TakeLab at SemEval-2017 Task 5: Linear Aggregation of Word Embeddings for Fine-Grained Sentiment Analysis on Financial News

Leon Rotim, Martin Tutek, Jan Šnajder
Text Analysis and Knowledge Engineering Lab
Faculty of Electrical Engineering and Computing, University of Zagreb
Unska 3, 10000 Zagreb, Croatia
{leon.rotim,martin.tutek,jan.snajder}@fer.hr

Abstract

This paper describes our system for fine-grained sentiment scoring of news headlines submitted to SemEval 2017 task 5, subtask 2. Our system uses a feature-light method that consists of a Support Vector Regression (SVR) with various kernels and word embedding vectors as features. Our best-performing submission scored 3rd on the task out of 29 teams and 4th out of 45 submissions, with a cosine score of 0.733.

1 Introduction

Sentiment analysis (Pang and Lee, 2008) is a task of predicting whether the text expresses a positive, negative, or neutral opinion in general or with respect to an entity of interest. Developing systems capable of performing highly accurate sentiment analysis has attracted considerable attention over the last two decades. The topic has been one of the main research areas in recent shared tasks, with main focus on social media texts, which are of particular interest for social studies (O'Connor et al., 2010; Wang et al., 2012) and marketing analysis (He et al., 2013; Yu et al., 2013). At the same time, social media texts pose a big challenge for sentiment analysis due to their short, informal and often ungrammatical format.

This work focuses on the second subtask of SemEval-2017 Task 5, which aims to perform fine-grained sentiment analysis of the financial news. Given that sentiments can affect market dynamics (Goonatilake and Herath, 2007; Van de Kauter et al., 2015), sentiment analysis of financial news can be a powerful tool for predicting market reactions. Similar to social media posts, finance news are short texts, but, unlike social media posts, the text is edited and hence grammatically correct. On the other hand, news headlines are notorious for the use of a specific language (Reah, 2002), which is often elliptical and compressed, and thus differs from the language used in the rest of the news story.

Many approaches to sentiment analysis resort to rich, domain-specific, hand-crafted features (Wilson et al., 2009; Abbasi et al., 2008). At the same time, there has been a growing interest in feature-light methods, including kernel-methods (Culotta and Sorensen, 2004; Lodhi et al., 2002a; Srivastava et al., 2013) and neural embeddings (Maas et al., 2011; Socher et al., 2013). These methods alleviate the need for manual creation of domain-specific features, while maintaining high accuracy. Most of the recently published work focuses on sentiment analysis problems that are framed as a classification task, while fine-grained analysis is framed as a regression problem. However, most of the high performing classification methods can be easily tuned to perform regression.

In this work we focus on feature-light methods as they do not require complex, time consuming feature engineering. More specifically, we focus on string kernels (Lodhi et al., 2002b) and methods using neural word embeddings (Mikolov et al., 2013a). Developing domain-specific, rich feature sets would probably make the method highly dependent to the specific problem and would be hardly applicable to similar problems in other domains. Feature-light methods have no such constrains: they typically offer satisfactory performance across different domains and may therefore be preferred to other domain-specific methods which use hand-crafted features.

2 Related Work

There has been considerable research focusing on sentiment analysis of short texts (Thelwall et al., 2010; Kiritchenko et al., 2014), especially within recent SemEval campaigns (Nakov et al., 2016;

Proceedings of the 11th International Workshop on Semantic Evaluations (SemEval-2017), pages 866–871,
Vancouver, Canada, August 3 - 4, 2017.

Rosenthal et al., 2015, 2014). A large body of recent work focuses on sentence-level sentiment prediction. Socher et al. (2012) and Socher et al. (2013) reported impressive results working with matrix-vector recursive neural network (MV-RNN) and recursive neural tensor networks models over sentence parse trees. Working with sentence parse trees Kim et al. (2015) and Srivastava et al. (2013) obtained competitive results using tree kernels as an alternative to recursive neural networks. These methods, while producing promising results, are highly dependent on parse trees. In practice, we often work with informal texts, where syntactic parsing often produces inaccurate results, which in turn heavily affects performances of the aforementioned methods. Furthermore, as noted by Le and Mikolov (2014), it is not straightforward how to extend these methods when working with text spans that range over multiple sentences.

There has been a growing amount of interest in methods that are not based on syntax. The most promising results have been achieved using neural word embeddings (Mikolov et al., 2013a), while string kernels (Zhang et al., 2008; Lodhi et al., 2002a; Leslie et al., 2002) offer a viable alternative. Maas et al. (2011) and Tang et al. (2014) reported promising results by learning sentiment specific word embeddings. By extending word embeddings to more complex paragraph embeddings Le and Mikolov (2014) reported state-of-the-art results on sentiment classification for both short and long English texts. Building on word embeddings, Joulin et al. (2016) developed an end-to-end, domain independent, high-performance text classification model.

3 Dataset

Our task was, given a news headline, to predict the sentiment score for a specific company mentioned in the headline. The dataset consisted of the name of the company, the text of the news headline and a value denoting the sentiment.

The sentiment was on a scale between -1 and 1 (inclusive), where -1 corresponds to very negative sentiment, 0 is considered neutral, while 1 stands for a very positive sentiment. The news headlines were on average 10 words in length and largely composed of abbreviations.

The training set was composed of 1142 news headlines, while the test set contained 491 headlines, i.e., a 70:30 train-test split. The training set and the test set mention 294 and 168 unique companies, respectively. The distribution of headlines for a specific company was not uniform, and only 58 companies in the train set were targets of more than 4 news headlines, while "Barclays" – the most frequently mentioned one – was the target 67 times. In total, 112 companies occur in both the train and test set.

id	5
company	Ryanair
title	EasyJet attracts more passengers in June but still lags Ryanair
sentiment	0.259

Table 1: Sample training data instance

An example of a training data instance is given in Table 1. This particular example also illustrates a possible difficulty regarding the headlines as they might refer to more than one company. Such examples, however, are pretty rare in the dataset.

As for the class breakdown in the training set, we observe that the number of positively labeled instances is significantly larger than the number of negatively labeled instances (a ratio of $653 : 451$ in favor of headlines with positive sentiment, including 38 headlines with a perfectly neutral score of 0.0). However, the distribution of the target variable has an almost zero mean value of 0.031 and a standard deviation of 0.39. All things considered, we conclude that the dataset was fairly well-balanced and the dependent variable was not skewed towards either class.

4 Methods

While working on fine-grained sentiment analysis, we focus on feature-light, domain independent methods. In all considered methods, we use support vector regression (SVR) model for sentiment prediction. The SVR allows us to experiment with both different features and kernels. Model training is performed using LIBSVM (Chang and Lin, 2011) for the non-linear kernel and LIBLINEAR (Fan et al., 2008) for the linear kernel.

BoW baseline. We use the standard bag-of-words (BoW) methods as a sensible baseline. BoW methods are implemented by creating a dictionary of words appearing in the train set. We implemented the BoW baseline using all uni-, bi-, and trigrams that occur at least twice in the dataset, while

filtering out words from the standard stopword list. We experiment with TF-IDF and Bernoulli weighting schemes for the word features. For generating the n-grams, we used NLTK toolkit (Bird et al., 2009), and filtered out n-grams consisting of stopwords.

String kernels. String kernels offer a dictionary-free alternative compared to other commonly-used methods. There are several known string kernels in use, the most popular being the spectrum kernel (SK) (Leslie et al., 2002) and the subsequence kernel (SSK) (Lodhi et al., 2002a). The SSK measures string similarity by first mapping each input string s to:

$$\varphi_u(s) = \sum_{\boldsymbol{i}:u=s[\boldsymbol{i}]} \lambda^{l(\boldsymbol{i})} \quad (1)$$

where u is a subsequence searched for in s, $\boldsymbol{i}$ is a vector of indices at which u appears in s, l is a function measuring the length of a matched subsequence and $\lambda \leq 1$ is a weighting parameter giving lower weights to longer subsequences. Using (1), the SSK kernel is defined as:

$$K_n(s,t) = \sum_{u \in \Sigma^n} \langle \varphi_u(s), \varphi_u(t) \rangle$$

where n is maximum subsequence length for which we calculate the kernel and Σ^n is a set of all finite strings of length n. Spectrum kernel can be defined as a special case of SSK where $\lambda = 1$ and $\boldsymbol{i}$ must yield continuous sequences. We experiment with both SK and SSK kernels, which we computed using the string similarity tool Harry.[1]

Word embeddings. Word embedding are task independent features, yet they offer competitive results on many text classification tasks. We experimented with pretrained word embeddings, namely GloVe (Pennington et al., 2014) and Skip-gram (Mikolov et al., 2013b) trained on the Google News corpus.[2] We achieved the best results with the 300-dimensional Google News vectors.

The feature vector that is fed to the classifier is computed as the linear aggregate of the words making up the headline, simply as the average of the word embeddings of the individual words. Lowercasing the words that appear in the title gave us a considerable performance gain, which is expected since most of the words appearing in news headlines are title-cased. We refer to this method as the *word embeddings method* (WEM).

We further experimented with additional filtering of the word tokens we use for building word embedding vectors. Our motivation was based on the observation that sentiment-bearing words typically exclude the named entities. We therefore used StanfordNLP (Manning et al., 2014) named entity recognition (NER) tools to filter out all named entities before building adding up the word embedding vectors. We refer to this method as the *filtered word embeddings method* (FWEM).

When using word embeddings as features, we experimented with the linear, RBF, and cosine kernel (CK). The latter is defined as:

$$CK(\boldsymbol{x},\boldsymbol{y}) = \left[\frac{1}{2}\left(1 + \frac{\langle \boldsymbol{x},\boldsymbol{y}\rangle}{\|\boldsymbol{x}\|\|\boldsymbol{y}\|}\right)\right]^{\alpha}$$

5 Results

Model evaluation was performed as defined on the task description page.[3] From the instances given in the test set, we create a vector containing ground truth annotations G and a vector containing our model predictions P. Model performance score is computed using cosine similarity between the two vectors, as follows:

$$cosine(G,P) = \frac{\sum_{i=1}^{n} G_i \cdot P_i}{\sqrt{\sum_{i=1}^{n} G_i^2}\sqrt{\sum_{i=1}^{n} P_i^2}} \quad (2)$$

To optimize the hyperparameters of the models (C for linear SVR, n and λ SSK, n for SK, α and C for cosine kernel, and C and γ for RBF kernel), we performed a grid search in a nested K-folded cross-validation on the train set, using 10 folds in the outer and 5 folds in the inner loop. To select the best parameters for a model, we choose the ones that consistently provided the best result across the 10 outer loops. Using the chosen hyperparameters, we finally train that model on the complete train set. The best results for all of the considered models are reported in Table 2.

While working with BoW models, the best results were obtained using the simple Bernoulli feature weighting scheme, indicating whether a term appeared in the headline with a weight of 1 and 0 otherwise.

[1] http://www.mlsec.org/harry/index.html

[2] https://code.google.com/archive/p/word2vec/

[3] http://alt.qcri.org/semeval2017/task5/index.php?id=evaluation

Method	Cosine score
$\text{BoW}_{Bernoulli}$	0.539
SSK	0.654
SK	0.671
WEM_{linear}	0.610
WEM_{RBF}	0.724
WEM_{Cosine}	0.730
FWEM_{linear}	0.612
FWEM_{RBF}	0.727
FWEM^{*}_{Cosine}	**0.733**

Table 2: Cosine similarity between ground truth annotations and model predictions (higher is better). Subscript displayed with (F)WEM methods indicate the kernel used to train the model. Model marked with (*) is the submitted model.

String kernels gave us a considerable performance gains in comparison to the BoW baseline. Interestingly, experiments showed that the SK kernel outperformed the SSK kernel.

Using word embeddings provided us with significant performance gains compared to the other two methods. Word embedding features combined with the linear kernel did not outperform string kernels. However, using non-linear kernel such as RBF and especially cosine kernel yielded substantial performance gains.

6 Conclusion

We described our system for fine-grained sentiment scoring of news headlines, which we submitted to the SemEval 2017 task 5, subtask 2. We implemented a number of feature-light methods for sentiment analysis with basic preprocessing. Our best performing method used skip-gram word embeddings trained on the Google News corpus, which were fed as features to a cosine kernel Support Vector Regression. We report our results on the gold set, where our system ranked 3rd place out of 29 teams, with a cosine score of 0.733.

It should be note that we did not use the information about which company the sentiment is measured for in any way. Arguably, not using this information leads to performance decreases when dealing with (1) headlines entirely unrelated to the company of interest and (2) headlines containing mentions of multiple companies. For future work, it would be interesting to consider encoding this information into the model or using additional preprocessing methods to detect specific parts of the headline related to the company of interest.

References

Ahmed Abbasi, Hsinchun Chen, and Arab Salem. 2008. Sentiment analysis in multiple languages: Feature selection for opinion classification in web forums. *ACM Transactions on Information Systems (TOIS)* 26(3):12.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.".

Chih-Chung Chang and Chih-Jen Lin. 2011. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology* 2:27:1–27:27.

Aron Culotta and Jeffrey Sorensen. 2004. Dependency tree kernels for relation extraction. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, page 423.

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. LIBLINEAR: A library for large linear classification. *Journal of machine learning research* 9:1871–1874.

Rohitha Goonatilake and Susantha Herath. 2007. The volatility of the stock market and news. *International Research Journal of Finance and Economics* 3(11):53–65.

Wu He, Shenghua Zha, and Ling Li. 2013. Social media competitive analysis and text mining: A case study in the pizza industry. *International Journal of Information Management* 33(3):464–472.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759* .

Jonghoon Kim, Francois Rousseau, and Michalis Vazirgiannis. 2015. Convolutional sentence kernel from word embeddings for short text categorization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Lisbon, Portugal, pages 775–780.

Svetlana Kiritchenko, Xiaodan Zhu, and Saif M. Mohammad. 2014. Sentiment analysis of short informal texts. *Journal of Artificial Intelligence Research* 50:723–762.

Quoc V. Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *Proceedings of The 31st International Conference on Machine Learning (ICML)*. volume 14, pages 1188–1196.

Christina S Leslie, Eleazar Eskin, and William Stafford Noble. 2002. The spectrum kernel: A string kernel for svm protein classification. In *Proceedings of the Pacific Symposium on Biocomputing*. volume 7, pages 566–575.

Huma Lodhi, Craig Saunders, John Shawe-Taylor, Nello Cristianini, and Chris Watkins. 2002a. Text classification using string kernels. *Journal of Machine Learning Research* 2(Feb):419–444.

Huma Lodhi, Craig Saunders, John Shawe-Taylor, Nello Cristianini, and Chris Watkins. 2002b. Text classification using string kernels. *Journal of Machine Learning Research* 2(Feb):419–444.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*. Association for Computational Linguistics, Stroudsburg, PA, USA, pages 142–150.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*. pages 55–60. http://www.aclweb.org/anthology/P/P14/P14-5010.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* .

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Proceedings of the Neural Information Processing Systems Conference (NIPS 2013)*. Lake Tahoe, USA, pages 3111–3119.

Preslav Nakov, Alan Ritter, Sara Rosenthal, Fabrizio Sebastiani, and Veselin Stoyanov. 2016. Semeval-2016 Task 4: Sentiment analysis in Twitter. *Proceedings of SemEval* pages 1–18.

Brendan O'Connor, Ramnath Balasubramanyan, Bryan R Routledge, and Noah A. Smith. 2010. From tweets to polls: Linking text sentiment to public opinion time series. In *International Conference on Web and Social Media (ICWSM)*. Washington, DC, pages 122–129.

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

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*. pages 1532–1543. http://www.aclweb.org/anthology/D14-1162.

Danuta Reah. 2002. *The language of newspapers*. Psychology Press.

Sara Rosenthal, Preslav Nakov, Svetlana Kiritchenko, Saif M 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.

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)*. Dublin, Ireland, pages 73–80.

Richard Socher, Brody Huval, Christopher D. Manning, and Andrew Y. Ng. 2012. Semantic compositionality through recursive matrix-vector spaces. In *Proceedings of the conference on empirical methods in natural language processing (EMNLP)*. Association for Computational Linguistics, pages 1201–1211.

Richard Socher, Alex Perelygin, Jean Y Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the conference on empirical methods in natural language processing (EMNLP)*. volume 1631, page 1642.

Shashank Srivastava, Dirk Hovy, and Eduard H Hovy. 2013. A walk-based semantically enriched tree kernel over distributed word representations. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Seattle, USA, pages 1411–1416.

Duyu Tang, Furu Wei, Nan Yang, Ming Zhou, Ting Liu, and Bing Qin. 2014. Learning sentiment-specific word embedding for twitter sentiment classification. In *The 52nd Annual Meeting of the Association for Computational Linguistics (ACL)*. Baltimore, MD, USA, pages 1555–1565.

Mike Thelwall, Kevan Buckley, Georgios Paltoglou, Di Cai, and Arvid Kappas. 2010. Sentiment strength detection in short informal text. *Journal of the American Society for Information Science and Technology* 61(12):2544–2558.

Marjan Van de Kauter, Diane Breesch, and Véronique Hoste. 2015. Fine-grained analysis of explicit and implicit sentiment in financial news articles. *Expert Systems with applications* 42(11):4999–5010.

Hao Wang, Dogan Can, Abe Kazemzadeh, François Bar, and Shrikanth Narayanan. 2012. A system for real-time twitter sentiment analysis of 2012 us presidential election cycle. In *Proceedings of the ACL 2012 System Demonstrations*. Association for Computational Linguistics, pages 115–120.

Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. 2009. Recognizing contextual polarity: An exploration of features for phrase-level sentiment analysis. *Computational linguistics* 35(3):399–433.

Yang Yu, Wenjing Duan, and Qing Cao. 2013. The impact of social and conventional media on firm equity value: A sentiment analysis approach. *Decision Support Systems* 55(4):919–926.

Changli Zhang, Wanli Zuo, Tao Peng, and Fengling He. 2008. Sentiment classification for Chinese reviews using machine learning methods based on string kernel. In *Proceedings of the 3rd International Conference on Convergence Information (ICCIT)*. IEEE, Busan, Korea, volume 2, pages 909–914.