Exploring Convolutional Neural Networks for Sentiment Analysis of Spanish tweets

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

[1]Computer Science Department, Universidad Calos III de Madrid, Madrid, Spain
[2]s|ngular, Data & Analytics division, Madrid, Spain
{isegura,pmf}@inf.uc3m.es
{antonio.quiros}@sngular.team

Abstract

Spanish is the third-most used language on the Internet, after English and Chinese, with a total of 7.7% of Internet users (more than 277 million of users) and a huge users growth of more than 1,400%. However, most work on sentiment analysis has focused on English. This paper describes a deep learning system for Spanish sentiment analysis. To the best of our knowledge, this is the first work that explores the use of a convolutional neural network to polarity classification of Spanish tweets.

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 even different people often assign different polarities to a given text. Moreover, sentiment analysis can involve several Natural Language Processing (NLP) tasks such as negation or subjectivity detection, which have not been fully resolved by the NLP research community to date. On Twitter, the task is even more difficult, because the texts are small (only 140 characters) and are characterized by a informal style language utilized by users, many grammatical errors and spelling mistakes, slang and vulgar vocabulary, and plenty of abbreviations.

The shortage of training and testing data is one of the main bottlenecks for most NLP tasks in general, and for sentiment analysis in particular. This drawback has been partially overcome thanks to the organization of shared tasks such as Sentiment Analysis in Twitter Task at SemEval 2013-2015 (Nakov et al., 2013; Rosenthal et al., 2014; Rosenthal et al., 2015; Nakov et al., 2016), which provided annotated corpora of tweets for sentiment analysis. Most research efforts in this task have focused on English texts, much less attention has been given to other languages (Abdul-Mageed et al., 2011; Kapukaranov and Nakov, 2015). However, Spanish is the third language most used on the internet, with a total of 7.7% (more than 277 million of users) and a huge internet growth of more than 1,400%.

Since its introduction in 2013, the workshop on Sentiment Analysis at SEPLN (Villena-Román et al., 2013; Villena-Román et al., 2015b; Villena-Román et al., 2015a; García Cumbreras et al., 2016) has had as main goal to promote the development of methods and resources for sentiment analysis of tweets written in Spanish. In this task, the participating systems have to determine the global polarity of each tweet in the test dataset. A detailed description of the task can be found in the overview paper of TASS 2016 (García Cumbreras et al., 2016).

As said above, sentiment analysis of tweets is a very challenging task because of their small size, and thereby, they usually contain very scarce contextual information. This shortage of contextual information can be supplied by exploiting knowledge from large collections of unlabelled texts. Our approach uses a convolutional neural network (CNN), which uses word embeddings as its only input. Word embeddings can be very useful for the sentiment analysis task because they are able to represent syntactic and semantic information of words (Collobert et al., 2011; Socher et al., 2013b). We do not only experiment with randomly initialized word vectors, but also explore the use of

Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers, pages 1014–1022,
Valencia, Spain, April 3-7, 2017. ©2017 Association for Computational Linguistics

pre-trained word embeddings. We also perform a detailed exploration of the hyper-parameters of the CNN and their effect on the results.

The paper is organized as follows. In Section 2, we discuss some related work. Section 3 describes our approach. The experimental results are presented and discussed in Section 4. We conclude in Section 5 with a summary of our findings and some directions for future work.

2 Related Work

For the past two decades, a remarkable amount of NLP research has been dedicated to the sentiment analysis task. Most of early works were focused on customer reviews of products and services, while more recently researches usually carry out the task on data from social media such as tweets and user comments. Polarity classification can be performed at three different levels: document, sentence and entity level, being the first one the one most addressed until now. However, it is increasingly demanded to know the opinion of users on specific topics (entities).

Both unsupervised and supervised approaches have been used to tackle this problem, using standard features such as unigram/bigrams, word counts or binary presence features, word position, POS tags and sentiment features from polarity lexicons such as SentiWordNet (Baccianella et al., 2010), AFFIN (Hansen et al., 2011) or iSOL (Molina-González et al., 2013). Additionally, attention has recently been directed to more complex linguistic processes such as negation and speculation detection (Pang and Lee, 2008; Cruz et al., 2015).

Only a few works have explored the use of neural networks for sentiment analysis. Socher and colleagues (2011) proposed a recursive model that is able to capture the recursive nature of sentences and learn auto-encoders for multi-word phrases. Later, they proposed a matrix-vector recursive neural network model to learn compositional vector representations for phrases and sentences of any length (Socher et al., 2012). A feed-forward neural network was designed by dos Santos and Gatti (2014) to learn relevant features from characters, words and sentences for the sentiment analysis task.

In the four editions of the TASS workshop, most systems have been based on the use of popular supervised machine learning classifiers (most of them used SVM) and very extensive feature sets, which included lexical and morphosyntactic features (such as tokens, lemmas, n-grams, PoS tags) and sentiment features from the polarity lexicons (such as ElhPolar (Saralegi and San Vicente, 2013), iSOL (Molina-González et al., 2013) or AFFIN (Hansen et al., 2011)) to represent the information of each tweet. Only one of the systems (Vilares et al., 2015) proposed an approach based on deep learning, in particular, a neural network Long Short-Term Memory (LSTM) with a logistic function at the output layer. The evaluation of the task showed that this deep learning approach did not overcome the classical classifiers such as SVM. In TASS, 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). The state-of-art result for the task with 4 polarity levels is around 0.70 of accuracy. As expected, the best accuracy is lower (around 0.67) for the task with 6 polarity levels. A more in-depth analysis of the results and the different participating systems can be found in (Villena-Román et al., 2013; Villena-Román et al., 2015b; Villena-Román et al., 2015a; García Cumbreras et al., 2016).

To the best of our knowledge, convolutional networks have not been applied to the sentiment analysis of Spanish tweets yet. Several works (dos Santos and Gatti, 2014; Severyn and Moschitti, 2015) have shown that they can be a valuable approach for English tweets, and thereby, the same could be expected also for Spanish. One of the main advantages of this architecture is that it does not require syntactic information from sentences. It should be noted that many tweets are grammatically incorrect, and thereby, those methods not based on syntactic information, could give better results.

3 Approach

3.1 The General Corpus

The General corpus was created for the TASS competition. It consists of 68,000 Spanish tweets, which were collected from November 2011 to March 2012, and covers a variety of topics such as economy, communication, politics, mass media and culture. The corpus was divided into training and test sets with a 10%-90% ratio. As said above, each tweet in the training set is classified with its polarity, which can take some of

the following values: strong positive (P+), positive (P), neutral (NEU), negative (N), strong negative (N+) and one additional for tweets without polarity (NONE). The annotation process of the training set was semi-automatic using a baseline machine learning model whose annotations were manually reviewed later by human experts. Although the test set has not been released with their gold annotations, the evaluation platform of the TASS competition is still open[1] for registered users. This platform allows participants to submit new runs and then obtain their scores.

3.2 A baseline approach

To start with, we developed a baseline based on the most common approach for polarity classification at document level: the use of very popular supervised machine learning algorithms such as SVM and logistic regression. Our goal is to compare this baseline with the CNN model.

Instead of using bag-of-words (BoW) to represent tweets, we exploited word embedding representation. 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 et al., 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 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 millions to just 300). The dimensionality reduction is a desirable goal, because it helps in avoiding over-fitting and leads to a reduction of the training and classification times, without any performance loss. Moreover, word embeddings have shown promising results in NLP tasks, such as named entity recognition (Segura-Bedmar et al., 2015), relation extraction (Alam et al., 2016), sentiment analysis (Socher et al., 2013b) or parsing (Socher et al., 2013a).

As a preprocessing step, tweets must be cleaned. First, all links, urls and usernames (these last ones can be easily recognized because their first character is always the symbol @) were removed. Then, the hashtags were transformed to words by removing its first character (that is, the symbol #). Taking advantage of regular expressions, the emoticons were detected and classified in order to count the number of positive and negative emoticons in each tweet and then were removed from the text. Table 1 shows the list of positive and negative emoticons, which have been taken from Wikipedia[2]. The tweets were converted to lower-case. Moreover, the misspelled accented letters were replaced by their correct ones (for instance á with a). We also treated 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 also took into account laughs (for instance jajaja) which turned out to be challenging because of the diverse ways they are expressed (i.e. expressions like "ja", "jaja", jajajaja, "jiji" 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 replaced them with the Spanish translation of laugh: "risa". Finally we removed all non-letters characters and all stopwords present in tweets[3].

Orientation	Emoticons
Positive	:-), :), :D, :o), :], D:3, :c), :>, =], 8), =), :}, :^), :-D, 8-D, 8D, x-D, xD, X-D, XD, =-D, =D, =-3, =3, B^D, :'), :'), :*, :-*, :^*, ;-), ;), *-), *), ;-], ;], ;D, ;^), >: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 were preprocessed, they were tokenized using the NLTK toolkit (a Python pack-

[1] www.sepln.org/workshops/tass/2016/private/evaluate.php

[2] https://en.wikipedia.org/wiki/List_of_emoticons

[3] http://snowball.tartarus.org/algorithms/spanish/stop.txt

age for NLP). To represent the tweets, we used Cardellino's pre-trained model (Cardellino, 2016). This model is available for research community and was built from several Spanish collection texts such as Spanish Wikipedia (2015), the OPUS corpora (Tiedemann and Nygaard, 2004) or the Ancora corpus (Taulé et al., 2008), among others. It contains nearly 1.5 billion words (Cardellino, 2016). The dimension of its vectors is 300. Then, for each token, we searched its vector in the word embedding model. It should be noted that this model was trained on a collection of texts from different resources such as Spanish Wikipedia, WikiSource and Wikibooks, and none of them contains tweets. Therefore, it is possible that the main characteristics of the social media texts (such as informal style language, grammatical errors and spelling mistakes, slang and vulgar vocabulary, abbreviations, etc) are not correctly represented in this model. Indeed, we found that there was a significant number of words from the tweets (almost a 6%) that were not found in this word embedding model. We performed a review of a small sample of these words, showing that most of them were mainly hashtags.

In our baseline 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.

In addition to using the centroid, we completed the feature set 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.

We also applied a set of emoticon's rules as a pre-classification stage, similar to Chikersal et al. (2015), in which we determined 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, the classification of tweets was 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. Initially, 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 (García Cumbreras et al., 2016).

After the classification, 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:-1" and the rules marked it as "P:1" the tweet will be classified as "NEU:0".

In order to choose the best-performing classifiers, we used 10-fold cross-validation because there was no development dataset and this strategy has become the standard method in practical terms. Our experiments showed that, although the results were similar, the best settings were:

- SVM+MIX: Support Vector Machine and applying the mixed rules approach.
- LR+MIX: Logistic Regression and applying the mixed rules approach.

3.3 CNN Model

In this study, the CNN model proposed for sentiment analysis is based on the model for sentence classification described in Kim (2014). The model has been implemented using Google's Tensorflow toolkit. Based on the fact that single level architectures seem to provide similar performance than larger networks (Kim, 2014), we decided to design our network with only a single convolutional layer, followed by a max-pooling layer and a softmax classifier as final layer. In this way, we were able to reduce the large amount of time needed to train a CNN on a large corpus as our training set (with almost 7,000 tweets).

Each tweet was represented by a matrix that concatenates the word embeddings of its words. This matrix is the input of the network. In our experiments, we learned our word embeddings from scratch, but also we tried with two different pre-trained word2vec models: Cardellino's model, which was described above, and a pre-trained model from tweets. To train this second model of word embeddings, we used the corpus provided by the TASS organizers (68,000 tweets) as well as a very extensive collection of 8,774,487 tweets, which we collected during 2014. To do this, we used the word2vec tool, which implements the continuous bag-of-words and skip-gram architectures for computing vector representations of words (Mikolov et al., 2013). We used the continuous bag of words model with a context window of size 8. The size of the pre-trained model from tweets is 347,970 words. We randomly initialized those words that were not in the pre-trained model.

In the next layer, convolutions were performed over the word embeddings using multiple filter sizes. A filter is a sliding window of a given number of words. That is, different size windows are treated at a time. Then, a max pooling layer extracted the most important feature (in our case, the maximum value) to reduce the computational complexity. In order to avoid over-fitting, dropout regularization was also used (Srivastava et al., 2014). This process randomly drops some units and their connections from the network during training. The final layer is a softmax prediction.

We used 10-fold cross-validation for parameter tuning. Many different combinations of hyper-parameters of the neural network can be defined. Summarizing, we experimented with several variants of the model:

- CNN-rand: all words are randomly initialized and then learned during training.
- CNN-wiki: a model initialized with the word vectors from Cardellino's pre-trained model. The word embeddings as well as the other parameters are fine-tuned for training.
- CNN-tweets: the network is initialized with the pre-trained model from tweets. As the previous model, both word embeddings and the other parameters are learned during the training.

4 Results

We adopt the same metrics used in the TASS shared task, which are the accuracy and the macro-averaged version of the precision, recall and F1.

One of our main goals is to study the effect of the word embeddings on the performance of our CNN model. Table 2 compares the models based on the type of word-embeddings used as input of the network: CNN-rand, CNN-wiki and CNN-Twitter. The other parameters of the model were set as follows: dimension of word embeddings = 300, number of filters = 128, size of filters = 3,4,5 dropout=0.5, $\lambda = 0$, batch size=64 and number of epochs=200. When the classification only considers 4 levels (POS, NEU, NEG, NONE), the best results are provided by the pre-trained model built from tweets, despite being much smaller (with less than 350,000 words) than Cardellino's pre-trained model (with 1.5 million of words). Even the word vectors from scratch (CNN-rand) provided slightly higher performance than the CNN model trained with Cardellino's pre-trained model. This may be due to the language style of tweets is completely different to the usual

4 polarity levels				
Approach	**Acc**	**P**	**R**	**F1**
CNN-rand	0.544	0.467	0.465	0.466
CNN-wiki	0.528	0.431	0.438	0.434
CNN-twitter	**0.578**	0.478	0.484	**0.481**

6 polarity levels				
Approach	**Acc**	**P**	**R**	**F1**
CNN-rand	0.431	0.354	0.408	0.379
CNN-wiki	**0.442**	0.345	0.378	0.361
CNN-twitter	0.427	0.378	0.407	**0.392**

Table 2: Results for 4 polarity levels (P, N, NEU and NONE) and 6 polarity levels (P+,P,N+,N,NEU and NONE)

4 polarity levels				
Approach	**Acc**	**P**	**R**	**F1**
SVM+MIX	**0.652**	0.506	0.510	0.508
LR+MIX	**0.652**	0.508	0.508	0.508
CNN-Twitter	0.637	0.518	0.519	**0.518**

6 polarity levels				
Approach	**Acc**	**P**	**R**	**F1**
SVM+MIX	**0.527**	0.411	0.449	0.429
LR+MIX	**0.527**	0.412	0.448	0.429
CNN-twitter	0.427	0.463	0.444	**0.538**

Table 3: Results of the baseline systems and the best CNN model

one of text collections (for example Wikipedia) that make up Cardellino's corpus. As expected, the results are lower when the classification is performed using 6 polarity levels because there are less examples in the training set for the classes: P, P+ and N, N+. The pre-trained model from tweets still provides the best F1, but the best accuracy is provided by Cardellino's pre-trained model. The worst results are obtained when the word vectors are randomly initialized.

Once we evaluated the impact of word embeddings on the performance of the CNN model, we decided to focus on the rest of its parameters. The dropout regularization is parameterized by the dropout probability $\in [0, 1]$. Our experiments using 10-fold cross validation revealed that the best performance is achieved when it is set to 0.5. As expected, several experiments showed that the larger the number of training epochs used, the more accurate the model. The final number of training epochs was set to 200. The experiments also show that the learning rate is the parameter with a largest impact in the prediction performance. The best results were obtained when this parameter was set to 3. The size of filter also seems to have a significant effect on results. Thus, the model achieved better results when our setting also considered smaller size such as 2. This may be due to tweets are small texts whose average number of words is 12 (Li et al., 2011). The best results were obtained with the filter-size parameter equals to "2,3,4,5". In order to determine the best model, we performed a comprehensive series of experiments, showing that the best parameter setting is as follows: dimension of word embeddings = 300, number of filters = 300, size of filters = 2,3,4,5 dropout=0.5, $\lambda = 3$, batch size=500 and number of epochs=200.

year	**system**	**levels**	**Acc**
2014	(Hurtado and Pla, 2014)	4	0.71
		6	0.64
2015	(Hurtado et al., 2015)	4	0.72
		6	0.66
2016	(Hurtado and Pla, 2016)	4	0.72
		6	0.67
	CNN-Twitter	4	0.64
		6	0.54

Table 4: Top ranking TASS systems

category	P	R	**F1**	**Total**
N	0.539	**0.845**	0.658	15,844
NEU	0.096	0.078	0.086	1,305
NONE	0.690	0.478	0.565	21,416
P	**0.746**	0.673	**0.708**	22,233

Table 5: CNN model's results for each polarity (4 polarities)

Table 3 shows the results of our baselines (using SVM or logistic regression trained with the feature set and the rules described above). It also presents the results of our best CNN model. We can observe that our CNN model provides slightly better results in terms of F1 than SVM and logistic regression. However, in terms of accuracy, these popular algorithms overcome CNN model. Meanwhile, SVM and Logistic regression provide results that are extremely similar. It is worth mentioning that the logistic regression's performance was observably faster than the other two models. Although our CNN model worked acceptably, its performance is still far from the top ranking systems of TASS (see Table4).

Tables 5 and 6 show the scores for each polarity using the best CNN model. As expected, the lower results are obtained for those classes with less instances in the training dataset.

category	P	R	F1	Total
N	0.430	**0.625**	0.510	11,287
N+	0.471	0.441	0.455	4,557
NEU	0.094	0.132	0.110	1,305
NONE	0.640	0.564	0.600	21,416
P	0.119	0.501	0.193	1,488
P+	**0.808**	0.514	**0.628**	20,745

Table 6: CNN model's results for each polarity (6 polarities)

5 Conclusion and future work

This paper explores the use of a convolutional neural network in order to extract relevant features without the necessity of handcrafted features (such as stems, named entities, PoS tags, syntactic information, etc). Our paper shows that a convolutional neural network architecture can be an effective method for sentiment analysis of Spanish tweets. Although our performance is lower than top ranking systems of the TASS workshop, our CNN model provides promising results.

As future work, we also plan to study if the inclusion of external features (such as sentiment features from polarity lexicons) to the final layer (softmax layer) achieves to improve the results. Because the General Corpus of the TASS task covers very different domains, we will perform leave-one-domain-out cross validation in order to asses the generality of our CNN model. We also plan to explore how balanced corpora and bigger datasets could help to increase the performance classification of minority classes in the training dataset. Moreover, we would like to explore other deep learning that could effectively deal with the polarity classification of Spanish tweets.

Acknowledgments

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

References

Muhammad Abdul-Mageed, Mona T. Diab, and Mohammed Korayem. 2011. Subjectivity and sentiment analysis of modern standard arabic. In *Proceedings of the Forty-Ninth Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers*, volume 2, pages 587–591, Portland, Oregon, USA., June. Association for Computational Linguistics.

Firoj Alam, Anna Corazza, Alberto Lavelli, and Roberto Zanoli. 2016. A knowledge-poor approach to chemical-disease relation extraction. *Database*, 2016:baw071.

Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. 2010. 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 (LREC 2010)*, volume 10, pages 2200–2204, Malta, May.

Cristian Cardellino. 2016. Spanish Billion Words Corpus and Embeddings.

Prerna Chikersal, Soujanya Poria, Erik Cambria, Alexander Gelbukh, and Chng Eng Siong. 2015. Modelling public sentiment in twitter: using linguistic patterns to enhance supervised learning. In *Proceedings of the International Conference on Intelligent Text Processing and Computational Linguistics (CICLing 2015)*, pages 49–65, Cairo, Egypt, April.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12:2493–2537.

Noa P. Cruz, Maite Taboada, and Ruslan Mitkov. 2015. A machine-learning approach to negation and speculation detection for sentiment analysis. *Journal of the Association for Information Science and Technology*, 67(9):2118–2136.

Cícero Nogueira dos Santos and Maira Gatti. 2014. Deep convolutional neural networks for sentiment analysis of short texts. In *Proceedings of the Twenty-Fifth International Conference on Computational Linguistics (COLING 2014)*, pages 69–78, Dublin, Irland., August.

Miguel Ángel García Cumbreras, Eugenio Martínez Cámara, Julio Villena Román, and Janine García Morera. 2016. Tass 2015-the evolution of the spanish opinion mining systems. *The Spanish Society for Natural Language Processing journal*, 56:33–40.

Lars Kai Hansen, Adam Arvidsson, Finn Årup Nielsen, Elanor Colleoni, and Michael Etter. 2011. Good friends, bad news-affect and virality in twitter. In *Future information technology*, pages 34–43. Springer.

Lluís-F Hurtado and Ferran Pla. 2014. Elirf-upv en tass 2014: Análisis de sentimientos, detección de tópicos y análisis de sentimientos de aspectos en twitter. In *Proceedings of TASS 2014: Workshop on Sentiment Analysis at SEPLN*, pages 75–79, Gerona, Spain., September.

Lluís-F Hurtado and Ferran Pla. 2016. Elirf-upv en tass 2016: Análisis de sentimientos en twitter. In *Proceedings of TASS 2016: Workshop on Sentiment Analysis at SEPLN*, pages 35–40, Salamanca, Spain., September.

Lluís-F Hurtado, Ferran Pla, and Davide Buscaldi. 2015. Elirf-upv en tass 2015: Análisis de sentimientos en twitter. In *Proceedings of TASS 2015: Workshop on Sentiment Analysis at SEPLN*, pages 35–40, Alicant, Spain., September.

Borislav Kapukaranov and Preslav Nakov. 2015. Fine-grained sentiment analysis for movie reviews in bulgarian. In *Proceedings of the International Conference Recent Advances in Natural Language Processing*, pages 266–274, Hissar, Bulgaria, September. INCOMA Ltd. Shoumen, BULGARIA.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar, October. Association for Computational Linguistics.

Baichuan Li, Xiance Si, Michael R Lyu, Irwin King, and Edward Y Chang. 2011. Question identification on twitter. In *Proceedings of the Twentieth ACM international conference on Information and knowledge management (CIKM 2011)*, pages 2477–2480, Glasgow, UK., October.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Proceedings of the Twenty-Seventh Conference on Neural Information Processing Systems (NISP 2013)*, volume 26, pages 5–10, December.

M. Dolores Molina-González, Eugenio Martínez-Cámara, María-Teresa Martín-Valdivia, and José M. Perea-Ortega. 2013. Semantic orientation for polarity classification in spanish reviews. *Expert Systems with Applications*, 40(18):7250–7257.

Preslav Nakov, Sara Rosenthal, Zornitsa Kozareva, Veselin Stoyanov, Alan Ritter, and Theresa Wilson. 2013. Semeval-2013 task 2: Sentiment analysis in twitter. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 312–320, Atlanta, Georgia, USA, June. Association for Computational Linguistics.

Preslav Nakov, Alan Ritter, Sara Rosenthal, Fabrizio Sebastiani, and Veselin Stoyanov. 2016. Semeval-2016 task 4: Sentiment analysis in twitter. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1–18, San Diego, California, June. Association for Computational Linguistics.

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

Sara Rosenthal, Alan Ritter, Preslav Nakov, and Veselin Stoyanov. 2014. Semeval-2014 task 9: Sentiment analysis in twitter. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 73–80, Dublin, Ireland, August. Association for Computational Linguistics and Dublin City University.

Sara Rosenthal, Preslav Nakov, Svetlana Kiritchenko, Saif Mohammad, Alan Ritter, and Veselin Stoyanov. 2015. Semeval-2015 task 10: Sentiment analysis in twitter. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 451–463, Denver, Colorado, June. Association for Computational Linguistics.

Xavier Saralegi and Iaki San Vicente. 2013. Elhuyar at TASS 2013. In *Proceedings of the Workshop on Sentiment Analysis at SEPLN (TASS 2013)*, pages 143–150, Madrid, Spain., September.

Isabel Segura-Bedmar, Víctor Suárez-Paniagua, and Paloma Martínez. 2015. Exploring word embedding for drug name recognition. In *Proceedings of the Sixth International Workshop on Health Text Mining and Information Analysis*, pages 64–72, Lisbon, Portugal, September. Association for Computational Linguistics.

Aliaksei Severyn and Alessandro Moschitti. 2015. Unitn: Training deep convolutional neural network for twitter sentiment classification. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 464–469, Denver, Colorado, June. Association for Computational Linguistics.

Richard Socher, Jeffrey Pennington, Eric H. Huang, Andrew Y. Ng, and Christopher D. Manning. 2011. Semi-supervised recursive autoencoders for predicting sentiment distributions. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 151–161, Edinburgh, Scotland, UK., July. Association for Computational Linguistics.

Richard Socher, Brody Huval, Christopher D. Manning, and Andrew Y. Ng. 2012. Semantic compositionality through recursive matrix-vector spaces. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1201–1211, Jeju Island, Korea, July. Association for Computational Linguistics.

Richard Socher, John Bauer, Christopher D. Manning, and Ng Andrew Y. 2013a. Parsing with compositional vector grammars. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 455–465, Sofia, Bulgaria, August. Association for Computational Linguistics.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013b. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on*

Empirical Methods in Natural Language Processing, pages 1631–1642, Seattle, Washington, USA, October. Association for Computational Linguistics.

Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958.

Mariona Taulé, Maria Antònia Martí, and Marta Recasens. 2008. Ancora: Multilevel annotated corpora for catalan and spanish. In *Proceedings of the Sixth Language Resources and Evaluation Conference (LREC 2008)*, pages 96–101, Marrakech, Morocco., May.

Jörg Tiedemann and Lars Nygaard. 2004. The opus corpus-parallel and free: http://logos.uio.no/opus. In *Proceedings of the Second Language Resources and Evaluation Conference (LREC 2004)*, pages 1183–1186, Lisbon, Portugal., May.

David Vilares, Yerai Doval, Miguel A Alonso, and Carlos Gómez-Rodrıguez. 2015. Lys at tass 2015: Deep learning experiments for sentiment analysis on spanish tweets. In *Proceedings of TASS 2015: Workshop on Sentiment Analysis at SEPLN*, pages 47–52, Alicante, Spain., September.

Julio Villena-Román, Sara Lana Serrano, Eugenio Martínez Cámara, and José Carlos González-Cristóbal. 2013. Tass-workshop on sentiment analysis at sepln. *The Spanish Society for Natural Language Processing journal*, 50:37–44.

Julio Villena-Román, Janine Garcıa-Morera, Miguel A. Garcıa-Cumbreras, Eugenio Martınez-Cámara, M. Teresa Martın-Valdivia, and L. Alfonso Urena-López. 2015a. Overview of tass 2015. In *Proceedings of the TASS 2015 Workshop on Sentiment Analysis at SEPLN*, volume 1397, pages 13–21, Alicant, Spain, September.

Julio Villena-Román, Eugenio Martínez-Cámara, Janine García-Morera, and Salud M. Jiménez-Zafra. 2015b. Tass 2014-the challenge of aspect-based sentiment analysis. *The Spanish Society for Natural Language Processing journal*, 54:61–68.