aueb.twitter.sentiment at SemEval-2016 Task 4: A Weighted Ensemble of SVMs for Twitter Sentiment Analysis

Stavros Giorgis, Apostolos Rousas
John Pavlopoulos, Prodromos Malakasiotis and **Ion Androutsopoulos**
NLP Group, Department of Informatics
Athens University of Economics and Business, Greece
Patission 76, GR-104 34 Athens, Greece
http://nlp.cs.aueb.gr

Abstract

This paper describes the system with which we participated in SemEval-2016 Task 4 (Sentiment Analysis in Twitter) and specifically the Message Polarity Classification subtask. Our system is a weighted ensemble of two systems. The first one is based on a previous sentiment analysis system and uses manually crafted features. The second system of our ensemble uses features based on word embeddings. Our ensemble was ranked 5th among 34 teams. The source code of our system is publicly available.

1 Introduction

This paper describes the system with which we participated in SemEval-2016 Task 4 (Sentiment Analysis in Twitter) and specifically the Message Polarity Classification subtask (Nakov et al., 2016). In this subtask, each tweet is classified as expressing a positive, negative, or no opinion (neutral). Our system is a weighted ensemble of two systems. The first one is based on a previous sentiment analysis system (Karampatsis et al., 2014) and uses manually crafted features. The second system of our ensemble uses features based on word embeddings (Mikolov et al., 2013; Pennington et al., 2014). Our ensemble was ranked 5th among 34 teams.

Section 2 discusses the datasets we used to train and tune our ensemble. Sections 3 and 4 describe our ensemble and its performance respectively. Finally, Section 5 concludes and discusses future work.

2 Data

For system training and tuning we used 19,305 tweets from the 2016 datasets provided by the organisers of SemEval-2016 Task 4, as well as data from SemEval-2013 Task 2. Specifically, the datasets were:

- $\text{TW}_{train16}$: train data for SemEval-2016 Task 4,
- TW_{dev16} : development data for SemEval-2016 Task 4,
- $\text{TW}_{devtest16}$: dev-test data for SemEval-2016 Task 4,
- $\text{TW}_{train13}$: train data for SemEval-2013 Task 2,
- TW_{dev13} : development data for SemEval-2013 Task 2.

The organisers also provided 6,908 tweets from old SemEval data, to allow system evaluation during development. These data could not be used directly for training or tuning and were the following:

- $\text{TW}_{devtest13}$: dev-test data for SemEval-2013 Task 2,
- $\text{TW}_{devtest14}$: dev-test data for SemEval-2014 Task 9,
- $\text{TW}_{sarcasm14}$: tweets containing sarcasm,
- SMS_{13} : SMS messages from 2013,
- LJ_{14} : messages from Live Journal.

Proceedings of SemEval-2016, pages 96–99,
San Diego, California, June 16-17, 2016.

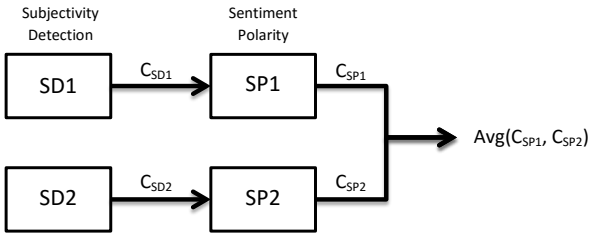

Figure 1: Ensemble of two sentiment polarity classifiers, SP1 and SP2, which are influenced by two subjectivity detection classifiers, SD1 and SD2, respectively.

3 System Overview

The main objective of SemEval-2016 Task 4 is to detect sentiment polarity, i.e., to identify whether a message (tweet) expresses positive, negative or no sentiment at all. We used a weighted ensemble of two sentiment polarity classifiers, namely SP1 and SP2 (Figure 1), each influenced by a subjectivity detection classifier, SD1 and SD2, respectively.

A correlation analysis between the confidence scores of SP1 and SP2 (C_{SP1} and C_{SP2} respectively) revealed that the two systems make different mistakes, which motivated combining them in an ensemble. Given a message and the confidence scores of the two systems (i.e., C_{SP1} and C_{SP2}), the ensemble computes a new confidence score for every sentiment label (C_{pos}, C_{neg} and C_{neu}) as follows:

$$C_{pos} = C_{SP1@pos} \cdot w_{pos} + C_{SP2@pos} \cdot (1 - w_{pos})$$

$$C_{neg} = C_{SP1@neg} \cdot w_{neg} + C_{SP2@neg} \cdot (1 - w_{neg})$$

$$C_{neu} = C_{SP1@neu} \cdot w_{neu} + C_{SP2@neu} \cdot (1 - w_{neu})$$

where w_{pos}, w_{neg}, w_{neu} are weights tuned on the development data. The sentiment with the highest confidence score is assigned to each tweet.[1]

Below, we describe the two Sentiment Polarity classifiers, along with the two subjectivity detection classifiers that influence them.

3.1 SP1 and SD1

First, each message is preprocessed by a a Twitter specific tokeniser and part-of-speech (POS) tagger (Owoputi et al., 2013) to obtain the tokens and the corresponding POS tags, which are necessary for some features.[2] Then, we extract features, which can be categorized as follows:[3]

- features based on morphology,
- POS based features,
- sentiment lexicon based features,
- negation based features,
- features based on clusters of tweets.

We used a linear SVM classifier (Vapnik, 1998; Cristianini and Shawe-Taylor, 2000; Joachims, 2002) trained on three labels, namely, positive, negative and neutral.[4]

As already mentioned, SP1 is influenced by a subjectivity detection classifier called SD1. That is, SP1 uses as a feature the confidence score of SD1. SD1 is also a linear SVM classifier, which is trained on data of two labels, neutral and subjective (i.e., positive or negative).[5] The higher the confidence score of SD1 the more likely it is for the message to express sentiment (positive or negative). Apart from the score of SD1 (which was used by SP1), SP1 and SD1 used the same features.

3.2 SP2 and SD2

The second system of our ensemble uses word embeddings (Mikolov et al., 2013; Pennington et al., 2014). We use the centroid of the word embeddings of each tweet as the feature vector of the tweet. The centroid of a tweet (message) M is computed as follows:

$$\vec{M} = \frac{1}{|M|} \sum_{i=1}^{|M|} \vec{w_i}$$

[1] Tuning led to w_{pos} = w_{neg} = w_{neu} = 0.66.

[2] No lemmatization or stemming was used and tokens could be words, emoticons, hashtags, etc.

[3] All the features of SP1 are described in detail in a publicly available report, accompanying the source code of the system. The code and the report are available at `https://github.com/nlpaueb/aueb.twitter.sentiment`.

[4] We used the SVM implementation of Scikit Learn (Pedregosa et al., 2011; Fan et al., 2008). The same implementation was used for all our SVM classifiers. The optimal C value was found to be 0.00341, by using 5-fold cross validation on $\text{TW}_{train16}$.

[5] The optimal C value for SD1 was found to be 0.00195, by using 5-fold cross validation on $\text{TW}_{train16}$.

Test set	Score	Ranking
$\text{TW}_{devtest13}$	66.61%	9/34
SMS_{13}	61.77%	6/34
$\text{TW}_{devtest14}$	70.81%	7/34
$\text{TW}_{sarcasm14}$	41.00%	18/34
LJ_{14}	69.51%	8/34
TW_{15}	62.34%	7/34
TW_{16}	60.52%	5/34

Table 1: Rankings of our system

	Train data	Dev data	Tweet2016
Strict 2 stages	62.60%	58.50%	54.83% (19/34)
SP1 (with SD1)	68.00%	64.70%	59.40% (7/34)
SP2 (with SD2)	60.80%	59.00%	57.50% (15/34)
ENS	68.40%	65.80%	60.52% (5/34)

Table 2: Average $F1$ scores of SP1, SP2, ENS (our ensemble) and a strict two-stage system.

where $|M|$ is the number of tokens in M and $\vec{w_i}$ is the embedding of word w_i.[6] We used the 200-dimensional word vectors for Twitter produced by GloVe (Pennington et al., 2014).[7]

As with SP1, SP2 incorporates the confidence score of SD2 as a feature. SD2 is a classifier trained on neutral and subjective data (positive or negative), again with centroid feature vectors. Given a message M, the confidence score of SD2 for M was added as a feature to its centroid and the resulting 201-dimension feature vector was used as input to SP2.[8] SP2 was then trained on the same three classes as SP1 (positive, negative, neutral).[9]

4 Experiments & Discussion

Our system was ranked 5th among 34 teams.[10] All teams were ranked by their score on the Twitter2016 Task 4 test dataset. Table 1 shows our rankings on each dataset. Below we discuss the results of our ensemble and we show how the subjectivity detection classifiers affect our system.

A strict two-stage approach, like the one suggested by Karampatsis et al. (2014), discards messages the sentiment detection (SD) classifier (first stage) decides they do not express sentiment, and classifies the rest as positive or negative. However, errors of the first stage propagate to the second, thus, playing a significant role in overall performance. We extend their approach and attempt to use the results of a subjectivity detection stage in a less rigorous manner; i.e., as a confidence factor along with various other features. Recall that our SD1 is actually the first stage of the system of Karampatsis et al. (2014), and that we use the confidence of SD1 as feature of SP1. Table 2 shows that SP1 (with the confidence of SP1 as a feature) outperforms the strict two-stage approach by 4.57%, yielding an increase in the ranking by 12 positions. Another interesting observation is that SP2 (with the confidence of SD2 as a feature) achieves a score only 1.9% lower than SP1 (with SD1) yielding a ranking around the middle of the list. This is achieved by using only features based on word embeddings along with the confidence of SD2 and no sophisticated feature engineering at all. A final, and also very interesting observation is that when we use an ensemble of SP1 and SP2, the results improve yielding a 5th place in the ranking.

5 Conclusions and future work

In this paper we presented the system with which we participated in the Message Polarity Classification subtask of SemEval-2016 Task 4. We used a weighted ensemble of two systems each operating in two stages. In a first, subjectivity detection stage, each message is assigned a confidence score representing the probability that the message expresses an opinion. This probability is then used as a feature by a classifier that detects sentiment. We used two different systems, one based on previous work by Karampatsis et al. (2014) (SP1 with the confidences of SD1 as a feature) and a second system that represents the messages by the centroids of their word embeddings (SP2 with the confidence of SD2 as a feature). The two systems are then combined with a weighted linear ensemble scheme in order to get the final sentiment label. Our experiments show that using the confidence of the subjectivity detection stage as a feature instead of using a strict two-stage ap-

[6] We allow multiple word occurrences in a sentence, while we ignore words without embeddings.

[7] The word vectors were pre-trained on a 2 billion tweets corpus. See `http://nlp.stanford.edu/projects/glove/`.

[8] The confidence scores of SD1 and SD2 were exponentially normalized (Bishop, 2006).

[9] The optimal C values were found to be 1.40688 for SD2 and 7.39618 for SP2, by using 5-fold cross validation.

[10] `http://alt.qcri.org/semeval2016/task4/data/uploads/semeval2016_task4_results.pdf`

proach can lead to an improved performance. Also, the ensemble performs better than any of its two systems on their own.

Despite the encouraging results of our approach (5th among 34 participating teams), there is still much room for improvement. A better continuous space vector representation of the messages might improve SD2 and SP2. Much research has been conducted recently on obtaining better continuous space vector representations of sentences (Le and Mikolov, 2014; Kiros et al., 2015; Hill et al., 2016) instead of centroid vectors. Another direction for future work would be to investigate replacing the SVM classifiers by multilayer perceptrons, possibly on top of recurrent neural nets that would compute vector representations of sentences.

Acknowledgments

This work was carried out during the BSc projects of the first two authors, which were co-supervised by the other three authors.

References

C. M. Bishop. 2006. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc.

N. Cristianini and J. Shawe-Taylor. 2000. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press.

R. E. Fan, K. W. Chang, C. J. Hsieh, X.-R. Wang, and C. J. Lin. 2008. Liblinear: A library for large linear classification. *The Journal of Machine Learning Research*, 9:1871–1874.

F. Hill, K. Cho, and A. Korhonen. 2016. Learning distributed representations of sentences from unlabelled data. *arXiv preprint arXiv:1602.03483*.

T. Joachims. 2002. *Learning to Classify Text Using Support Vector Machines: Methods, Theory, Algorithms*. Kluwer.

R. M. Karampatsis, J. Pavlopoulos, and P. Malakasiotis. 2014. AUEB: Two stage sentiment analysis of social network messages. In *Proceedings of the 8th International Workshop on Semantic Evaluation*, pages 114–118, Dublin, Ireland.

R. Kiros, Y. Zhu, R. Salakhutdinov, R. S. Zemel, A. Torralba, R. Urtasun, and S. Fidler. 2015. Skip-thought vectors. *arXiv preprint arXiv:1506.06726*.

Q. Le and T. Mikolov. 2014. Distributed representations of words and phrases. In *Proceedings of the 31th International Conference on Machine Learning*, pages 1188–1196, Beijing, China.

T. Mikolov, K. Chen, G. Corrado, and J. Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

P. Nakov, A. Ritter, S. Rosenthal, V. Stoyanov, and F. Sebastiani. 2016. SemEval-2016 task 4: Sentiment analysis in Twitter. In *Proceedings of the 10th International Workshop on Semantic Evaluation*, San Diego, California.

O. Owoputi, B. O'Connor, C. Dyer, K. Gimpel, N. Schneider, and N. A. Smith. 2013. Improved part-of-speech tagging for online conversational text with word clusters. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 380–390, Atlanta, Georgia.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

J. Pennington, R. Socher, and C. D. Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1532–1543, Doha, Qatar.

V. Vapnik. 1998. *Statistical learning theory*. John Wiley.