ECNU at SemEval-2017 Task 4: Evaluating Effective Features on Machine Learning Methods for Twitter Message Polarity Classification

Yunxiao Zhou[1], **Man Lan**[1,2*] ,**Yuanbin Wu**[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
51164500061@stu.ecnu.edu.cn, {mlan, ybwu}@cs.ecnu.edu.cn

Abstract

This paper reports our submission to subtask A of task 4 (Sentiment Analysis in Twitter, SAT) in SemEval 2017, i.e., Message Polarity Classification. We investigated several traditional Natural Language Processing (NLP) features, domain specific features and word embedding features together with supervised machine learning methods to address this task. Officially released results showed that our system ranked 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. The study on this platform is increasingly drawing attention of many researchers and organizations. SemEval 2017 provides a universal platform for researchers to explore sentiment analysis in Twitter (Rosenthal et al., 2017) (Task 4, Sentiment Analysis in Twitter, SAT) which includes five subtasks, and we participated in subtask A: Message Polarity Classification. It aims at sentiment polarity classification of the whole tweet on a three-point scale(i.e., *Positive*, *Negative* and *Neutral*).

Given the character limitations on tweets, the sentiment orientation classification on tweets can be regarded as a sentence-level sentiment analysis task. Following previous work (Mohammad et al., 2013; Zhang et al., 2015; Wasi et al., 2014), we adopted a rich set of traditional NLP features, i.e., linguistic features (e.g., word n-gram, part-of-speech (POS) tags, etc), sentiment lexicon features (i.e., the scores calculated from eight sentiment lexicons), and domain content features (e.g., emoticons, capital words, elongated words, etc). In consideration of rich information in the metadata of tweets, we also extracted metadata features from tweets. Moreover, several word embeddings (including general word embeddings and sentiment word vectors) were adopted. We performed a series of experiments to explore the effectiveness of each type of features and supervised machine learning algorithms.

2 System Description

We first performed data preprocessing, then extracted several types of features from tweets and metadata for sentiment analysis and constructed supervised classification models for this task.

2.1 Data Preprocessing

Firstly, we used about $5,000$ abbreviations and slangs[1] to convert the informal writing into regular forms, e.g., "*3q*" replaced by "*thank you*", "*asap*" replaced by "*as soon as possible*", etc. And we recovered the elongated words to their original forms, e.g., "*soooooo*" to "*so*". Then the processed data was performed for tokenization, POS tagging, parsing, stemming and lemmatization using *Stanford CoreNLP* (Manning et al., 2014).

2.2 Feature Engineering

In this task, we evaluated four types of features, i.e, linguistic features, sentiment lexicon features, domain-specific features and word embedding features.

2.2.1 Linguistic Features

- *Word_RF n-grams:* We extracted *unigrams*, *bigrams* and *trigrams* features at two different levels, i.e., the original word level and the word stem level. Considering that different words make different contribution to sentimental expression, for each n-gram feature,

[1] https://github.com/haierlord/resource/blob/master/slangs

Proceedings of the 11th International Workshop on Semantic Evaluations (SemEval-2017), pages 812–816,
Vancouver, Canada, August 3 - 4, 2017. ©2017 Association for Computational Linguistics

we calculated *rf* (relevance frequency) value (Lan et al., 2009) to weight its importance.

- *POS:* Generally, the sentences carrying subjective emotions (i.e., positive and negative sentiment) are inclined to contain more adjectives and adverbs while the sentences without sentiment orientation (i.e., neutral) would contain more nouns. Therefore, we recorded the number of each POS tag in one sentence.

- *Negation:* Negation in a message always reverses its sentiment orientation. We manually collected 29 negations[2] from previous work in (Zhang et al., 2015) and designed two binary features. One is to indicate whether there is any negation in the tweet and the other is to record whether this tweet contains more than one negation.

2.2.2 Sentiment Lexicon Features (SentiLexi)

We employed the following eight sentiment lexicons to extract sentiment lexicon features: *Bing Liu lexicon*[3], *General Inquirer lexicon*[4], *IMDB*[5], *MPQA*[6], *NRC Emotion Sentiment Lexicon*[7], *AFINN*[8], *NRC Hashtag Sentiment Lexicon*[9], and *NRC Sentiment140 Lexicon*[10]. Since certain words may consist of mixed sentiments based on different contexts, it is not appropriate to assign only one sentiment score for this type of word. Therefore, the first five lexicons use two values for each word to represent its sentiment scores, i.e., one for positive sentiment and the other for negative sentiment. In order to unify the formats, we transformed the two scores into a one-dimensional value by subtracting negative emotion scores from positive emotion scores. Then in all sentiment lexicons, for each word the positive number indicates a positive emotion and the minus sign represents a negative emotion.

Given a tweet, we first converted all words into lowercase. Then on each sentiment lexicon, we calculated the following six scores for one message: (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, (6) the sentiment score of the last word in tweet. If the word does not exist in one sentiment lexicon, its corresponding score is set to 0.

2.2.3 Domain-Specific Features

Domain-specific features are extracted from two sources. One is from the content of tweets and the other is from tweet metadata information.

Firstly, the domain specific features extracted from tweet content are shown as follows:

- *All-caps:* One binary feature is to check whether this tweet has words in uppercase.

- *Bag-of-Hashtags:* We constructed a vocabulary of hashtags appearing in the training data and then adopted the bag-of-hashtags method for each tweet.

- *Elongated:* It indicates whether the raw text of tweet contains words with one continuous character repeated more than two times, e.g., “goooood”.

- *Emoticon:* We manually collected 67 emoticons from Internet[11] and designed the following 4 binary features:
  - to record the presence or absence of positive and negative emoticons respectively in the tweet;
  - to record whether the last token is a positive or a negative emoticon.

- *Punctuation:* Punctuation marks (e.g, exclamation mark (!) and question mark (?)) usually indicate the expression of sentiment. Therefore, we designed the following 6 binary features to record:
  - whether the tweet contains an exclamation mark;
  - whether the tweet contains more than one exclamation mark;
  - whether the tweet has a question mark;
  - whether the tweet contains more than one question mark;

[2] https://github.com/haierlord/resource

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

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

[5] http://www.aclweb.org/anthology/S13-2067

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

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

[8] http://www2.imm.dtu.dk/pubdb/views/publication details.php?id=6010

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

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

[11] https://github.com/haierlord/resource/blob/master/Emoticon.txt

  - whether the tweet contains both exclamation marks and question marks;
  - whether the last token of this tweet is an exclamation or question mark.

Recently, several studies using tweet metadata are reported to have good performance on sentiment classification (Tang et al., 2015; Chen et al., 2016). Inspired by them, the second tweet domain-specific features we used are extracted from tweet metadata information. We first used Twitter API[12] to collect tweet metadata and then designed the following two types of features.

- *Tweet metadata*: Two binary features are to check whether this tweet has been *retweeted* and whether it has been *liked* by authenticating users. Furthermore, given one tweet, two numeric features are to record the count of *retweeted* and the count of *liked*. These two numeric features were standardized using [0-1] normalization.

- *User metadata*: In addition to the metadata of tweets, users who write tweets may also contain useful information. Thus the following 5 user metadata features are collected: *friends count*, *followers count*, *statuses count*, *verified* and *default profile image*. The first three numeric items are standardized using [0-1] normalization and the rest are binary values.

In total, we collected 9 metatdata features.

2.2.4 Word Embedding Features

Word embedding is a continuous-valued vector representation for each word, which usually carries syntactic and semantic information. In this work, we employed five different types of word embeddings. The *GoogleW2V* and *GloVe* are two pre-trained word vectors downloaded from Internet. The former is pre-trained on News domain and the latter is pre-trained on tweets. We also trained the *TweetW2V* on tweet domain using Google word2vec tool. Besides, taking into consideration the sentiment information of each word, previous work in (Tang et al., 2014) and (Lan et al., 2016) presented methods to learn sentiment word vectors rather than general word vectors. The last two word vectors i.e., *SWV* and *SSWE*, are expected to endow word embeddings with sentiment information and semantic information.

- *GoogleW2V:* The 300-dimensional word vectors are pre-trained on Google News with 100 billion words, available in Google[13].

- *GloVe:* The 100-dimensional word vectors are pre-trained on Twitter using GloVe, available in *GloVe*[14].

- *TweetW2V:* We adopted the *word2vec* tool[15] to obtain 100-dimensional word vectors (i.e., TweetW2V) 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).

- *SWV:* Our previous work in (Lan et al., 2016) proposed a combined model to learn sentiment word vector (SWV) for sentiment analysis task. In this work, we learned the *SWV* on *NRC140 tweet corpus* and the dimension is set as 200.

- *SSWE:* The sentiment-specific word embedding (SSWE) model has been proposed by (Tang et al., 2014) used a multi-hidden-layers neural network to train *SSWE* on 10 million tweets with dimensionality of 50.

In order to obtain a sentence vector, we simply adopted the *min*, *max* and *mean* pooling operations on all words in a tweet message. 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 as $[V_{min}(s) \bigoplus V_{max}(s) \bigoplus V_{mean}(s)]$.

2.3 Learning Algorithms

We granted this task as a three-way classification task and explored four supervised machine learning algorithms: Logistic Regression (LR) implemented in *Liblinear*[16], Support Vector Machine (SVM), Stochastic Gradient Descent (SGD) and AdaBoost all implemented in *scikit-learn tools*[17].

2.4 Evaluation Metric

To evaluate the system performance, the official evaluation criterion is *macro-averaged recall*,

[12] https://dev.twitter.com/overview/api

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

[14] http://nlp.stanford.edu/projects/glove

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

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

[17] http://scikit-learn.org/stable/

which is calculated among three classes (i.e., positive, negative and neutral) as follows:

$$R_{macro} = \frac{R_{Pos} + R_{Neg} + R_{Neu}}{3}$$

3 Experiments

3.1 Datasets

For training set, the organizers provided only the list of tweet ID and a script for all participants to collect tweets and their corresponding metadata. However, since not all tweets and their metadata are available when downloading, participants may collect slightly different numbers of tweets for training data. Table 1 shows the statistics of the tweets we collected in our experiments. Similarly, due to missing tweets or metadata and system errors when downloading, the metadata of training, development and test set is not complete. Specifically, approximately 21% training, 18% development and 39% test sets lost their metadata information.

Dataset	Positive	Negative	Neutral	Total
train	7,310 (43%)	2,613 (15%)	7,077 (42%)	17,000
dev	7,059 (34%)	3,231 (16%)	10,342 (50%)	20,632
test	2,375 (19%)	3,972 (32%)	5,937 (48%)	12,284

Table 1: The statistics of data sets in training, development and test data. The numbers in brackets are the percentages of different classes in each data set.

3.2 Experiments on Training Data

Firstly, in order to explore the effectiveness of each feature type, we performed a series of experiments. Table 2 lists the comparison of different contributions made by different features on development set with *Logistic Regression* algorithm. We observe the following findings.

(1) All feature types make contributions to sentiment polarity classification. Their combination achieves the best performance (i.e., 63.14%).

(2) Linguistic features act as baseline and have shown their effectiveness for sentiment polarity prediction. Besides, SentiLexi makes more contributes than other domain-specific and word embeddings features. Since sentiment lexicons are constructed by expert knowledge, it is beneficial for tweet sentiment polarity prediction.

(3) The domain-specific metadata is not as effective as expected. One possible reason results from the missing metadata downloaded by Twitter API.

Features	$\mathbf{R_{macro}}$
Linguistic	0.584
.+SentiLexi	0.621 (+0.037)
.+Domain Metadata	0.623 (+0.002)
.+Domain Content	0.628 (+0.005)
.+Word Embedding	0.631 (+0.003)

Table 2: Performance of different features on development data. ".+" means to add current features to the previous feature set. The numbers in the brackets are the performance increments compared with the previous results.

Algorithms	$\mathbf{R_{macro}}$
LR	0.631
SVM	0.612
SGD	0.623
AdaBoost	0.603

Table 3: Performance of different learning algorithms on development data.

Secondly, we also explored the performance of different learning algorithms. Table 3 lists the comparison of different supervised learning algorithms with all above features. Clearly, Logistic Regression algorithm outperformed other algorithms.

Therefore, the system configuration for submission is all features and LR algorithm.

3.3 Results on Test Data

Table 4 shows the results of our system and the top-ranked systems provided by organizers for this sentiment classification task. Compared with the top ranked systems, there is much room for improvement in our work. There are several possible reasons for this performance lag. First, although the linguistic features are effective, the dimensionality of *word_RF n-gram* features is quite huge (approximately $79K$ n-grams), which dominates the performance of classification rather than other low dimension features. Second, the usage of word embeddings is simple and straightforward, which neglects the word sequence and sentence structure. Third, the effects of metadata may be reduced due to lots of missing metadata.

Team ID	$\mathbf{R_{macro}}$	F_{macro}	Acc
ECNU	0.628 (15)	0.613 (13)	0.630 (12)
DataStories	0.681 (1)	0.677 (2)	0.651 (5)
BB_twtr	0.681 (1)	0.685 (1)	0.658 (3)
LIA	0.676 (3)	0.674 (3)	0.661 (2)

Table 4: Performance of our system and the top-ranked systems. The numbers in the brackets are the official rankings.

4 Conclusion

In this paper, we extracted several traditional NLP features, domain specific features and word embedding features from tweets and their metadata and adopted supervised machine learning algorithms to perform sentiment polarity classification. The system performance ranks above average. In future work, we consider to focus on developing neural networks method to model sentence with the aid of sentiment word vectors.

Acknowledgements

This research is supported by grants from Science and Technology Commission of Shanghai Municipality (14DZ2260800 and 15ZR1410700), Shanghai Collaborative Innovation Center of Trustworthy Software for Internet of Things (ZF1213) and NSFC (61402175).

References

Huimin Chen, Maosong Sun, Cunchao Tu, Yankai Lin, and Zhiyuan Liu. 2016. Neural sentiment classification with user and product attention. In *Proceedings of EMNLP*.

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

Man Lan, Chew Lim Tan, Jian Su, and Yue Lu. 2009. Supervised and traditional term weighting methods for automatic text categorization. *IEEE transactions on pattern analysis and machine intelligence* 31(4):721–735.

Man Lan, Zhihua Zhang, Yue Lu, and Ju Wu. 2016. Three convolutional neural network-based models for learning sentiment word vectors towards sentiment analysis. In *Neural Networks (IJCNN), 2016 International Joint Conference on*. IEEE, pages 3172–3179.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. pages 55–60.

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)*. Atlanta, Georgia, USA, pages 321–327.

Sara Rosenthal, Noura Farra, and Preslav Nakov. 2017. SemEval-2017 task 4: Sentiment analysis in Twitter. In *Proceedings of the 11th International Workshop on Semantic Evaluation*. Association for Computational Linguistics, Vancouver, Canada, SemEval '17.

Duyu Tang, Bing Qin, and Ting Liu. 2015. Learning semantic representations of users and products for document level sentiment classification. In *ACL (1)*. pages 1014–1023.

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.

Sabih Bin Wasi, Rukhsar Neyaz, Houda Bouamor, and Behrang Mohit. 2014. Cmuq@ qatar: Using rich lexical features for sentiment analysis on twitter. *SemEval 2014* page 186.

Zhihua Zhang, Guoshun Wu, and Man Lan. 2015. Ecnu: Multi-level sentiment analysis on twitter using traditional linguistic features and word embedding features. *Proceedings of SemEval* pages 561–567.