Polarity Classification for Target Phrases in Tweets: A Word2Vec Approach

Andi Rexha[1(✉)], Mark Kröll[1], Mauro Dragoni[2], and Roman Kern[1]

[1] Know-Center GmbH, Graz, Austria
{arexha,mkroell,rkern}@know-center.at
[2] FBK-IRST, Trento, Italy
dragoni@fbk.eu

Abstract. Twitter is one of the most popular micro-blogging services on the web. The service allows sharing, interaction and collaboration via short, informal and often unstructured messages called tweets. Polarity classification of tweets refers to the task of assigning a positive or a negative sentiment to an entire tweet. Quite similar is predicting the polarity of a specific target phrase, for instance *@Microsoft* or *#Linux*, which is contained in the tweet.

In this paper we present a Word2Vec approach to automatically predict the polarity of a target phrase in a tweet. In our classification setting, we thus do not have any polarity information but use only semantic information provided by a Word2Vec model trained on Twitter messages. To evaluate our feature representation approach, we apply well-established classification algorithms such as the Support Vector Machine and Naive Bayes. For the evaluation we used the *Semeval 2016 Task #4* dataset. Our approach achieves F1-measures of up to ∼90 % for the positive class and ∼54 % for the negative class without using polarity information about single words.

1 Introduction

With the growing popularity of online social media services, different types and means of communication are available nowadays. There is an observable trend towards microblogging and shorter text messages (snippets) which often are unstructured and informal. One of the most popular microblogging platforms is Twitter which allows for spreading short text messages (140 characters) called tweets. The language used in these messages often is very informal, with creative spelling and punctuation, misspellings, slang, URLs and abbreviations. In short, a challenge as well as opportunity for every researcher in the NLP area.

Automatically predicting the polarity of tweets represents an ongoing endeavor and relates to the task of assigning a positive or a negative sentiment to an entire tweet. Quite similar is predicting the polarity of a specific target phrase which is contained in the tweet. Consider following example where the references to *@Microsoft* and to *#Linux*) are called target phrases:

New features ***@Microsoft*** *suck. Check them back!* ***#Linux*** *solutions are awesome*

H. Sack et al. (Eds.): ESWC 2016 Satellite Events, LNCS 9989, pp. 217–223, 2016.
DOI: 10.1007/978-3-319-47602-5_40

The overall polarity of this tweet might turn out neutral, since the first part "New features **@Microsoft** suck" expresses a negative sentiment while the last part of the tweet "**#Linux** solutions are awesome." expresses a positive one. So, the averaging of sentiment assignments might lead to loss of information, i.e. the individual attitude towards products or the like.

In this paper we present an algorithm which automatically predicts the polarity of target phrases in a tweet. In the previous example, our algorithm will return a positive rating about the target *@Microsoft* and a negative one about the target *#Linux*. To do that, we explore using but semantic information (cf. [8]) given by a Word2Vec[1] model trained on Twitter messages, i.e. without using polarity information about single words. Word2Vec models provide a feature space representation of words which reflects their relation to other words in the training corpus. To evaluate our algorithm, we use the test and golden standard dataset of the *Semeval 2016 Task #4*[2] challenge about Twitter sentiment mining.

The paper is structured as follows. In Sect. 2 we present and discuss related work. In Sect. 3 we describe two different feature representations of tweets using the Word2Vec model. Section 4 evaluates our algorithm on the Semeval dataset. The paper concludes and presents future work in Sect. 5.

2 Related Work

The task of Sentiment Analysis, also known as opinion mining (cf. [7,9]), is to classify textual content according to expressed emotions and opinions. Sentiment classification has been a challenging topic in Natural Language Processing (cf. [14]). It is commonly defined as a binary classification task to assign a sentence either positive or negative polarity (cf. [10]). Turneys work was among the first ones to tackle automatic sentiment classification [13]. He employed an information-theoretic measure, i.e. mutual information, between a text phrase and the words excellent and poor as a decision metric.

Micro-blogging data such as tweets differs from regular text as it is extremely noisy, informal and does not allow for long messages (which might not be a disadvantage (cf. [3]). As a consequence, analyzing sentiment in Twitter data poses a lot of opportunities. Traditional feature representations such as part-of-speech information or the usage of lexicon features such as SentiWordNet have to be re-evaluated in the light of Twitter data. In case of part-of-speech information, Gimpel et al. [4] annotated tweets and developed a tagset and features to train an adequate tagger. Kouloumpis et al. [6] investigated the usefulness of existing lexical resources and other features including part-of-speech information in the analysis task.

Go et al. [5], for instance, used emoticons as additional features, for example, ":)" and ":-)" for the positive class, ":(" and ":-(" for the negative class.

[1] Word2Vec models provide a representation of words in a feature space reflecting their relation to other words in the corpus.

[2] http://alt.qcri.org/semeval2016/task4/data/uploads/semeval2016_task4_report.pdf.

They then applied machine learning techniques such as support vector machines to classify the tweets into a positive and a negative class. Agarwal et al. [1] introduced POS-specific prior polarity features along with using a tree kernel for tweet classification. Barbosa and Feng [2] present a robust approach to Twitter sentiment analysis. The robustness is based on an abstract representation of tweets as well as the usage of noisy/biased labels from three websites to train their model.

Last but not least, recent years have seen a lot of participation in the annual SemEval tasks on Twitter Sentiment Analysis (cf. [11,12,15]). This event provides optimal conditions to implement novel ideas and is a good starting point to catch up on the latest trends in this area.

3 A Word2Vec Approach

In this section we describe our algorithm's feature engineering which encompasses three steps (1) pre-processing, (2) feature representation (two approaches), and (3) post-processing.

3.1 Pre-processing

As preprocessing step, we add additional information to the words, i.e. part-of-speech, by using the Tweet NLP library[3]. Furthermore, we extract the Word2Vec vector[4] representation for each word of the tweet by using a Twitter model trained on ∼400 million tweets.

3.2 Feature Representation

In this paper, we experimented with two approaches of representing a tweet using a Word2Vec trained model on Twitter messages. In the first approach, we do not consider the position of target phrases and use Word2Vec information for every word in a tweet (see Fig. 1).

In the second approach, we consider only the neighborhood of a target phrase in our polarity classification task (see Fig. 2) by using a window of size n.

The tweets are preprocessed as described in Sect. 3.1, target phrases of the tweets are located according to their annotation in the dataset and window of size "n" is determined. Per tweet, there is only one target phrase. In case that a target phrase occurs several times in a tweet, only the first occurrence is taken into account resulting in exactly one window per tweet. For each word in the window, the Word2Vec information is extraced and an average vector is formed.

[3] http://www.cs.cmu.edu/~ark/TweetNLP/.
[4] http://www.fredericgodin.com/software/.

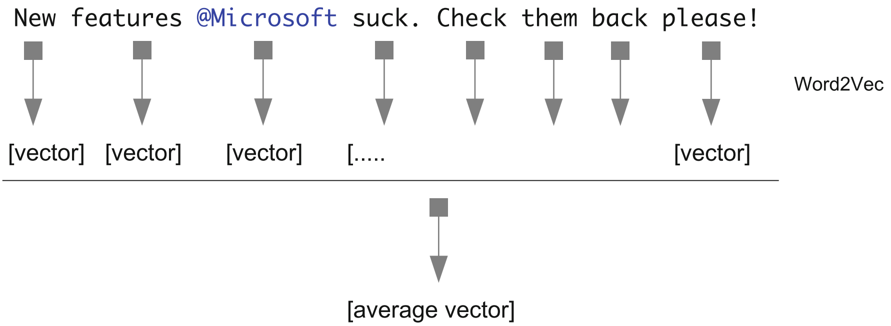

Fig. 1. In this approach, we extract Word2Vec information for all words in a tweet and form an average vector (post-processing step). We do not take into account the position of the target phrase

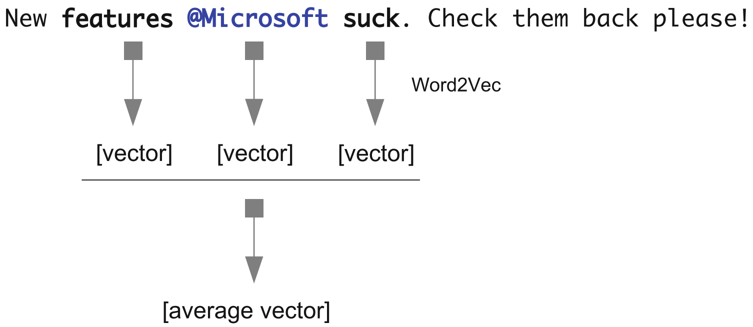

Fig. 2. In the second approach, we extract Word2Vec information for words in the neighborhood of a target phrase (covered by a window of size 1 in this case) and form an average vector (post-processing step)

3.3 Post-processing

In the postprocessing step we generate one average vector per tweet - either from every word (approach 1) or only from words within the window (approach 2).

As last step we introduce a binary feature which is set to 1 if in the tweet exists any negation word (don't, not, ...). Out of our experience, this feature provides valuable information to the subsequent learning step.

4 Results

We used the *Semeval 2016 Task 4* dataset to evaluate our two feature extraction approaches. The training set is composed by 3858 entries and the evaluation set by 10551 entries. Both datasets are skewed, i.e. the training set contains 17 % of negative and 83 % of positive and the evaluation set of 22 % of negative and 78 % of positive examples. In our experiments, we applied four different well-established classification models, i.e. Naive Bayes, Support Vector Machines, Logistic Regression and Random Trees, to our feature representations. For each feature representation version, we present precision, recall and F1-measures for the positive and negative classes: Tables 1 and 2 contain performance values of the positive and negative classes for the full text approach. Tables 3 and 4 contain performance values for the second approach with a window size n of 3. Other window sizes did not lead to better results. We remark that we experimented with small window sizes rather than large ones to focus on the proximity aspect.

Table 1. Precision, Recall and F1-Measure results for the negative class when using Word2Vec information of all words in a tweet to generate the average vector

	Precision	Recall	F1-Measure
Naive Bayes	0.396	**0.733**	0.514
Support vector machine	**0.724**	0.347	0.469
Logistic regression	0.606	0.481	**0.536**
Random tree	0.321	0.266	0.291

Table 2. Precision, Recall and F1-Measure results for the positive class when using Word2Vec information of all words in a tweet to generate the average vector

	Precision	Recall	F1-Measure
Naive Bayes	**0.900**	0.681	0.775
Support vector machine	0.838	**0.962**	**0.896**
Logistic regression	0.860	0.911	0.885
Random tree	0.801	0.840	0.820

Table 3. Precision, Recall and F1-Measure results for the negative class when using Word2Vec information of a target phrases' neighboring words (a window size of 3) in a tweet to generate the average vector

	Precision	Recall	F1-Measure
Naive Bayes	0.303	**0.732**	**0.429**
Support vector machine	**0.442**	0.273	0.338
Logistic regression	0.396	0.391	0.394
Random tree	0.260	0.281	0.271

Using the Word2Vec information for all words in a tweet yielded high F1-measures (up to ∼90 %) for the positve class and average F1-measures (up to ∼54 %) for the negative class. The performance difference might be due to the difference in available training examples for both classes. Table 1 reveals that the Support Vector Machine is capable of identifying instances of the negative class with a precision of ∼72 %.

Table 4. Precision, Recall and F1-Measure results for the positive class when using Word2Vec information of a target phrases' neighboring words (a window size of 3) in a tweet to generate the average vector

	Precision	Recall	F1-Measure
Naive bayes	**0.872**	0.520	0.652
Support vector machine	0.813	**0.902**	**0.855**
Logistic regression	0.827	0.830	0.829
Random tree	0.791	0.773	0.781

Tables 3 and 4 show worse performance values for both classes when compared to Tables 1 and 2, i.e. using the Word2Vec information of all the words in a tweet. However, in particular for the positive class, the proximity of a target phrase often contains sufficient semantic information for the prediction task as taking the entire tweet into account. Both approaches do quite well with the positive class. Both approaches yield considerably lower F1-measures for the negative class than for the positive one - probably due to the skewness in the dataset.

5 Conclusion

In this paper we introduced an algorithm which automatically predicted the polarity of a target phrase in a tweet. Our algorithm uses only semantic information provided by a Word2Vec model trained on Twitter messages. Evaluating our algorithm on the *Semeval 2016 Task #4* dataset shows that F1-measures of up to ∼90 % for the positive class and ∼54 % for the negative class are achievable without using polarity information about single words.

In future work we intend to exploit information provided by dependency trees of tweets for the polarity classification task. We hypothesize that going beyond textual proximity, i.e. taking into account remoter structures, might improve the performance of the classification algorithm.

Acknowledgments. This work is funded by the KIRAS program of the Austrian Research Promotion Agency (FFG) (project number 840824). The Know-Center is funded within the Austrian COMET Program under the auspices of the Austrian Ministry of Transport, Innovation and Technology, the Austrian Ministry of Economics and Labour and by the State of Styria. COMET is managed by the Austrian Research Promotion Agency FFG.

References

1. Agarwal, A., Biadsy, F., Mckeown, K.R.: Contextual phrase-level polarity analysis using lexical affect scoring and syntactic n-grams. In: Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2009 (2009)

2. Barbosa, L., Feng, J.: Robust sentiment detection on twitter from biased and noisy data. In: Proceedings of the 23rd International Conference on Computational Linguistics: Posters, COLING 2010 (2010)
3. Bermingham, A., Smeaton, A.F.: Classifying sentiment in microblogs: is brevity an advantage? In: Proceedings of the 19th ACM International Conference on Information and Knowledge Management, CIKM 2010. ACM (2010)
4. Gimpel, K., Schneider, N., O'Connor, B., Das, D., Mills, D., Eisenstein, J., Heilman, M., Yogatama, D., Flanigan, J., Smith, N.A.: Part-of-speech tagging for twitter: annotation, features, and experiments. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Short Papers - vol. 2, HLT 2011 (2011)
5. Go, A., Bhayani, R., Huang, L.: Twitter sentiment classification using distant supervision. Technicak report, Stanford (2009)
6. Kouloumpis, E., Wilson, T., Moore, J.D.: Twitter sentiment analysis: the good the bad and the omg! In: Proceedings of the Fifth International Conference on Weblogs and Social Media, Barcelona, Spain (2011)
7. Liu, B., Zhang, L.: Mining text data. In: Aggarwal, C.C., Zhai, C. (eds.) A Survey of Opinion Mining and Sentiment Analysis, pp. 415–463. Springer, New York (2012)
8. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)
9. Pang, B., Lee, L.: Opinion mining and sentiment analysis. Found. Trends Inf. Retrieval **2**(1–2), 1–135 (2008)
10. Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up? sentiment classification using machine learning techniques. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 79–86, EMNLP 2002. Association for Computational Linguistics, Stroudsburg (2002). http://dx.doi.org/10.3115/1118693.1118704
11. Rosenthal, S., Nakov, P., Kiritchenko, S., Mohammad, S., Ritter, A., Stoyanov, V.: Semeval-2015 task 10: sentiment analysis in twitter. In: Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015). Association for Computational Linguistics (2015)
12. Rosenthal, S., Ritter, A., Nakov, P., Stoyanov, V.: Semeval-2014 task 9: sentiment analysis in twitter. In: Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014). Association for Computational Linguistics and Dublin City University (2014)
13. Turney, P.D.: Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. In: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics. Association for Computational Linguistics (2002)
14. Wiebe, J., Wilson, T., Cardie, C.: Annotating expressions of opinions and emotions in language. Lang. Resour. Eval. **1**(2), 165–210 (2005)
15. Wilson, T., Kozareva, Z., Nakov, P., Alan, R., Rosenthal, S., Stoyonov, V.: Semeval-2013 task 2: sentiment analysis in twitter. In: Proceedings of the 7th International Workshop on Semantic Evaluation. Association for Computation Linguistics (2013)