Sentiment Polarity Detection From Amazon Reviews: An Experimental Study

Efstratios Sygkounas[1], Giuseppe Rizzo[2], Raphaël Troncy[1]

[1] EURECOM, Sophia Antipolis, France,
{efstratios.sygkounas,raphael.troncy}@eurecom.fr
[2] ISMB, Turin, Italy,
giuseppe.rizzo@ismb.it

Abstract. With the ever increasing number of electronic commerce portals and the selling of goods on the Web, customers' reviews are usually used as means to grasp the goodness of products. Mining and understanding the polarity of reviews is therefore crucially important for future customers that seek opinions and sentiments to support their decision buying process. This paper proposes an experimental study of SentiME, our approach for extracting the sentiment polarity of a message, on the Amazon review-based corpus provided by the ESWC SSA challenge. We use an Ensemble Learning algorithm implementing five state-of-the-art classifiers that are known to well perform in the domains of tweets and movie reviews. The Ensemble Learning is trained with a Bootstrapping Aggregating process using a set of linguistic (such as ngrams), and semantics (such as dictionary-based of polarity values for emojis) features. The approach presented in this paper has been first successfully tested on the SemEval Twitter-based corpora. It has then been tested in the ESWC Semantic Sentiment Analysis 2016 challenge, where properly trained, it reaches a F-measure of 88.05% over the test set for the detection of positive and negative polarity, which ranks our approach as the first system among the ones competing in this challenge.

1 Introduction

Nowadays, people frequently purchase goods on the Web. One of the most popular web site where goods are bought is Amazon. After a purchase, the buyer is generally invited to publish a review about the product. These reviews are useful for future customers that seek opinions and sentiments to support them in their decision buying process.

Sentiment analysis has already been widely successfully applied on tweets. For instance, the SemEval Task 10 [10] is a competition where participants must classify the message polarity of tweets among three classes (positive, neutral, negative). Numerous systems have been proposed over the series of the SemEval Sentiment Analysis challenges. We have decided to further study and improve the performance of the best performing system named Webis [5], which is based on an ensemble of 4 classifiers and that won the 2015 SemEval Sentiment Analysis Task on polarity detection. Our improvements include the usage of a different training method relying on the Bootstrap Aggregating algorithm and the integration in the model of an additional state-of-the-art classifier. As outcome, we observed an improvement in F-measure score of 1% for the general tweet corpus and of 6.5% for the sarcasm corpus of SemEval 2015 [13].

Inspired by these results, we apply a similar approach to tackle the first task of the ESWC Semantic Sentiment Analysis (SSA) 2016 challenge, which is about binary polarity detection of customer reviews coming from Amazon. A participant system should take as input an XML document containing textual comments within the text tag, and should generate another XML document that adds, for each sentence, the result of the classification enclosed in a polarity tag. Precision, recall and F-measure of the detected polarity values (positive or negative) are the metrics used for evaluating the participant system on each review of the evaluation dataset. Our experimental study showed the robustness of our approach in this domain and we observed the importance of the ensemble learning and how all classifiers of our model complement each other with the broad variety of the features they implement. We also learned that the bagging algorithm introduced in our system improves the training process, avoiding over-fitting due to the random creation when sampling the training dataset. Concerning the limitations of our system, we encounter scalability problems which occurred when reviews are long (in terms of characters).

The remainder of this paper is organized as follows. In Section 2, we present some related work in the field of sentiment polarity and opinion mining across different domains such as Twitter and movie reviews. We then detail our approach implemented in the SentiME system (Section 3). We describe the corpus proposed by the SSA 2016 challenge providing some useful statistics (Section 4). We discuss the experimental study we carried out and the results that our approach has achieved (Section 5). We present some limitations and lessons learned in Section 6 and we conclude with future plans in Section 7.

2 Related Work

Numerous research efforts have been proposed for performing sentiment analysis on tweets. The SemEval series of challenges on Sentiment Analysis from tweets is the major venue where practitioners can compare their systems on a common benchmark. In this section, we further describe the top three systems of the 2015 SemEval challenge [10]: Webis [5], Unitn [11] and Lsislif [6].

Webis is the winning system of the 2015 SemEval challenge Task 10. The system implements an ensemble system composed of four classifiers, each of them having participated in previous editions of the SemEval challenges. The four classifiers are NRC-Canada [8], which ranked 1st in SemEval 2013, GU-MLT-LT [4] ranking 2nd in SemEval 2013, KLUE [9] raking 5th in SemEval 2013 and TeamX [7] ranking 1st in SemEval 2014. Webis trains those four classifiers separately. When evaluating the ensemble system, Webis uses a linear function which averages the classification distributions provided by the four sub-classifiers and produce the final classification according to the maximum value of the labels in the average classification distribution.

Unitn is a deep learning system that implements a three-step process to train model that is used for the classification. First, a neural language model trained on a large unsupervised tweet dataset is used for initializing the word embeddings. Second, a convolutional neural network is used to further refine the embeddings on a large distant supervised corpus. In the end, word embeddings and other parameters from the previous steps are used to initialize the neural network that is trained with the supervised training dataset [11].

Lsislif uses a logistic regression classifier that is trained with different weighting schema for each domain for positive and negative labels and several groups of features are extracted including lexical, syntactic, semantic, lexicon and Z score features. Z score can be considered as a standardization of the term frequency using multinomial distribution and it can distinguish the importance of each term in each class [6].

Beyond the studies of how to extract sentiment analysis from tweets, numerous research efforts have also studied the same task on reviews. For instance, [3] explains how to use NLP techniques to categorize Amazon reviews according to their sentiment. Similarly, the Stanford Sentiment System [12] has been proposed recently in the domain of movie reviews. It contains the Stanford Tree Parser, a machine-learning model that parses the input text into the Stanford Tree format and uses some existing models, some of them being trained especially for parsing tweets. The Stanford Sentiment Classifier is at the heart of the system. This classifier takes as input Stanford Trees and outputs their classification results. The Stanford Sentiment Classifier provides also useful detailed results such as classification label and classification distribution on all the nodes in the Stanford Tree. The Stanford Sentiment System is a Recursive Neural Tensor Network trained on the Stanford Sentiment TreeBank that is the first corpus with fully labeled parse trees which makes possible training a model with large and labeled dataset. This model stores the information for compositional vector representations, its size of parameters is not very large and the computation cost is empirically tested as feasible in the movie review domain. In addition, the Stanford Sentiment System captures the meaning of longer phrases and shows a great strength in classifying negative sentences. It beats the bag of word approaches when predicting fine-grained sentiment labels.

Despite the vast and mature research results for detecting the polarity of tweets, in terms of precision and scalability, we encountered scalability issues when parsing Amazon reviews due to the length of the review text. The use of conventional machine learning approaches based on linguistic and semantic features showed therefore some limitations. The Stanford Sentiment System, as we mentioned above, uses the Stanford Sentiment TreeBank which creates a large tree containing the classification for each word. This becomes troublesome when sentences become large (more than 300 characters), the computation of the final classification of the sentence taking then more time. In our approach, we have addressed this scalability issues by performing statistical sampling of the reviews making sure to respect the distributions of features and the relevancy of the final sample.

3 SentiME: A Sentiment Analysis System for Tweets

We develop the SentiME system which implements an ensemble learning of 5 classifiers [13]. It is inspired by and built upon the Webis [5] system that implements 4 state-of-the-art classifiers. We extend this system using the Boostrap Aggregating Algorithm [1] (referred as bagging) for training, and adding a fifth classifier, namely the

Stanford Sentiment System [12], that is used as an off-the-shelf classifier[3] trained with a corpus of movie reviews and best performing, from our observation, in sarcasm detection.

The Stanford Sentiment System implements a recursive neural tensor network parsed by the Stanford Tree Bank. It is significantly different from all the other classifiers used on tweets polarity prediction and it shows great performance on negative classification. Hence, the negative recall of the Stanford Sentiment System is over 90% on average that makes it trustworthy to detect negation (Table 1). We want to investigate whether the addition of this new classifier in the ensemble actually improves the SentiME system.

Corpus	Negative Recall
SemEval2014-test-gold-B	91.09
SemEval2015-gold-B	89.81

Table 1. Negative recall of the sole Stanford Sentiment System on SemEval datasets.

The classification distribution provided by the Stanford Sentiment Classifier consists of five labels: very positive, positive, neutral, negative, and very negative. Consequently, we map these five labels into the two classes expected by the SemEval Sentiment Analysis challenge for a consistent integration with our system. We only extract the root classification distribution because it represents the classification distribution of the entire tweet text. We have tested different configurations for mapping the Stanford Sentiment System classification to the conventional Positive, Negative, and Neutral classes. According to the results of these tests, we have decided to use the following mapping algorithm: very positive and positive are mapped to Positive, neutral are mapped to Neutral and negative and very negative are mapped to Negative.

There are multiple ways to do an ensemble of different systems. In the case of SentiME, we propose to use the Stanford Sentiment System as an off-the-shelf classifier that will not be re-trained. We also propose to use a bagging algorithm for boosting the training of the four other classifiers. To perform bagging, we generate new training dataset for the classifiers using uniformly sampling with replacement. In the whole procedure of selecting the documents used as training, the random selection process generates $(1-\frac{1}{e})$ unique sentences, while the remaining ones are duplicated documents. Due to the fact that bagging introduces some randomness into the training process, and that the size of the bootstrap samples are not fixed, we performed multiple experiments with different sizes ranging from 33% to 175%. We perform this process three times and get three models to test for each size. We observed that doing bagging with 150% of the initial dataset size leads to the best performance in terms of F1 score (Table 2).

To combine the results of the classifiers, we use the linear regression, averaging the sum of the classifier values by the total number of them. In detail, the linear aggregating function averages the classification distributions of each classifiers and choose the

[3] The Stanford Sentiment System is used with the default model provided by the Stanford Sentiment System.

Model	**19842(175%)**	**17007(150%)**	**14173(125%)**	**11338(100%)**
Model 1	64,76	65,45	65,26	65,05
Model 2	64,65	65,81	64,40	64,15
Model 3	64,50	65,71	64,29	65,25
	9000(80%)	**7525(66%)**	**5644(50%)**	**3780(33%)**
Model 1	63,37	63,80	64,64	62,81
Model 2	64,54	64,92	62,93	62,85
Model 3	63,54	64,67	63,85	61,65

Table 2. Experiments performed with different bagging sizes on SemEval2013-train+dev-B (11338 tweets initial size) training dataset. As model, we define the features created at the end of the training process. The percentages represent the percentage of the sentences chosen from the initial dataset during the bagging process.

polarity which holds the maximum value among the average classification distributions as the final classification.

We performed four different experiments to evaluate the performance of SentiME compared to a baseline system. The corpora used for this experiment are SemEval2014-test, SemEval2014-sarcasm, SemEval2015-test and SemEval2015-sarcasm (Table 3).

1. Baseline system: this is the Webis system using re-trained models as explained in Section 2;
2. SentiME system: this is the ensemble system composed of the four sub-classifiers used by Webis plus the Stanford Sentiment System. The ensemble uses a bagging approach for the training phase;
3. Baseline system without TeamX;
4. SentiME system without TeamX;

The third and fourth experiments are variations of the first two experiments respectively where we simply remove the TeamX classifier, based on the observation that this particular classifier plays a similar role than the Stanford Sentiment system.

Table 3 reports the F-measure scores for the four different setups we described above on four different datasets: two datasets contain regular tweets and two datasets contain sarcasm tweets. We evaluated the four sub-classifiers on SemEval2014 test data set, SemEval2014 sarcasm data set, SemEval2015 test data set and SemEval2015 sarcasm data set in order to figure out whether the Stanford Sentiment System has significant impacts on the performance of our ensemble system. The last row of the Table 3 presents the F-measure scores of the baseline system as reported in the authors' paper [5].

We observe that the SentiME system outperforms the baseline system on all datasets except on the SemEval2014-test, in which the SentiME system without the TeamX subclassifier has almost the same performance than the Webis Replicate system. Concerning the performance on both sarcasm datasets, it is clear that that the SentiME system improves the F1 score by respectively 2.5% and 6.5% on SemEval2014-sarcasm and SemEval2015-sarcasm datasets.

We notice that some features used in TeamX come from the Stanford NLP Core package and we assume that TeamX shares some common characteristics with the Stan-

System	**SemEval2014-test**	**SemEval2014-sarcasm**	**SemEval2015-test**	**SemEval2015-sarcasm**
Baseline	**69,31**	60,00	66,57	54,19
SentiME system	68,27	**62,57**	**67,39**	**60,92**
Baseline without TeamX	68,56	62,04	66,19	56,86
SentiME system without TeamX	69,27	62,04	66,38	58,92

Table 3. F1 scores for the four systems on four different datasets. Highest scores are in bold

ford Sentiment System. Since the idea of our experiments was to figure out what benefits the Stanford Sentiment System can bring to our replicate system, we consider it is reasonable to exclude the TeamX sub-classifier from our replicate system.

The complete workflow of the SentiME system is depicted in Figure 1. The SentiME system trains the four sub-classifiers independently and the results are stored into four different attribute files. Then, concerning the test process, each classifier reads the attribute files as well as the test dataset. The Stanford Sentiment System just reads the test dataset. The final step is to average the five different classification results of the classifiers in order to derive the final classification result.

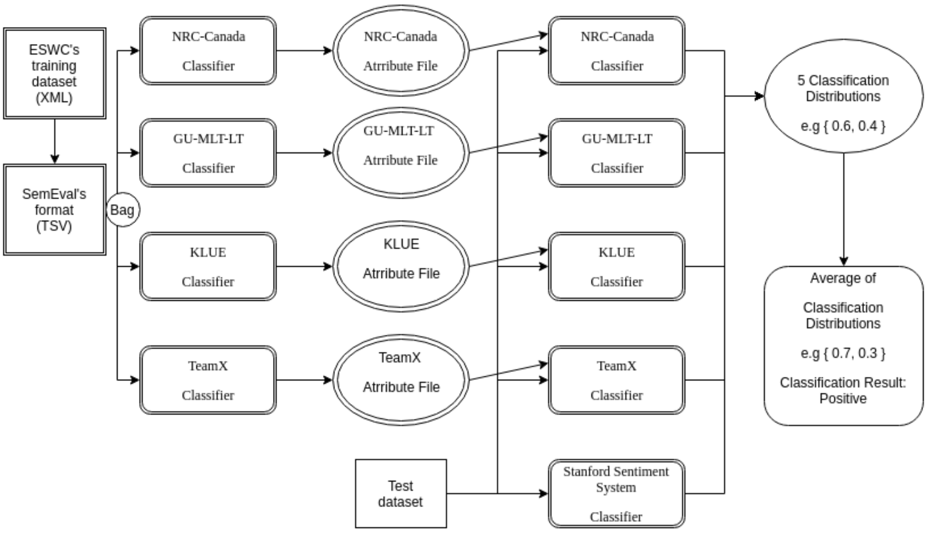

Fig. 1. Workflow of training and test for SentiME

4 The Semantic Sentiment Analysis Challenge Corpus

The training corpus of the SSA challenge 2016 [2] consists of one million Amazon reviews that are split in twenty different categories: Amazon Instant Video, Automotive, Baby, Beauty, Books, Clothing Accessories, Electronics, Health, Home Kitchen, Movies TV, Music, Office Products, Patio, Pet Supplies, Shoes, Software, Sports Outdoors, Tools Home Improvement, Toys Games, and Video Games. In each one of these twenty categories, there are ten xml files. There is a perfect balance between positive and negative reviews because each category contains five xml files of 5000 positive reviews each and another five xml files of 5000 negative reviews each. Consequently, each of the 20 category has 50000 reviews, 25000 positive and 25000 negative.

The training dataset consists of XML files. The structure of these files is as follows:

```
<?xml version="1.0" encoding="UTF−8" standalone="yes"?>
<Sentences>
    <sentence id="B000PVAFT0−A394FOR62LMMXO−1325203200−1">
        <domain>amazon_instant_video</domain>
        <polarity>positive</polarity>
        <summary>
        IMAX
        </summary>
        <text>
        This product works great and would definitely buy more of. I would definitely do
             business with this company again. Very Happy.
        </text>
    </sentence>
    ...
    ...
</Sentences>
```

For each sentence, the file contains a sentence id, a polarity result, and the textual content of the review. We parse those XML files using a DOM parser and we create one file in the TSV format (following the structure of the SemEval format) that contains the one million reviews (Table 4). Then, we use this TSV file to perform the experiment. In the end, there is a conversion of the TSV file to the XML format so that the classification of the SentiME system produces can be evaluated against a gold standard.

Sentence ID	Polarity	Sentence (Review)
B000PVAFT0-A394FOR62LMMXO-1325203200-1	positive	This product works great ...
B00004OCL2-A3T7V207KRDE2O-1227398400-841	positive	We find it perfect for ...
B0002ZQB4M-A5IJ8JA8TMU57-1287705600-1701	negative	It’s a fine pen, but ...

Table 4. Dataset structure in the TSV format.

In the initial corpus, we encountered some problems while parsing the sentences due to Unicode encoding issues. We contacted the organisers letting them know about the problem and they fixed the corpus. We observed 73 duplicated sentences, i.e. sentences repeated more than once in the dataset. There are also two sentence ids with a blank review. We performed some computations on the corpus concerning the average number of words per sentence and the average number of characters per sentence. We perform these computations for both the positive and the negative sentences (Table 5).

Corpus	**Number of sentences**	**Average number of words per sentence**	**Average number of characters per sentence**
Full corpus	1M	90.39	489
Positive Sentences	500K	86.04	466
Negative Sentences	500K	94.75	513

Table 5. Statistics for the training dataset of 1 Million Amazon Reviews.

5 Experimental Study with the SSA Corpus

In this section, we present our experimental setup and we discuss the different experiments we performed to evaluate the performance of SentiME with the SSA dataset. All the dataset configurations used in the following experiments keep a perfect balance of positive and negative sentences (50%), since we applied a stratified sampling according to the polarity. We keep this balance since each category of the training dataset has equally positive and negative sentences.

We started our experiments by finding the best mapping for the Stanford Sentiment System. The original classification distribution provided by the Stanford Sentiment Classifier consists of five labels (very positive, positive, neutral, negative and very negative). For the SemEval experiments, we have mapped very positive and positive as Positive, neutral as Neutral and negative and very negative as Negative. According to the SSA's rules, the classification distribution should be binary (Positive or Negative). We performed a 10-fold cross validation with smaller random samples of the dataset by selecting sentences with the average length of 250 words. We preserved the 50% balance, in a stratified setting. The result of this experiment mapped very positive and positive as Positive, and negative and very negative as Negative with a hybrid mapping for neutral which had the best performance regarding the final F-measure scores. We name hybrid mapping the case when neutral confident score is the greatest among the five confident scores of Stanford Sentiment System. In this case neutral is classified as Positive when the confident score of very positive and positive is greater than the confident score of negative and very negative. If it is not greater, the sentence is classified as Negative.

Figure 2 shows how we implemented the bagging process. Initially, we have the full corpus. Then, we translated the Amazon file which contains the percentage of sentences of the initial corpus which has been selected. Finally, we parse this "new" dataset to each sub-classifier separately.

We experimented with different sizes of the training set for the bagging algorithm and we empirically assessed that the best performing setting of SentiME was with the

	VP + P + Neutral = positive & VN +N = negative	**VP + P = positive & VN +N + Neutral= negative**	**VP + P = positive & VN +N= negative (Hybrid)**
Average F1 score of 10-fold	0,870839208	0,871577973	**0,872691237**

Table 6. Performance of Stanford Sentiment System with different mapping in a 10-fold cross validation setup.

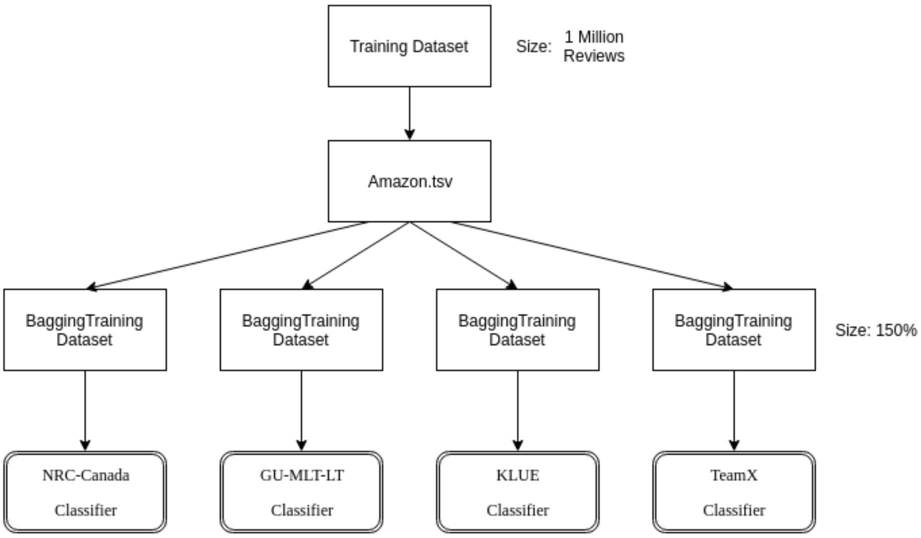

Fig. 2. Workflow for training SentiME with bagging

150% of the corpus size as shown in Figure 2. In detail, we used the SSA corpus in a 4-fold cross validation setting. We trained different models doing bagging for 150%, 120%, 100%, 90% and without bagging. We obtain a similar result when experimenting on the SemEval datasets [13]. Doing bagging for 150% of the initial training dataset yields the best performance for the same test dataset regarding the final F1 score.

The final experiment is about the F-measure scoring of SentiME in three different training datasets and two test datasets. We divided the training dataset in four parts in order to do a 4-fold cross-validation. We preferred 4-fold cross validation over a 10-fold cross validation because of the scalability issues we encountered. In each fold, we created three training datasets of ten thousand, fifty thousand and one hundred thousand sentences randomly selected among the 750,000 sentences we had available for training each time. Concerning the test datasets, we created two in each fold. The first was of one thousand and the second was of ten thousand sentences randomly selected from the 250,000 sentences we had available for testing in each fold. The experimental results are presented in Table 7.

Training–Test	**Fold 1**	**Fold 2**	**Fold 3**	**Fold 4**
10K–1K	0.8824	0.8964	0.8738	0.8784
50K–10K	0.8954	0.8969	0.8992	0.9011
100K–10K	0.9042	0.9065	0.9095	**0.9113**

Table 7. Results of experiment in a 4-fold cross validation setup.

Finally, we report the performance of SentiME over the test set in Table 8.

Team	Precision	Recall	F1 score
1st SentiME	0.85686	0.90541	**0.88046**
2nd App2Check	0.82777	0.90789	0.87142
3rd IRS-MDSA	0.81837	0.89198	0.85359
4th EPOM	0.50494	0.81665	0.62403

Table 8. Performance of teams competing on the SSA 2016 corpus.

6 Lessons Learned from the Experimental Study

The SentiME system has demonstrated consistent results in the experimental setup. It seems to be agnostic to textual variations as it is the case in the sentences collected from a broad variety of categories. The addition of the Stanford Sentiment System in our ensemble improves the performance on sarcasm sentences, that are quite common when conveying sentiments. This happens because the native Stanford Sentiment System has a great strength to classify sentences whose golden standard is negative. This means that we can use Stanford Sentiment System to help our system to classify the sarcasm sentences. On the other hand, Stanford Sentiment System is heavily skewed towards negative. We address this problem in using the Bootstrap Aggregation Algorithm (bagging) with the size of 150% of the original training set.

Amazon reviews are much longer in terms of number of characters (max 5000 characters) than tweets (max 140 characters). This transition made our system slower to compute the final classification due to the longer length of sentences. The limitation of our system is that the computing time for a big number of long sentences is huge. In the SSA training set, reviews consist of more than 3,000 characters while the average number of characters per sentence is 489 (Table 5). This number is almost four times bigger than the maximum tweet length. This has significantly challenged the computing time of the SentiME system as whole. In particular, the longer length of the content to analyze has negatively impacted the Stanford Sentiment System which uses the Stanford Tree Bank making the parsing of larger sentences slower. We need more processing power and to rely on parallel computing in order to make this work faster to scale (which was out of the scope of this experimental study).

We experiment with three different Stanford Sentiment System's mappings to the two classes only expected by the SSA challenge. While we did not observe significant different in the F-measure score, we chose the one with the slightly better performance. The best mapping is the one we called "hybrid" which classifies a sentence with the bigger confidence score neutral in Standford Sentiment System, as positive or negative according to the dominant confident scores of very positive + positive and very negative + negative.

We tried to train models on the complete training dataset, but it results to be unfeasible. We decided to build training model based on smaller dataset, built by sampling reviews from the entire set in a statistical manner and preserving the distributions of the classes.

7 Conclusions and Future Work

We have presented an experimental study aiming to evaluate the performance of SentiME, a sentiment analysis system we developed, on both tweets and Amazon reviews. The results show the robustness of our system that implements an ensemble learning approach of 5 state-of-the-art classifiers, working on different feature sets and trained with different training set (by using the bagging algorithm). After doing multiple experiments in different datasets and setups, we observe that implementing a bagging technique in the initial dataset with the percentage of 150% improves the training process. The integration of the Stanford Sentiment System in SentiME improves the results in sarcasm corpora by 6,5%. This improvement indicates that the Stanford Sentiment System deals better with sarcasm and this is due to the very good classification of negative sentences.

As future work, we will perform a thorough error analysis in the test dataset and we will investigate why some sentences have been wrongly classified. We aim to play with the features used in each of the sub-classifiers in order to better understand their respective contribution in the classification result. Moreover, we will experiment with different aggregation functions such as a weighting schema that would give priority to a particular classifier. Last but not least, we will investigate further how to improve the scalability of the Stanford Sentiment System.

Acknowledgments

The authors would like to thank Xianglei Li for his earlier work on the SentiME system. This work was partially supported by the innovation activity 3cixty (14523) of EIT Digital and by the European Union's H2020 Framework Programme via the FREME Project (644771).

References

1. Breiman, L.: Bagging Predictors. Machine Learning 24(2), 123–140 (1996)
2. Dragoni, M., Tettamanzi, A., Pereira, C.: DRANZIERA: An Evaluation Protocol For Multi-Domain Opinion Mining. In: 10^{th} International Conference on Language Resources and Evaluation (LREC) (2016)
3. Fang, X., Zhan, J.: Sentiment analysis using product review data. Journal of Big Data 2(5) (2015)
4. Günther, T., Furrer, L.: GU-MLT-LT: Sentiment Analysis of Short Messages using Linguistic Features and Stochastic Gradient Descent. In: 7^{th} International Workshop on Semantic Evaluation (SemEval) (2013)
5. Hagen, M., Potthast, M., Büchner, M., Stein, B.: Webis: An Ensemble for Twitter Sentiment Detection. In: 9^{th} International Workshop on Semantic Evaluation (SemEval) (2015)
6. Hamdan, H., Bellot, P., Bechet, F.: Lsislif: Feature Extraction and Label Weighting for Sentiment Analysis in Twitter. In: 9^{th} International Workshop on Semantic Evaluation (SemEval) (2015)
7. Miura, Y., Sakaki, S., Hattori, K., Ohkuma, T.: TeamX: A Sentiment Analyzer with Enhanced Lexicon Mapping and Weighting Scheme for Unbalanced Data. In: 8^{th} International Workshop on Semantic Evaluation (SemEval) (2014)
8. Mohammad, S., Kiritchenko, S., Zhu, X.: NRC-Canada: Building the State-of-the-Art in Sentiment Analysis of Tweets. In: 7^{th} International Workshop on Semantic Evaluation (SemEval) (2013)
9. Proisl, T., Greiner, P., Evert, S., Kabashi, B.: KLUE: Simple and robust methods for polarity classification. In: 7^{th} International Workshop on Semantic Evaluation (SemEval) (2013)
10. Rosenthal, S., Nakov, P., Kiritchenko, S., Mohammad, S., Ritter, A., Stoyanovm, V.: SemEval-2015 Task 10: Sentiment Analysis in Twitter. In: 9^{th} International Workshop on Semantic Evaluation (SemEval) (2015)
11. Severyn, A., Moschitti, A.: UNITN: Training Deep Convolutional Neural Network for Twitter Sentiment Classification. In: 9^{th} International Workshop on Semantic Evaluation (SemEval) (2015)
12. Socher, R., Perelygin, A., Wu, J.Y., Chuang, J., Manning, C., Ng, A., Potts, C.: Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank. In: Conference on Empirical Methods in Natural Language Processing (EMNLP) (2013)
13. Sygkounas, E., Rizzo, G., Troncy, R.: A Replication Study of the Top Performing Systems in SemEval Twitter Sentiment Analysis. In: 15^{th} International Semantic Web Conference (ISWC) (2016)