. 2016. Overview of tass 2016. In *Proceedings of TASS 2016: Workshop on Sentiment Analysis at SEPLN colocated with the 32nd SEPLN Conference (SEPLN 2016)*, Salamanca, Spain, September.

Mikolov, T. and J. Dean. 2013. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*.

Molina-González, M. D., E. Martínez-Cámara, M.-T. Martín-Valdivia, and J. M. Perea-Ortega. 2013. Semantic orientation for polarity classification in spanish reviews. *Expert Systems with Applications*, 40(18):7250–7257.

Pang, B. and L. Lee. 2008. Opinion mining and sentiment analysis. *Foundations and trends in information retrieval*, 2(1-2):1–135.

Segura-Bedmar, I., V. Suárez-Paniagua, and P. Martınez. 2015. Exploring word embedding for drug name recognition. In *SIXTH INTERNATIONAL WORKSHOP ON HEALTH TEXT MINING AND INFORMATION ANALYSIS (LOUHI)*, page 64.

Socher, R., J. Bauer, C. D. Manning, and A. Y. Ng. 2013a. Parsing with compositional vector grammars. In *ACL (1)*, pages 455–465.

Socher, R., A. Perelygin, J. Y. Wu, J. Chuang, C. D. Manning, A. Y. Ng, and C. Potts. 2013b. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the conference on empirical methods in natural language processing (EMNLP)*, volume 1631, page 1642. Citeseer.