ECNU at SemEval-2016 Task 4: An Empirical Investigation of Traditional NLP Features and Word Embedding Features for Sentence-level and Topic-level Sentiment Analysis in Twitter

Yunxiao Zhou[1], **Zhihua Zhang**[1], **Man Lan**[1,2*]
[1]Department of Computer Science and Technology,
East China Normal University, Shanghai, P.R.China
[2]Shanghai Key Laboratory of Multidimensional Information Processing
{10122130215, 51131201039}@ecnu.cn, mlan@cs.ecnu.edu.cn*

Abstract

This paper reports our submissions to Task 4, i.e., Sentiment Analysis in Twitter (SAT), in SemEval 2016, which consists of five subtasks grouped into two levels: (1) sentence level, i.e., message polarity classification (subtask A), and (2) topic level, i.e., tweet classification and quantification according to two-point scale (subtask B and D) or five-point scale (subtask C and E). We participated in all these five subtasks. To address these subtasks, we investigated several traditional Natural Language Processing (NLP) features including sentiment lexicon, linguistic and domain specific features, and word embedding features together with supervised machine learning methods. Officially released results showed that our systems rank above average.

1 Introduction

In recent years, with the emergence of social media, more and more users have shared and obtained information through microblogging websites, such as Twitter. As a result, a huge amount of available data attracts a lot of researchers. SemEval 2016 provides such a universal platform for researchers to explore in the task of Sentiment Analysis in Twitter (Nakov et al., 2016) (Task 4), which includes five subtasks grouped into two levels, i.e., sentence level and topic level. Subtask A is a sentence level task aiming at sentiment polarity classification of the whole tweet. The other four subtasks are at topic level, i.e., given one topic, the sentiment polarity of tweets are classified or assigned by a two-point scale (i.e., subtask B and D) and by a five-point scale (i.e., subtask C and E). Specifically, subtask B is to identify the sentiment polarity label (i.e, *Positive* and *Negative*) of tweets with respect to the given topic while subtask D aims at estimating the sentiment distribution of tweets with respect to the given topic. Both subtask B and D are on a two-point scale. The purposes of subtask C and E are similar with that of subtask B and D, except for using a five-point scale, that is, the class labels are of 5 values, i.e, 2, 1, 0, -1 and -2 representing *Very Positive*, *Positive*, *Neutral*, *Negative* and *Very Negative*.

Given the character limitations on tweets, sentiment orientation classification on tweets can be regarded as a sentence-level sentiment analysis. Many researchers focus on feature engineering to improve the performance of SAT. For example, (Turian et al., 2010; Liu, 2012; Zhang et al., 2006) showed that one-hot representation on n-gram features is a relatively strong baseline. Furthermore, (Mohammad et al., 2013) proposed a state-of-the-art model which implemented several sentiment lexicons and a variety of manual features. Apart from the traditional methods, more and more researchers have paid their attention to use deep learning methods. Word embedding is one of such methods, where each word is represented as a continuous, low-dimension vector and has been applied into NLP tasks as a critical and fundamental step. Commonly, there are several types of word embedding models, e.g., Bengio proposed a *Neural Probabilistic Language Model (NNLM)* in (Bengio et al., 2003) to learn distributed representation for each word and Mikolov simplified the structure of *NNLM* and presented t-

Proceedings of SemEval-2016, pages 261–266,
San Diego, California, June 16-17, 2016.

wo efficient log-linear models in (Mikolov et al., 2013). Moreover, (Zhang and Lan, 2015; Tang et al., 2014) further proposed learning sentiment-based word embeddings to settle SAT. Meanwhile, topic-based opinion always adheres on certain words or phrases rather than whole tweet. To address topic-based SAT, (Wang et al., 2011) used the hashtag information, (Lin and He, 2009) utilized the topic model to extract topic information from tweets and (Zhang et al., 2015) picked out related words rather than all words in whole tweet as pending words for consequential feature extraction.

Previous work showed that feature engineering has a significant impact on this task. Thus, in this work, we presented multiple types of traditional NLP features to perform SAT, e.g., sentiment lexicon features (e.g., *MPQA, IMDB, Bing Liu opinion lexicon*, etc), linguistic features (e.g., negations, n-gram at the word level and character level, etc) and tweet specific features (e.g., emoticons, capital words, elongated words, hashtags, etc,). Besides, the word embedding features were adopted. We also performed a series of experiments to select effective feature subsets and supervised machine learning algorithms with optimal parameters.

The rest of this paper is organized as follows. Section 2 describes our system framework including preprocessing, feature engineering, evaluation metrics, etc. The experiments are reported in Section 3. Finally, this work is concluded in Section 4.

2 System Description

2.1 Data Preprocessing

With the aid of approximate $5,000$ abbreviations and slangs collected from Internet, we converted the informal writing into regular forms, e.g., *"asap"* replaced by *"as soon as possible"*, *"3q"* replaced by *"thank you"*, etc. And we recovered the elongated words to their original forms, e.g., *"soooooo"* to *"so"*. Finally, the processed data was performed for tokenization, POS tagging and parsing by using *CMU Parsing tools* (Owoputi et al., 2013).

2.2 Feature Engineering

We used four types of features, i.e, linguistic features (e.g., negations, n-gram, etc), tweet specific features (e.g., emoticons, all-caps, hashtags, etc), sentiment lexicon features (the score calculated from eight sentiment lexicons) and word embedding features.

2.2.1 Linguistic Features:

- *Character n-grams:* The character-level n-grams are used, where $n = \{3, 4, 5\}$.
- *Word n-grams:* The word-level *unigram*s, *bigram*s, *trigram*s and 4-*gram*s are adopted.
- *POS:* The absolute frequency of each part-of-speech tag is recorded.
- *Negation:* Negation in a message always reverses its sentiment orientation. We collected 29 negations from Internet and recorded the frequency of negations in the whole tweet.
- *Cluster:* The *CMU TweetParser tool* provides $1,000$ token clusters produced with the Brown clustering algorithm on 56 million English language tweets. We recorded the existence of tokens in tweets with respect to these $1,000$ clusters.
- *Dependency triple:* The dependency tree is generated by *Stanford Parser tool* and each tweet contains several dependency triples (e.g., *relation(government, dependent)*). We used a binary feature to record if a dependency triple is present or absent in a tweet.

2.2.2 Tweet Specific Features:

- *Punctuation:* Punctuation marks (e.g, exclamation mark (!) and question mark (?)) usually indicate the strength of sentiment. Therefore, we recorded the numbers of these marks in isolation and in combination. Besides, the position of punctuation in tweet is also an important clue for sentiment, thus we used a binary feature to indicate whether it is the last token of tweet.
- *All-caps:* The number of words in uppercase is recorded.
- *Hashtag:* We recorded the number of hashtags in the tweet.

- *Emoticon:* We collected 67 emoticons from Internet and this feature type records the number of positive and negative emoticons respectively. Moreover, two binary values are to record whether the last token is a positive or negative emoticon respectively.

- *Elongated:* It indicates the number of elongated words in the raw text of tweet.

2.2.3 Sentiment Lexicon Features (SentiLexi):

We employed the following eight sentiment lexicons to extract sentiment lexicon features: *Bing Liu lexicon*[1], *General Inquirer lexicon*[2], *AFINN*[3], *IMDB*[4], *MPQA*[5], *NRC Emotion Sentiment Lexicon*[6], *NRC Hashtag Sentiment Lexicon*[7], and *NRC Sentiment140 Lexicon*[8]. Generally, we transformed the scores of all words in all sentiment lexicons to the range of -1 to 1, where the positive number indicates positive sentiment and the minus sign denotes negative sentiment.

The following six scores are calculated on the whole data for each sentiment lexicon: (1) the ratio of positive words to all words, (2) the ratio of negative words to all words, (3) the maximum sentiment score, (4) the minimum sentiment score, (5) the sum of sentiment scores, and (6) the sentiment score of the last word in tweet. If a word does not exist in one sentiment lexicon, its corresponding score is set to 0.

2.2.4 Word Embedding Features:

In this work, we employed three different types of word vectors. The general word vectors are trained by Google on huge amount of News, which is a different domain from Twitter. The other two sentiment word vectors are both trained on tweets but using different methods. The purpose of this feature type is to examine the effects of word embedding and sentiment word embedding on performance.

- *General Word Vector (GeneralW2V):* We adopted the *word2vec* tool[9] to obtain word vectors with the dimensionality of 300 (i.e., *GeneralW2V*), trained on 100 billion words from Google News.

- *Sentiment Word Vector (SWV):* (Zhang and Lan, 2015) proposed a *Combined-Sentiment Word Embedding Model* to learn sentiment word vectors (*SWV*) for sentiment analysis task. In this work, we learn *SWV* on *NRC140 tweet corpus*(Go et al., 2009), where the corpus is made up of 1.6 million tweets (0.8 million positive and 0.8 million negative). The vector dimension is set as 100.

- *Sentiment-specific Word Embedding (SSWE):* Similar with *SWV*, the sentiment-specific word embedding model proposed by (Tang et al., 2014) used a multi-hidden-layers neural network to train *SSWE* with dimensionality of 50.

To convert the above word vectors into a sentence vector, we simply adopted the *min*, *max* and *average* operations. Obviously, this combination strategy neglects the word sequence in tweet but it is simple and straightforward. As a result, the final sentence vector $V(s)$ was concatenated by $V_{min}(s)$, $V_{max}(s)$ and $V_{average}(s)$.

2.3 Evaluation Metrics

For subtask A, we used the macro-averaged F score of positive and negative classes (i.e., $F_{macro} = \frac{F_{pos}+F_{neg}}{2}$) to evaluate the performance. Subtask B and D just contain positive and negative labels. The official metric for subtask B is *macro-averaged recall* among positive and negative (i.e., $R_{macro} = \frac{R_{Pos}+R_{Neg}}{2}$). As for subtask D, it is *Kullback-Leibler Divergence* (KLD) among distributions of two classes (i.e., $KLD(pos, neg) = \sum_{c_j \in pos,neg} P(c_j) \cdot log\frac{P(c_j)}{\hat{P}(c_j)}$, where P denotes the probability of predicted label and $\hat{P}$ is the probability of gold label). There are 5 classes existing in subtask C and E, and

[1]http://www.cs.uic.edu/liub/FBS/sentiment-analysis.html#lexicon

[2]http://www.wjh.harvard.edu/inquirer/homecat.htm

[3]http://www2.imm.dtu.dk/pubdb/views/publication_details.php?id=6010

[4]http://anthology.aclweb.org//S/S13/S13-2.pdf#page=444

[5]http://mpqa.cs.pitt.edu/

[6]http://www.saifmohammad.com/WebPages/lexicons.html

[7]http://www.umiacs.umd.edu/saif/WebDocs/NRC-Hashtag-Sentiment-Lexicon-v0.1.zip

[8]http://help.sentiment140.com/for-students/

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

the organizers adopted *Macroaveraged Mean Absolute Error* (i.e., MAE^M) and *Earth Mover's Distance* (EMD) among 5 predefined classes for two subtasks respectively, where the detail information of two metrics for evaluation is described in the official document available on the website[10].

3 Experiments

3.1 Datasets

Since only tweet IDs are provided by organizers, different participants may collect different numbers of tweets due to missing tweets or system errors. Subtask B and D are of the same data set. And subtask C and E share one common data set. The statistics of all datasets for these subtasks are shown in Tables 1, 2, and 3, respectively.

For subtask A, the training data set consists of four parts which are shown in Table 1, i.e., $2013train$, $2013dev$, $2016train$ and $2016dev$. The data set $2013train$ means SemEval-2013 Task 2 training data set (Nakov et al., 2013), and the following data sets are named in the same way. Actually, in consideration of the difference of polarity distribution between data set $2016devtest$ and $2013\&2014test$, we just adopted $2016devtest$ as development data. For subtask B, C, D and E, the data is divided into many topic sets.

dataset		Positive	Negative	Neutral	Total
train	2013train	3,250(37%)	1,276(15%)	4,151(48%)	8,677
	2013dev	575(35%)	340(20%)	739(46%)	1,654
	2016train	2,839(51%)	787(14%)	1,892(34%)	5,518
	2016dev	772(42%)	359(20%)	702(38%)	1,833
dev	2016devtest	886(49%)	287(16%)	626(35%)	1,799
	2013&2014test	5,078(40%)	2,142(16%)	5,580(44%)	12,800
test	2016test	7,059(34%)	3,231(16%)	10,342(50%)	20,632

Table 1: Statistics of data sets in training (train), development (dev), test (test) data for subtask A.

dataset	Positive	Negative	Total
train	4,191(81%)	997(19%)	5,188
dev	1,027(81%)	238(19%)	1,265
test	8,212(78%)	2,339(22%)	10,551

Table 2: Statistics of data sets in training, development, test data for subtask B and D.

3.2 Experiments on Training Data

In order to improve the performance of sentiment analysis, we performed feature selection experiments on all subtasks and the optimum feature sets are shown in Table 4. From Table 4, it is interesting to find that: (1) Negation features and tweet specific features such as emoticon and all-caps make contributions to almost all subtasks. (2) The feature set with the best performance of subtask B is not quite beneficial for subtask D even though they have the same data set, perhaps because of the essential difference between binary classification and binary quantification: in the latter, errors of different polarity compensate each other. The similar observation is found in subtask C and E. (3) The sentiment lexicon features make contributions to performance improvement of subtask A, B and C, but are not quite useful for subtask D and E. A possible reason is that the latter two subtasks focus on quantification analysis while the sentiment lexicon only contains sentiment orientation rather than sentiment strength. (4) The word embedding features are not as effective as expected. It maybe because we obtained sentence vectors by the simplest combination method described above, which does not take into account contextual information and semantic relations among words.

dataset	2	1	0	-1	-2	Total
train	475(6%)	3,815(50%)	2,295(30%)	906(12%)	120(2%)	7,611
dev	123(7%)	900(50%)	535(30%)	211(12%)	29(1%)	1,798
test	382(2%)	7,830(38%)	10,081(49%)	2,201(10%)	138(1%)	20,632

Table 3: Statistics of data sets in training, development, test set for subtask C and E.

Besides, since subtask B, C, D and E focus on topic-level sentiment analysis, we tried to extract features from related words rather than whole tweet. But the preliminary experimental results showed that extracting features from related words underperformed the latter strategy for extracting features. The possible reason is that in many cases a tweet only has one single sentiment polarity. Thus the sentiment polarity of sentence can always represent that of topic and extracting features from the related words may drop important information.

3.3 Learning Algorithm

For these subtasks, we examined several supervised machine learning classification algorithms with different parameters (e.g., Logistic Regression with c=$\{0.1, 1\}$, Support Vector Machine with $kernel$=$\{linear, rbf\}$, c=$\{0.01, 0.1, 1\}$, Random

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

Features		Subtask A (F_{macro})	Subtask B (R_{macro})	Subtask C (MAE^M)	Subtask D(KLD)	Subtask E (EMD)
SentiLexi	Sentiment Lexicon	√	√	√		
Linguistic	Unigram		√			√
	Bigram		√			
	Trigram	√	√		√	
	4-gram	√				
	3-char			√	√	
	4-char			√	√	√
	5-char		√			
	POS	√				
	Negation	√	√	√	√	√
	Cluster		√			
	Dependency Triple			√	√	√
Tweet-specific	All-caps	√	√	√	√	√
	Elongated		√		√	
	Punctuation	√			√	
	Emoticon		√	√	√	√
	Hashtag	√				
Word Embedding	GoogleW2V	√	√			
	SWV	√				
	SSWE		√			
Results		0.63	0.82	0.87	0.01	0.03

Table 4: Results of feature selection experiments for subtask A, B, C, D and E in terms of the corresponding evaluation metrics on the training data, where √ indicates this feature was employed in the corresponding subtask system.

Forest with n=$\{20, 50, 100, 400, 1000, 2000\}$, SGD with $loss$=$\{hinge, log\}$, etc). Finally, Logistic Regression with c = 1 implemented in *Liblinear*[11] was adopted for all five subtasks for its good performance in preliminary experiments.

3.4 Results on Test Data

Based on the optimum feature sets shown in Table 4 and configuration of classifiers described above, we trained separate models for each subtask.

Subtask	System	Score
A	ECNU	0.585(10)
	SwissCheese	0.633(1)
	SENSEI-LIF	0.630(2)
B	ECNU	0.768(4)
	Tweester	0.797(1)
	LYS	0.791(2)
C	ECNU	0.806(2)
	twise	0.719(1)
	PUT	0.860(3)
D	ECNU	0.121(10)
	finki	0.034(1)
	LYS	0.053(2)
E	ECNU	0.341(5)
	QCRI	0.243(1)
	finki	0.316(2)

Table 5: Performance of our systems and the top-ranked systems on all five subtasks. The numbers in the brackets are the official ranking.

Table 5 shows the results of our systems and the top-ranked systems on all five subtasks. Our systems ranked $10th$ out of 34 submissions for subtask A, $4th/19$ for subtask B, $2nd/11$ for subtask C, $10th/14$ for subtask D and $5th/10$ for subtask E. Compared with the top ranked systems, there is much room for improvement in our work. Although word embedding features were adopted in this work, we used the simplest combination method to convert word vectors to sentence vectors. The effective convolution method is expected to be able to improve the performance of sentiment analysis.

4 Conclusion

In this paper, we extracted several traditional NLP features(e.g., linguistic features, tweet specific features, etc) and word embedding features from whole tweet and constructed classifiers using supervised machine learning algorithms to accomplish sentiment analysis towards sentence level(i.e., subtask A) and topic level(i.e., subtask B, C, D and E). Word embedding features are not as effective as expected since the way of using these features are quite simple and naive, thus it is too hasty to make a conclusion that the word embedding features make marginal contribution. In future work, we consider to focus on developing advanced convolution neural network to model sentence with the aid of sentiment word vector.

Acknowledgements

This research is supported by grants from Science and Technology Commission of Shanghai Municipality (14DZ2260800 and 15ZR1410700), Shanghai Collaborative Innovation Center of Trustworthy

[11] https://www.csie.ntu.edu.tw/ cjlin/liblinear/

Software for Internet of Things (ZF1213).

References

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. 2003. A neural probabilistic language model. *The Journal of Machine Learning Research*, 3:1137–1155.

Alec Go, Richa Bhayani, and Lei Huang. 2009. Twitter sentiment classification using distant supervision. *CS224N Project Report, Stanford*, pages 1–12.

Chenghua Lin and Yulan He. 2009. Joint sentiment/topic model for sentiment analysis. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 375–384. ACM.

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

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

Saif Mohammad, Svetlana Kiritchenko, and Xiaodan Zhu. 2013. NRC-Canada: Building the state-of-the-art in sentiment analysis of tweets. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 321–327, Atlanta, Georgia, USA, June.

Preslav Nakov, Sara Rosenthal, Zornitsa Kozareva, Veselin Stoyanov, Alan Ritter, and Theresa Wilson. 2013. SemEval-2013 Task 2: Sentiment analysis in Twitter. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 312–320, Atlanta, Georgia, USA, June.

Preslav Nakov, Alan Ritter, Sara Rosenthal, Veselin Stoyanov, and Fabrizio Sebastiani. 2016. SemEval-2016 task 4: Sentiment analysis in Twitter. In *Proceedings of the 10th International Workshop on Semantic Evaluation*, SemEval '16, San Diego, California, June. Association for Computational Linguistics.

Olutobi Owoputi, Brendan O'Connor, Chris Dyer, Kevin Gimpel, Nathan Schneider, and Noah A Smith. 2013. Improved part-of-speech tagging for online conversational text with word clusters. In *HLT-NAACL*, pages 380–390.

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 Annual Meeting of the Association for Computational Linguistics*, pages 1555–1565.

Joseph Turian, Lev Ratinov, and Yoshua Bengio. 2010. Word representations: a simple and general method for semi-supervised learning. In *The Annual Meeting of the Association for Computational Linguistics*, pages 384–394. Association for Computational Linguistics.

Xiaolong Wang, Furu Wei, Xiaohua Liu, Ming Zhou, and Ming Zhang. 2011. Topic sentiment analysis in Twitter: a graph-based hashtag sentiment classification approach. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 1031–1040.

Zhihua Zhang and Man Lan. 2015. Learning sentiment-inherent word embedding for word-level and sentence-level sentiment analysis. In *2015 International Conference on Asian Language Processing, IALP*.

Kuo Zhang, Hui Xu, Jie Tang, and Juanzi Li. 2006. Keyword extraction using support vector machine. In *Advances in Web-Age Information Management*, pages 85–96. Springer.

Zhihua Zhang, Guoshun Wu, and Man Lan. 2015. Ecnu: Multi-level sentiment analysis on twitter using traditional linguistic features and word embedding features. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 561–567, Denver, Colorado, June. Association for Computational Linguistics.