funSentiment at SemEval-2017 Task 5: Fine-Grained Sentiment Analysis on Financial Microblogs Using Word Vectors Built from StockTwits and Twitter

Quanzhi Li, Sameena Shah, Armineh Nourbakhsh, Rui Fang, Xiaomo Liu

Research and Development
Thomson Reuters
3 Times Square, NYC, NY 10036
{quanzhi.li, sameena.shah, armineh.nourbakhsh, rui.fang, xiaomo.liu}@thomsonreuters.com

Abstract

This paper describes the approach we used for SemEval-2017 Task 5: Fine-Grained Sentiment Analysis on Financial Microblogs. We use three types of word embeddings in our algorithm: word embeddings learned from 200 million tweets, sentiment-specific word embeddings learned from 10 million tweets using distance supervision, and word embeddings learned from 20 million StockTwits messages. In our approach, we also take the left and right context of the target company into consideration when generating polarity prediction features. All the features generated from different word embeddings and contexts are integrated together to train our algorithm.

1 Introduction

Domain specific Sentiment Analysis has received much attention recently. The financial domain is a high-impact use case for Sentiment Analysis because it has been shown that sentiments and opinions can affect market dynamics [9, 48]. Given the link between sentiment and market dynamics, the analysis of public sentiment becomes a powerful method to predict the market reaction. One main source of public sentiment is social media, such as Twitter and StockTiwts.

In this paper, we describe our approach for SemEval-2017 Task 5: Fine-Grained Sentiment Analysis on Financial Microblogs (Cortis et al., 2017). The task is: given a microblog message, predict the sentiment score for each of the companies/stocks mentioned. Sentiment values needed to be floating point values within the range of -1 (very negative/bearish) to 1 (very positive/ bullish), with 0 designating neutral sentiment. Our approach uses word embeddings (WE-Twitter) learned from general tweets, sentiment specific word embeddings (SSWE) learned from distance supervised tweets, and word embeddings learned from StockTwits messages (WE-StockTwits).

Message or sentence level sentiment classification has been studied by many previous works (Go et al., 2009; Mohammand et al., 2013; Pang et al., 2002; Liu, 2012; Tang et al., 2014), but there are few studies on target-dependent, or entity level, sentiment prediction (Jiang et al., 2011; Dong et al., 2014; Vo and Zhang, 2015). A target entity in a message does not necessarily have the same polarity type as the message, and different entities in the same message may have different polarities. For example, in the tweet "iPhone is better than Blackberry", the two named entities, *iPhone* and *Blackberry*, will have different sentiment polarities. Recent studies have focused on learning features directly from tweet text. One approach is to generate sentence representations from word embeddings. Several word embedding generation algorithms have been proposed in previous studies (Collobert et al., 2011; Mikolov et al., 2013). Using the general word embeddings directly in sentiment analysis is not effective, since they mainly model a word's semantic context, ignoring the sentiment clues in text. Therefore, words with opposite polarity, such as *worst* and *best*, are mapped onto vectors embeddings that are close to each other in some dimensions. Tang et al. (2014) propose a sentiment-specific word embedding (SSWE) method for sentiment analysis, by extending the word embedding algorithm. SSWE encodes sentiment information in the word embeddings.

Many terms in financial market have different meanings, especially sentiment polarity, from that in other domains or sources, such as general news articles and Twitter. For example, terms

Proceedings of the 11th International Workshop on Semantic Evaluations (SemEval-2017), pages 852–856,
Vancouver, Canada, August 3 - 4, 2017. ©2017 Association for Computational Linguistics

long, *short*, *put* and *call* have special meanings in stock market. Another example is the term *underestimate*, which is a negative term in general, but it can suggest an opportunity to buy when used in stock market messages. Therefore, in this study we also build word embedding specifically from StockTwits messages.

The context of an entity will affect its polarity value, and usually an entity has a left context and also a right one, unless it is at the beginning or end of a message. Both the context information and the interaction between these two contexts are included in the algorithm features of our approach. In this task, the financial microblogs are from StockTwits and Twitter, so in our approach, we incorporate features generated from WE-Twitter, SSWE and WE-StockTwits to represent these contexts, since they complement each other.

2 Previous Studies

Sentence or Message Level Sentiment: Traditional sentiment analysis approaches use sentiment lexicons (Mohammad et al., 2013; Thelwall et al., 2012; Turney, 2002) to generate various features. Pang et al. treat sentiment classification as a special case of text categorization, by applying learning algorithms (2002). Many studies follow Pang's approach by designing features and applying different learning algorithms on them (Feldman, 2013; Liu, 2012). Go et al. (2009) proposed a distance supervision approach to derive features from tweets obtained by positive and negative emotions. Some studies (Hu et al., 2013; Liu, 2012; Pak and Paroubek 2010) follow this approach. Feature engineering plays an important role in microblog sentiment analysis; Mohammad et al. (2013) implemented hundreds of handcrafted features for tweet sentiment analysis.

Deep learning has been used in the sentiment analysis tasks, mainly by applying word embeddings (Collobert et al., 2011; Mikolov et al., 2013). Learning the compositionality of phrase and sentence and then using them in sentiment classification is also explored by some studies (Hermann and Blunsom, 2013; Socher et al., 2011; Socher et al., 2013). Using the general word embeddings directly in sentiment analysis may not be effective, since they mainly model a word's semantic context, ignoring the sentiment clues in text. Tang et al. (2014) propose a sentiment-specific word embedding method by extending the word embedding algorithm from (Collobert et al., 2011) and incorporating sentiment data in the learning of word embeddings.

Entity or Target Level Sentiment: Jiang et al. (2011) use both entity dependent and independent features generated based on a set of rules to assign polarity to entities. By using POS features and the CRF algorithm, Mitchell et al. (2013) identify polarities for people and organizations in tweets. Dong et al. (2014) apply adaptive recursive neural network on the entity level sentiment classification. These two approaches use syntax parsers to parse the tweet to generate related features. In our approach, we consider both the left and right contexts of a target when generating features.

3 Methodology

In this section, we describe the three main components used in our method, the WE-Twitter, SSWE and WE-StockTwits models, and how the learning features are generated from them and integrated together.

3.1 WE-Twitter and WE-StockTwits Word Embedding

Word Embedding: A word embedding is a dense, low-dimensional and real-valued vector for a word. The embeddings of a word capture both the syntactic structure and semantics of the word. Traditional bag-of-words and bag-of-n-grams hardly capture the semantics of words. Word embeddings have been used in many NLP tasks. The C&W model (Collobert et al., 2011) and the word2vec model (Mikolov et al., 2013), which is used in this study to generate the WE-Twitter and WE-StockTwits embeddings, are the two popular models.

The embeddings are learned to optimize an objective function defined on the original text, such as likelihood for word occurrences. One implementation is the word2vec from Mikolov et al. (2013). This model has two training options, continuous bag of words and the Skip-gram model. The Skip-gram model is an efficient method for learning high-quality distributed vector representations that capture a large number of precise syntactic and semantic word relationships. This model is used in our method for building WE-Twitter and WE-StockTwits models. Generating word embeddings from text corpus is an unsupervised process. To get high quality embedding vectors, a large amount of training data is necessary. After training, each word, including all hashtags in the case of tweet, is represented by a low-dimensional, dense and real-valued vector.

WE-Twitter model construction: The tweets for building the WE-Twitter model include tweets obtained through Twitter's public streaming API and The Decahose data (10% of Twitter's streaming data) obtained from Twitter. Only English tweets are included in this study. In total there are about 200 million tweets. Each tweet text is preprocessed to get a clean version, following steps blow:

- all URLs and mentions are removed.
- dates are converted to a symbol.
- all ratios are replaced by a special symbol.
- integers and decimals are normalized to two special symbols.
- all special characters, except hashtags, cashtags, emoticons, question marks and exclamations, are removed.

Stop words are not removed, since they provide important information on how other words are used. In total, about 2.9 billion words were used to train the WE-Twitter model. Based on our pilot experiments, we set the embedding dimension size, word frequency threshold and window size as 300, 5 and 8, respectively. There are about 1.9 million unique words in this model.

WE-StockTwits model construction: StockTwits is a financial social network for sharing ideas among traders. Anyone on StockTwits can contribute content – short messages limited to 140 characters that cover ideas on specific investments. Most messages have a *cashtag*, which is a stock symbol, such as $aapl, to specify the entity (stock) this message is about. We received the permission from StockTwits to access their historical message archive from 2011 to 2016. We extract 20 million messages from this data set to build the WE-StockTwits model. Some preprocessing steps are performed to clean the messages:

- messages that contain only cashtags, URLs, or mentions are discarded, since they do not have meaningful terms.
- message text is converted to lower case.
- all URLs are removed.
- all mentions are converted to a special symbol, for privacy reason.
- all cashtags are replaced by a special symbol, to avoid cashtags to gain a polarity value related to a particular time period.

The embedding dimension size, word frequency threshold and window size are set as 300, 5 and 8, respectively.

3.2 Sentiment-Specific Word Embedding

SSWE: The C&W model (Collobert et al., 2011) learns word embeddings based on the syntactic contexts of words. It replaces the center word with a random word and derives a corrupted n-gram. The training objective is that the original n-gram is expected to obtain a higher language model score than the corrupted n-gram. The original and corrupted n-grams are treated as inputs of a feed-forward neural network, respectively.

SSWE extends the C&W model by incorporating the sentiment information into the neural network to learn the embeddings; it captures the sentiment information of sentences as well as the syntactic contexts of words (Tang et al., 2014). Given an original (or corrupted) n-gram and the sentiment polarity of a tweet as input, it predicts a two-dimensional vector (f_0, f_1), for each input n-gram, where (f_0, f_1) are the language model score and sentiment score of the input n-gram, respectively. The training objectives are twofold: the original n-gram should get a higher language model score than the corrupted n-gram, and the polarity score of the original n-gram should be more aligned to the polarity label of the tweet than the corrupted one. The loss function is the linear combination of two losses - $loss_0\ (t, t')$ is the syntactic loss and $loss_1\ (t, t')$ is the sentiment loss:

$$loss\ (t, t') = \alpha * loss_0\ (t, t') + (1\text{-}\alpha) * loss_1\ (t, t')$$

The SSWE model used in this study was trained from massive distant-supervised tweets, collected using positive and negative emotions.

SSWE model construction: The SSWE model for Twitter was trained from massive distant-supervised tweets, collected using positive and negative emoticons, such as *:), =), :(and :-(*. A total of 10 million tweets were collected, where 5 million contain positive emotions and the other 5 million contain negative ones. The embedding dimension size was set as 50 and the window size as 3.

3.3 Feature Generation

3.3.1 Features

Given a message and the target entity, nine types of features are generated based on WE-Twitter, SSWE, and WE-StockTwits models. They are integrated together to train the algorithm. Figure 1 shows the nine types of features. Three types of features are generated from SSWE embeddings for a target entity. The red ones are SSWE embeddings, and the blue ones are WE-Twitter and WE-StockTwits embeddings. The subscript letter $_L$ and $_R$ refer to the left and right side of an entity, respectively. These features are described below:

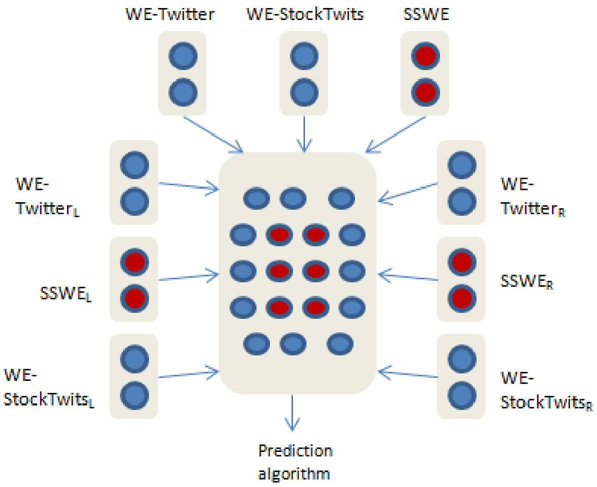

Figure 1. The features generated from different word embedding models and different contexts.

WE-Twitter$_L$ and WE-Twitter$_R$: These are the WE-Twitter embeddings for the text on the left side and right side of the target entity, respectively. In this task, occasionally, the given cashtag (company stock symbol) does not appear in the message text. In this case, the whole tweet text is used for both the left and right contexts, and this case is handled in the same way when generating WE-StockTwits$_L$, WE-StockTwits$_R$, SSWE$_L$ and SSWE$_R$ described below.

WE-StockTwits$_L$ and WE-StockTwits$_R$: These are the WE-StockTwits embeddings for the text on the left side and right side of the target entity, respectively.

SSWE$_L$ and SSWE$_R$: These are the SSWE embeddings for the text on the left side and right side of the target entity, respectively.

WE-Twitter, WE-StockTwits and SSWE: these are the embeddings generated from the whole message text, which means they are entity independent features. We use these three features to capture the whole message, which reflects the interaction between the left and right sides of the entity.

These nine types of embeddings together capture different types of information we are interested: the entity's left and right contexts, the interaction of the two sides, the sentiment specific word embedding information, and the general word embedding information learned from Twitter and StockTwits.

3.3.2 Text Representation from Term Embeddings

A message or text segment, such as the left/right context of an entity, has multiple words and each word has its own embedding vector. How to combine them together to represent this message so that all messages will have the same size of embedding vector needs to be explored. There are different ways to do this, e.g. for each embedding dimension, using the max value of all the words. In our approach, we use the concatenation convolution layer, which concatenates the layers of max, min and average of word embeddings, because this layer gives the best performance based on our pilot experiments. The concatenation layer is expressed as follow:

$$Z(t) = [Z_{max}(t), Z_{min}(t), Z_{ave}(t)]$$

where $Z(t)$ is the representation of text segment t.

4 Experiment and Result

For this task, the training data are provided by the task organizers. There are 1,704 tweets and StockTwits messages. We downloaded them from Twitter and StockTwits. To build our model, we split this data set into three parts: 80% as training data and 20% as development data. Since the predicted output in this task is a real value, so we use a liner regression algorithm in our approach. Based on the cosine similarity metric and the evaluation data set, which consists of 800 tweets and StockTwits messages (Cortis et al., 2017), the score of our approach is 0.7153, and our team is ranked at #6 among the 38 submissions.

5 Conclusion

This paper describes the approach we used for SemEval-2017 Task 5: Fine-Grained Sentiment Analysis on Financial Microblogs. We use three types of word embeddings in our algorithm: general word embeddings learned from 200 million tweets, sentiment-specific word embeddings learned from 10 million tweets using distance supervision, and word embeddings learned from 20 million StockTwits messages. We treat the task as a target-dependent sentiment analysis problem and consider the context of the target company.

References

Stefano Baccianella, Andrea Esuli , Fabrizio Sebastiani, SentiWordNet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining, *LREC 2010*

Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. 2009. Evaluation measures for ordinal regression. *In Proceedings of the 9th IEEE International Conference on Intelligent Systems Design and Applications (ISDA 2009)*. Pisa, IT, pages 283–287.

Ronan Collobert, Jason Weston, Leon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *Journal Machine Learning Research*, 12:2493–2537, 2011.

Keith Cortis, Andre Freitas, Tobias Daudert, Manuela H¨urlimann, Manel Zarrouk, and Brian Davis, SemEval-2017 Task 5: Fine-Grained Sentiment Analysis on Financial Microblogs and News, SemEval-2017

L. Dong, F. Wei, C. Tan, D. Tang, M. Zhou, K. Xu, adaptive recurisive neural network for target-depedant twitter sentiment classification. In *Proceedings of ACL 2014.*

R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. LIBLINEAR: A Library for Large Linear Classification, *Journal of Machine Learning Research*, 9(2008), 1871-1874.

Ronen Feldman. 2013. Techniques and applications for sentiment analysis. *Communications of the ACM*, 56(4):82–89.

Alec Go, Lei Huang, and Richa Bhayani. 2009. Twitter sentiment analysis. *Final Projects from CS224N for Spring 2008/2009* at The Stanford Natural Language Processing Group.

Karl M. Hermann and Phil Blunsom. 2013. The role of syntax in vector space models of compositional semantics. In *Proceedings ACL*, pages 894–904

Xia Hu, Jiliang Tang, Huiji Gao, and Huan Liu. 2013. Unsupervised sentiment analysis with emotional signals. In *Proceedings of the International World Wide Web Conference*, pages 607–618.

L. Jiang, M. Yu, M. Zhou, X. Liu, T. Zhao, target-depedant twitter sentiment classification, In *Proceedings of ACL* 2011

S.S. Keerthi, S.K. Shevade, C. Bhattacharyya, K.R.K. Murthy, 2001. Improvements to Platt's SMO Algorithm for SVM Classifier Design. *Neural Computation*. 13(3):637-649.

Bing Liu. 2012. Sentiment analysis and opinion mining. *Synthesis Lectures on Human Language Technologies*, 5(1):1–167

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed Representations of Words and Phrases and their Compositionality. In *Proceedings of NIPS*, 2013.

Christopher Manning, Prabhakar Raghavan and Hinrich Schütze, (2008), Vector space classification, *Introduction to Information Retrieval*. Cambridge University Press.

Margaret Mitchell, Jacqui Aguilar,Theresa Wilson, and Benjamin Van Durme. Open domain targeted sentiment. I *Proceedings of EMNLP*, 2013.

Saif M Mohammad, Svetlana Kiritchenko, and Xiaodan Zhu. 2013. Nrc-canada: Building the state-ofthe- art in sentiment analysis of tweets. In *Proceedings of SemEval* 2013.

P. Nakov, A. Ritter, S. Rosenthal, F. Sebastiani and V. Stoyanov. 2016. SemEval-2016 Tasks: Sentiment Analysis in Twitter. In *Proceedings of SemEval* 2016

Alexander Pak and Patrick Paroubek. 2010. Twitter as a corpus for sentiment analysis and opinion mining. In *Proceedings of Language Resources and Evaluation Conference*, volume 2010.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up? sentiment classification using machine learning techniques. In *Proceedings of EMNLP*, pages 79–86

J. Platt, Fast Training of Support Vector Machines using Sequential Minimal Optimization. In B. Schoelkopf and C. Burges and A. Smola, editors, *Advances in Kernel Methods - Support Vector Learning*, 1998

Yossi Rubner, Carlo Tomasi, and Leonidas J. Guibas. 2000. The Earth Mover’s Distance as a metric for image retrieval. *International Journal of Computer Vision*, 40(2):99–121.

Richard Socher, J. Pennington, E.H. Huang, A.Y. Ng, and C.D. Manning. 2011. Semi-supervised recursive autoencoders for predicting sentiment distributions. In *Proceedings of EMNLP* 2011, pages 151–161.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of EMNLP* 2013, pages 1631–1642.

Duyu Tang, Furu Wei, Nan Yang, Ming Zhou, Ting Liu, and Bing Qin. 2014. Learning sentimentspecific word embedding for twitter sentiment classification. In *Proceedings of ACL* 2014, pages 1555–1565.

Duyu Tang, Bing Qin, Xiaocheng Feng, Ting Liu, 2016. Effective LSTMs for Target-Dependent Sentiment Classification, In *Proceedings of COLING* 2016

Mike Thelwall, Kevan Buckley, and Georgios Paltoglou. 2012. Sentiment strength detection for the social web. *Journal of the American Society for Information Science and Technology*, 63(1):163–173.

Peter D Turney. 2002. Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In *Proceedings of ACL*, pages 417–424

D. Vo and Y. Zhang, Target-dependant twitter sentiment classification with rich automatic features, *IJCAI 2015*.

Sida Wang and Christopher D Manning. 2012. Baselines and bigrams: Simple, good sentiment and topic classification. In *Proceedings of ACL* 2012, pages 90–94

.