# Ensemble classifier for Twitter Sentiment Analysis

Eugenio Martínez-Cámara[1], Yoan Gutiérrez-Vázquez[2], Javi Fernández[2], Arturo Montejo-Ráez[1], and Rafael Muñoz-Guillena[2] *

[1] Computer Science Department
University of Jaén
Campus Las Lagunillas, 23071, Jaén, Spain
[2] University of Alicante
Carretera San Vicente del Raspeig, 03690, Alicante, Spain
emcamara@ujaen.es,ygutierrez@dlsi.ua.es,javifm@ua.es,
amontejo@ujaen.es,rafael@dlsi.ua.es

**Abstract.** In this paper, we present a combination of different types of sentiment analysis approaches in order to improve the individual performance of them. These ones consist of (I) ranking algorithms for scoring sentiment features as bi-grams and skip-grams extracted from annotated corpora; (II) a polarity classifier based on a deep learning algorithm; and (III) a semi-supervised system founded on the combination of sentiment resources. By means assembling of the outputs of these approaches, we made a new evaluation in order to reach a complementation among them. The evaluations were based on the General Corpus of the *TASS* competition. The good results reached encourage us to continue studying the application of ensemble methods to resolve sentiment analysis problems.

**Keywords:** Sentiment Analysis, Deep Learning, Ensemble methods

## 1 Introduction

Textual information has become one of the most important sources of data to extract useful and heterogeneous knowledge. Texts can provide *factual information*, such as descriptions or even instructions, and *opinion-based information*, which would include reviews, emotions, or feelings. Subjective information can be expressed through different textual genres, such as blogs, forums, and reviews, but also through social networks and microblogs. Social networks like Twitter, Facebook, etc. have gained much popularity last years. These sites involve a large amount of subjective information, due to millions of users share opinions on different aspects of their everyday life. Extracting this subjective information has a great value for both general and expert users. For example, users can

* This work has been partially supported by a grant from the Fondo Europeo de Desarrollo Regional (FEDER), ATTOS project (TIN2012-38536-C03-0) from the Spanish Government.

find opinions about a product they are interested in, and companies and public figures can monitor their on-line reputation.

*Sentiment Analysis* (SA) can deal with this task; however, it is difficult to exploit it accordingly, mainly because of the short length of the tweets, the informality, and the lack of context. SA systems must be adapted to this face the challenges of this new textual genre.

In this paper, we present an ensemble classifier which makes use of different types of SA approaches in order to improve the performance of the base classifiers on the context of polarity classification of tweets written in Spanish. The base systems are polarity classifiers that have participated in the *Task 1* of the *TASS* competition[3][20]. The task is focused on the development of polarity classifiers at tweet level, which must identify six levels of polarity intensity: *strong positive* (`P+`), *positive* (`P`), *neutral* (`NEU`), *negative* (`N`), *strong negative* (`N+`) and *no sentiment* (`NONE`).

The paper is structured as follows. Next section provides related works where the main insights of each approach are exposed. The base systems of the final ensemble classifier are described in Section 3. Subsequently, in Section 4 is exposed in detail the ensemble classifier that we are exposing. Finally, the conclusions and future work are presented in Section 5.

## 2 Related Work

In order to build sentiment resources, several studies have been conducted. One of the first is the relevant work performed by Hu and Liu [8] using lexicon expansion techniques by adding synonymy and antonymy relations provided by WordNet [11]. A similar approach has been used for building WordNet-Affect [19] which expands six basic categories of emotion; thus, increasing the lexicon paths in WordNet.

Another well presented lexicon can be found in [14], where 2,496 words in Spanish are annotated into two different lexicons: *Full Strength Lexicon* and *Medium Strength Lexicon*.

Nowadays, many sentiment and opinion messages are provided by Social Medias. In it, new expression manners characterise the communication streaming across the Social Medias. That reason is very important to us, because it allows us to retrieve available information in these medias to build new types of sentiment resources.

Deep neural networks have already been used to construct polarity classifiers. The work [16] proposes a sophisticated approach where they concatenate word-level vectors with character-level vectors. These vectors feed a second convolutional network for obtaining the final polarity. In case of the approach presented in Section 3.2, linear averaging of word-level vectors is performed, reducing the amount of computation and simplifying the process. Another approach to compute the final vector for a sentence or text is to consider the parse tree and

[3] `http://www.daedalus.es/TASS2013/`

calculate the vector of a node according to the vectors of the child nodes. This approach has been successfully applied in [18], although it requires syntactic information to be available in order to train the system, so it may not be a preferred option with short texts like tweets are. Besides, it is more complex to export the system to languages other than English.

## 3 Polarity classifiers

Our proposal is based on the combination of three different polarity classifiers, so firstly we are going to describe the base systems of the final combined polarity classifier.

### 3.1 Approach I: Ranking Algorithm and Skip-grams

The first approach consists of a modified version of the ranking algorithm RA-SR [7] using bi-grams, combined with a skip-gram scorer [4]. Both approaches share the same strategy:

- From a corpora of polarity-annotated sentences, a *sentiment lexicon* is created. This lexicon assigns a different score for each term and polarity. A term can consist on a single word or several context-related words, by implementing n-gram[4] and skip-gram[5] strategies.
- A *machine learning* model is generated using the corpora and the sentiment resource created. Each text in the corpora is employed as a training instance, considering the polarities as the training categories. The features are obtained by combining the scores of the terms in each text, given by the sentiment lexicon. In both cases the algorithm used is Support Vector Machines, due to its good performance in text categorisation tasks [17].

The differences between these approaches reside basically in the term generation, the term scoring, and the features employed for the machine learning modeling. The subsequent lines explain these differences.

**Ranking Algorithm RA-SR** In this approach we use a method named *RA-SR* (using Ranking Algorithms to build Sentiment Resources) [7], which produces sentiment inventories based on senti-semantic evidence, obtained after exploring text with annotated sentiment polarity information. A wider description can be found in [7].

To generate the sentiment lexicon, each sentence in the corpora is tokenised and lemmatised into *words* and *word bi-grams*. These terms are used as nodes of a RS-SR contextual graph, where the links between two terms represent the

[4] An n-gram is a sequence of $n$ consecutive words found in text.

[5] A skip-gram is a generalisation of n-grams where words might be skipped and do not need to be consecutive.

appearance of both terms in the same text. Finally, each term is assigned a value of *positivity*, *negativity* and *objectivity* obtained by applying the *PageRank* algorithm over lexical graphs that represent each polarity respectively.

This method generates several features for a single text, such as the number of positive terms, the addition of the positive scores of the terms in the text, or the number of positive emoticons (and the same for the negative and objective polarities).

**Skip-gram Scorer** In this approach, terms are not only uni-grams and bi-grams, but also *skip-grams*. Skip-grams are a technique largely used in the field of speech processing, whereby n-grams are formed but in addition to allowing adjacent sequences of words, it also allows tokens to be *skipped* [6]. It should be noted that in this approach the terms are not lemmatised.

To create the sentiment lexicon, the scores for each skip-gram and polarity depend on their occurrences in the corpora. The score is calculated taking into account the number of skipped terms on each text, the number of occurrences, and the proportion of occurrences in a specific polarity.

In this method a feature is created for each polarity. The value of each polarity feature for a text will be calculated by adding all the scores for that polarity of the skip-grams it contains. For example, the feature called *positive* for a specific text is calculated by adding all the positive scores of the skip-grams in that text. A deep explanation of this approach can be found in [5].

### 3.2 Approach II: Word2Vec

This approach projects every tweet text to a space of fixed dimensionality where every word is semantically modeled. The text is represented as the centroid of all words in this semantic space. For representing each word as a vector we applied the deep learning algorithm known as Word2Vec. Once the tweet is represented as a vector, traditional supervised learning is applied. Word2Vec is an implementation of the representation of words architecture by means of vectors in the continuous space, based on Milokov's method of bags of words or n-grams [10]. This method has been applied to manage the semantic of words in problems such as analogy at word level or word clustering.

The main idea of the method is depicted as follows: every word is associated with an n-dimensional space whose weights are calculated from a neural network structure by using a recurrent algorithm. There are two possible topologies of the neural network based on a model called Skip-gram Model. This models computes the weights of a hidden layer using the target term $w(t)$ as input and the surrounding terms as expected output. Another model is the Continuous Bag-of-Words Model (CBOW). In this case, the prediction of a term $w(t)$ is based on a window of two up to five terms around the term $t$.

It is possible to represent the semantics of each word by using these topologies, if we have a high enough volume of data. We used deeplearning4j software[6] in order to calculate the Word2Vec model.

In order to obtain suitable vectors for each word, it is required to generate a model from a high volume of text. Thus, we downloaded XML versions of Spanish Wikipedia. Then, we extracted text included into each XML document. In this way we obtained about 2.2 GB of Spanish texts.

Therefore, this classifier [13] requires two learning steps: firstly, we generate the word vector model using Wikipedia, by means of the Word2Vec unsupervised algorithm. Thus, tweets can be vectorised in this new vector space model. The *vectoriser* module computes the centroid of all the vectors of the words in the tweet. The second step is the supervised learning phase.

This approach is supported by processing messages following the next steps. First, stop words are removed. The resulting texts are passed to the vectorisation module that generates the vectors of words using the Word2Vec model of Wikipedia in the corresponding language. Supervised learning is performed by considering only provided training samples. Word2Vec is parameterised as the following table shows, obtaining as result models with a vocabulary size of 404,916 terms since Wikipedia was used as basic resource.

| Parameter | Values |
|---|---|
| Window size | 5 terms |
| Network model | CBOW model |
| Number of dimensions | 200 |
| Hierarchical clustering | enabled |
| Min. occurrences for every word | 10 times |

### 3.3 Approach III: Combination of linguistic resources

The third approach, which we attempt to combine, is based on the joint use of several sentiment lexical resources and the exploitation of the lexical information of the jargon of Twitter, such as emoticons, exclamation marks and onomatopoeia of laugh.

The sentiment resources used for the development of the system are:

- **SentiWordNet**: It is a knowledge-base founded on the structure of WordNet [11], which links to each synset of WordNet three values that correspond to the likelihood to be positive, negative or neutral. A wider description of SentiWordNet can be read in [3].
- **Q-WordNet**: It is also based on WordNet. The authors of Q-WordNet [1] consider the polarity of a word as a quality of the different senses of a word. For this reason, they associate to each sense of WordNet a label of polarity. They use two different labels of polarity, Positive and Negative. Q-WordNet is only composed by 15,510 synsets.
- **iSOL**: List of opinion bearing words, which is composed by 2,509 positive words and 5,626 negative words. The evaluation of the list in [12] shows the quality of the list.

[6] http://deeplearning4j.org/word2vec.html

The system is developed as a pipeline of processing modules that begins with a tokenisation module and finishes with the classification of the polarity of the tweet. With the aim of clarifying the function of each module, we are going to explain each of them.

- *Tokenisation.* We developed a tokeniser for tweets written in Spanish based on the tokeniser of Christopher Potts[7].
- *Normalisation.* The users of Twitter usually employ abbreviations due to the length limitation of Twitter. The subsequent disambiguation process needs that the input text is well-formed, thus the abbreviations were expanded with the aim of facilitating the work of the disambiguation module. Misspellings are also common in tweets, so because the same reason, a spelling checker was used with the intention of correcting the misspellings of the tweets. The spelling checker is based on the GNU Aspell spelling checker.
- *Disambiguation.* We used the graph-based disambiguation algorithm UKB [2].
- *Polarity classification.* The polarity classification system attempts to rightly combine the sentiment information of the three linguistic resources, the emoticons, the exclamation marks and the onomatopoeia of laugh. The system is based on the assignation of sentiment scores to each token of the tweet, and finally adding up all the sentiment scores to reach the final polarity label. Firstly, a sentiment lexicon of emoticons was compiled, and they were labelled taking into account four levels of sentiment. Each level of sentiment was assigned a sentiment score. Secondly, the onomatopoeia of laugh is usually a signal of a positive sentiment, so the system links to each laugh token a sentiment score of 0.75. The rest of the tokens that are neither emoticons nor onomatopoeias of laugh can be sentiment tokens, so they are sought in iSOL and their corresponding synsets are searched in SentiWordNet and Q-WordNet. iSOL and Q-WordNet return a unique polarity score that can be 1 (positive) or -1 (negative), but SentiWordNet returns two continuous values of polarity, one that corresponds to the likelihood to be positive, and the other one the likelihood to be negative. The polarity score of SentiWordNet consists on calculating the difference between the positive and the negative score. The positive, negative and the neutral scores returned by each sentiment resource are added up, so each token has three polarity scores, one positive, one negative and one neutral. If the word is accompanied by an exclamation mark, then its polarity scores are augmented in 0.1 points. The final polarity score of the token is the greater of the positive and negative values. The final polarity score ($pt$) is reached by adding up the polarity score of each token, and the sentiment label returned by the classifier is determined by the Equation 1.

[7] http://sentiment.christopherpotts.net/code-data/happyfuntokenizing.py

## 4 Ensemble method and Evaluation

The main idea of ensemble methodology is to combine a set of classifiers in order to obtain more accurate estimations than can be achieved by using a single classifier [15]. Broadly speaking, the ensemble methodology attempts to learn from the errors of the base classifiers with the aim of achieving a more accurate final classifier.

$$polarity = \begin{cases} P+ & \text{if } pt > 0.6 \\ P & \text{if } 0.12 < pt \leq 0.6 \\ NEU & \text{if } -0.209 \leq pt \leq -0.05 \\ & \text{and } 0.02 \leq pt \leq 0.12 \\ N & \text{if } -0.209 < pt \geq -0.45 \\ N+ & \text{if } pt \leq -0.45 \\ NONE & \text{if } -0.05 < pt < 0.02 \end{cases} \quad (1)$$

In order to assess the effectiveness of the ensemble classifier, we performed a series of experiments over the provided datasets. The measures used are *precision* (Pr) and *recall* (R). We do not use *accuracy* because it is not a good measure for text categorisation when using an imbalanced corpus [21]. Instead, we also use the *F-score* (F1) because it represents a balance between precision and recall. Due to the fact that the classifier has to identify six classes, we used the Macro-averages measures of Precision, Recall and F1.

The results reached in TASS workshop by the three polarity classification systems described previously are shown in Table 1. We have to say that in the edition of 2014, the organisation of the workshop admitted that they calculated wrongly the recall, because they considered SA as a Information Retrieval task, which is completely wrong, but these are the official results, which we take as reference.

| System | Precision | Recall | F1 |
|---|---|---|---|
| System I | 61.60% | 61.60% | 61.60% |
| System II | 51.35% | 51.35% | 51.35% |
| System III | 31.40% | 31.40% | 31.40% |

**Table 1.** Results reached by base classifiers in TASS workshop

The three previous described systems (see Sections 3.1, 3.2 and 3.3) can be considered as base classifiers of an ensemble classifier. The results (see Table 1) and the description of the system show that the three classifiers are very different, so it is a sign that a first approach to study the level of rapport among the classifiers can be the combination of the outputs of them. A straightforward methodology to combine the outputs of several classifiers is the voting scheme, and more specifically the one that we are going to describe, which is the plurality rule voting scheme. The plurality vote is called in a wide sense “the majority

vote", and it is the most often used rule from the majority vote group. According to Kuncheva [9] the majority rule could be represented mathematically as follows:

$$\sum_{i=1}^{L} d_{i,k} = \max_{j=1}^{c} \sum_{i=1}^{L} d_{i,j} \quad (2)$$

where it is assumed that the label outputs of the classifiers are given as $c$-dimensional binary vectors $[d_{i,1}, ..., d_{i,c}] \in 0, 1, i = 1, ..., L$ where $d_{i,j} = 1$ if the base classifiers $D_i$ labels the document with label $w_j$ and 0 otherwise.

Regarding the equation 2, in our case, the set $D$ is composed by the three base classifiers described previously (see Section 3.1, Section 3.2 and Section 3.3), and the set $w$ is composed by six classes (P+, P, NEU, N, N+, NONE). In a voting system if the number of classes or labels is greater or equal to the number of classifiers, as in our case, it is possible that the result of the voting is a tie. There is not a consensus in the literature to resolve ties in voting systems, so a rule has to be defined to solve them. Our voting system has three base classifiers and has to identify 6 classes, so it is very likely that a tie can be produced during the classification process. Thus, two strategies to break the ties have been defined:

1. NEU_Default: In this case, we consider that an instance with dissimilar classification results must have a neutral semantic orientation.
2. NONE_Default: In this case, we think that the cause of the disagreement among the base classifiers is that the instance has not polarity, so the system returns as label NONE.

Before showing the results reached by the voting system, we would like to remark that the performance of an ensemble method depends on the nature of the base classifiers. If the base classifiers perform well and their classification results are not homogeneous, then the resultant ensemble method has a high likelihood of improving the performance of the base classifiers. However, if there are differences between the performance of the base classifiers, as it occurs in our case, then it is highly likelihood that the performance of the best classifier is not be improved. Table 2 shows the results reached by our two voting systems.

| **Voting system** | Macro-P. | Macro-R. | Macro-F1 |
|---|---|---|---|
| NEU_Default | 64.09% | 61.91% | 62.98% |
| NONE_Default | 58.49% | 65.50% | 61.79% |

**Table 2.** Results reached by the voting systems

Regarding the results shown in Table 2 the most adequate strategy to resolve the ties of our voting system is to label as neutral those instances that provoke disagreement among the base classifiers. The reason of this behaviour has a simple explication, in the set of possible classes ($P+, P, NEU, N, N+, NONE$) there are more sentiment classes ($P+, P, NEU, N, N+$) than no-sentiment classes ($NONE$),

so it is more likely that a disagreement could be triggered because the semantic orientation of the input text is not well-defined or it has a low intensity of polarity, thus these types of instances are good candidates to be labeled as NEU. The former assertion is corroborated by the performance of the system in each class (see Table 3).

| | **NEU_Default** | | | | | | **NONE_Default** | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | P+ | P | NEU | N | N+ | NONE | P+ | P | NEU | N | N+ | NONE |
| Precision | 81.06% | 84.78% | 19.40% | 69.97% | 53.64% | 75,68% | 81.06% | 90.14% | 2.00% | 69.97% | 53.64% | 54.11% |
| Recall | 58.43% | 90.01% | 70.15% | 67.07% | 42.22% | 45.54% | 58.43% | 93.67% | 58.83% | 67.07% | 40.22% | 74.76% |
| F1 | 67.91% | 87.32% | 30.39% | 68.49% | 45.97% | 56.86% | 67.91% | 91.87% | 3.87% | 68.49% | 45.97% | 62.78% |

**Table 3.** Results reached per each class by the voting systems

Are the base classifiers improved by the ensemble method proposed herein? To answer this question Table 1 and Table 2 should be compared. The two voting systems (NEU_Default and NONE_Default) improve the base classifiers, but the improvement is higher when the NEU class is considered as the default. Our base classifiers are very different among them, for instance, System I reached the first position in the assessment of TASS, while System III the 42nd position of 47 systems. Therefore, it is very relevant that the combination of three very dissimilar systems improve all the base classifiers. Previously, we said that if there is a big difference in the performance of the base classifiers it is highly likely that the performance of the ensemble classifier will not be improved. However, the results shown in Table 2 indicate that, the combination improves the results of the best of the base classifier, which is System I.

## 5 Conclusion and further works

The results show that the combination of several polarity classifiers allows the improvement of the base classifiers. This results encourage us to continue studying the most adequate way to combine the classification power of different methodologies. Our future work will be focused on the analysis of the resolution of ties in the voting system, because, when a tie is caused by the output $NEU, P, P+$, the system return NEU as class. We have to follow analysing what could be the best way to combine the classifiers.

## References

1. Agerri, R., Garca-Serrano, A.: Q-wordnet: Extracting polarity from wordnet senses. In: Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10). European Language Resources Association (ELRA), Valletta, Malta (may 2010)
2. Agirre, E., Soroa, A.: Personalizing pagerank for word sense disambiguation. Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics pp. 33–41 (2009), cited By (since 1996)55

3. Esuli, A., Sebastiani, F.: Sentiwordnet: A publicly available lexical resource for opinion mining. In: Proceedings of LREC. vol. 6, pp. 417–422 (2006)
4. Fernández, J., Gómez, J.M., Martínez-Barco, P.: A supervised approach for sentiment analysis using skipgrams (2014)
5. Fernández, J., Gutiérrez, Y., Gómez, J.M., Martınez-Barco, P., Montoyo, A., Munoz, R.: Sentiment analysis of spanish tweets using a ranking algorithm and skipgrams. XXIX Congreso de la Sociedad Espanola de Procesamiento de Lenguaje Natural (SEPLN 2013) pp. 133–142 (2013)
6. Guthrie, D., Allison, B., Liu, W., Guthrie, L., Wilks, Y.: A closer look at skip-gram modelling. In: Proceedings of the 5th international Conference on Language Resources and Evaluation (LREC-2006). pp. 1–4 (2006)
7. Gutiérrez, Y., González, A., Orquín, A.F., Montoyo, A., Muñoz, R.: RA-SR: Using a ranking algorithm to automatically building resources for subjectivity analysis over annotated corpora. WASSA 2013 p. 94 (2013)
8. Hu, M., Liu, B.: Mining and summarizing customer reviews. In: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 168–177. ACM (2004)
9. Kuncheva, L.L.: Combining Pattern Classifiers: Methods and Algorithms. Wiley, 2 edn. (2014)
10. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. CoRR abs/1301.3781 (2013)
11. Miller, G.A.: Wordnet: An on-line lexical database. International Journal of Lexicography 3(4), 235–312 (1990), http://ijl.oxfordjournals.org/content/3/4.toc
12. Molina-González, M.D., Martínez-Cámara, E., Martín-Valdivia, M.T., Perea-Ortega, J.M.: Semantic orientation for polarity classification in spanish reviews. Expert Systems with Applications 40(18), 7250 – 7257 (2013), http://www.sciencedirect.com/science/article/pii/S0957417413004752
13. Montejo-Ráez, A., García-Cumbreras, M.A., Díaz-Galiano, M.C.: Participacin de sinai word2vec en tass 2014. In: Proc. of the TASS workshop at SEPLN 2014 (2014)
14. Pérez-Rosas, V., Banea, C., Mihalcea, R.: Learning Sentiment Lexicons in Spanish. In: LREC. pp. 3077–3081 (2012)
15. Rokach, L.: Ensemble methods for classifiers. In: Maimon, O., Rokach, L. (eds.) Data Mining and Knowledge Discovery Handbook, pp. 957–980. Springer US (2005), http://dx.doi.org/10.1007/0-387-25465-X_45
16. dos Santos, C.N., Gatti, M.: Deep convolutional neural networks for sentiment analysis of short texts. In: Proceedings of the 25th International Conference on Computational Linguistics (COLING), Dublin, Ireland (2014)
17. Sebastiani, F.: Machine learning in automated text categorization. ACM computing surveys (CSUR) 34(1), 1–47 (2002)
18. Socher, R., Bauer, J., Manning, C.D., Ng, A.Y.: Parsing with compositional vector grammars. In: In Proceedings of the ACL conference. Citeseer (2013)
19. Strapparava, C., Valitutti, A.: WordNet Affect: an Affective Extension of WordNet. In: LREC. vol. 4, pp. 1083–1086 (2004)
20. Villena Román, J., Lana Serrano, S., Martínez Cámara, E., González Cristóbal, J.C.: TASS-Workshop on sentiment analysis at sepln (2013)
21. Yang, Y., Liu, X.: A re-examination of text categorization methods. In: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval. pp. 42–49. ACM (1999)