ECNU: Multi-level Sentiment Analysis on Twitter Using Traditional Linguistic Features and Word Embedding Features

Zhihua Zhang, Guoshun Wu, Man Lan*
Shanghai Key Laboratory of Multidimensional Information Processing
Department of Computer Science and Technology,
East China Normal University Shanghai 200241, P. R. China
{51131201039, 51141201064}@ecnu.cn, mlan@cs.ecnu.edu.cn*

Abstract

This paper reports our submission to task 10 (Sentiment Analysis on Tweet, SAT) (Rosenthal et al., 2015) in SemEval 2015 , which contains five subtasks, i.e., contextual polarity disambiguation (subtask A: expression-level), message polarity classification (subtask B: message-level), topic-based message polarity classification and detecting trends towards a topic (subtask C and D: topic-level), and determining sentiment strength of twitter terms (subtask E: term-level). For the first four subtasks, we built supervised models using traditional features and word embedding features to perform sentiment polarity classification. For subtask E, we first expanded the training data with the aid of external sentiment lexicons and then built a regression model to estimate the sentiment strength. Despite the simplicity of features, our systems rank above the average.

1 Introduction

In the past few years, hundreds of millions of people shared and expressed their opinions through microblogging websites, such as Twitter. The study on this platform is increasingly drawing attention of many researchers and organizations. Given the character limitations on tweets, the sentiment orientation classification on tweets is usually analogous to the sentence-level sentiment analysis (Kouloumpis et al., 2011; Kim and Hovy, 2004; Yu and Hatzivassiloglou, 2003). However, considering opinions adhering on different topics and expressed by various expression words in tweets, (Wang et al., 2011; Jiang et al., 2011; Chen et al., 2012) have investigated various ways to settle these target dependent issues. Recently, inspired by (Mikolov et al., 2013a) using neural network to construct distributed word representation (word embedding), several researchers employed neural network to perform sentiment analysis. For example, (Kim, 2014; dos Santos and Gatti, 2014) adopted convolutional neural networks to learn sentiment-bearing sentence vectors, and (Mikolov et al., 2013b) proposed *Paragraph vector* which outperformed bag-of-words model for sentiment analysis.

The task of Sentiment Analysis in Twitter (SAT) in SemEval 2015 consists of five subtasks. The first three subtasks focus on determining the polarity of the given tweet, phrase or topic (i.e., subtask A aims at classifying the sentiment of a marked instance in a given message, subtask B is to determine the polarity of the whole message and subtask C focuses on identifying the sentiment of the message towards the given topic). The fourth subtask D is to detect the sentiment trends of a given set of messages towards a topic from the same period of time. The last subtask E is to predict a score between 0 and 1, which is indicative of the strength of association of twitter terms with positive sentiment.

Following previous works (Rosenthal et al., 2014; Zhao et al., 2014; Mohammad et al., 2013; Evert et al., 2014; Mohammad et al., 2013; Wasi et al., 2014), we adopted a rich set of traditional features, e.g., linguistic features (e.g., *n-gram* at word level, part-of-speech (POS) tags, negations, etc), sentiment lexicon features (e.g., MPQA, Bing Liu opinion lexicon, SentiWordNet, etc) and twitter specif-

Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), pages 561–567,
Denver, Colorado, June 4-5, 2015.

ical features (e.g., the number of *URL*, emoticons, capital words, elongated words, hashtags, etc). Besides, inspired by (Kim, 2014; Mikolov et al., 2013b), we also employed novel word embedding features in these tasks.

The remainder of this paper is organized as follows. Section 2 reports our systems including preprocessing, feature engineering, evaluation metrics, etc. The data sets and experiments descriptions are shown in Section 3. Finally, we conclude this paper in Section 4.

2 System Description

For subtask A and B, we compared two classifiers built on traditional NLP features (linguistic and Sentiment Lexicon) and word embedding features respectively. We also combined the results of the above two classifiers by summing up the predicted probability score. Due to time limitation, for subtask C and D, we only used the traditional feature sets to build a classifier. Unlike the above four subtasks, for subtask E we built a regression model to calculate a sentimental strength score for each target term with the aid of sentiment lexicon score features and word embedding features.

2.1 Data Preprocessing

Firstly, we collected about $5,000$ slangs or abbreviations from Internet to convert the irregular writing to formal forms. By doing this, we also recovered the elongated words to its initial forms, e.g., *"goooooood"* to *"good"*, *"asap"* to *"as soon as possible"*, *"3q"* to *"thank you"*, etc. Then the processed data was performed for tokenization, POS tagging and parsing by using *CMU Parsing tools* (Owoputi et al., 2013).

2.2 Feature Engineering

Although the first four subtasks all focus on sentiment polarity classification, they have very different definitions. For example, since subtask B focuses on sentiment classification on whole tweet, we extract features from all words in tweet. However, the other three subtasks, i.e, A, C, and D, perform sentiment polarity classification only on a certain piece of tweet, i.e., expression words or topic in tweet. Since organizers have provided the annotated target words (for A) and topics (for C and D) for each tweet, we only chose related words rather than all words in whole tweet as pending words for consequential feature extraction. To pick out related words from whole tweet, following (Kiritchenko et al., 2014), for each annotated target word we only treated the surrounding words from parse tree with distance $d \leq 2$ as its relevant words.

In this task, we used four types of features: sentiment lexicon features (the score calculated from seven sentiment lexicons), linguistic features (*n-grams*, POS tagger, negations, etc), tweet-specific features (emoticons, all-caps, hashtag, etc) and word embedding features.

Sentiment Lexicon Features (SentiLexi):

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

Given extracted pending words, we first converted them to lowercase. Then for each sentiment lexicon, we calculated the following five sentimental scores on the processed pending words: (1) the ratio of positive words to pending words, (2) the ratio of negative words to pending words, (3) the maximum sentiment score, (4) the minimum sentiment score, (5) the sum of sentiment scores. If the pending word does not exist in one sentiment lexicon, its corresponding score is set to zero. Specifically, before locating the corresponding term in *SentiWordNet* lexicon, we conducted lemmatization for words and selected its first item in searched results according to its POS tag.

Linguistic Features:

- *Word n-grams:* We first converted all pending

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

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

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

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

[5]http://sentiwordnet.isti.cnr.it/

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

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

words to lowercase and removed URLs, mentions, hashtags, and low frequency (threshold value is 10) words. Then we extracted **unigram** and **bigram** features. Besides, inspired by (Kiritchenko et al., 2014), the words connected on parse tree are extracted as **pairgram**.

- *POS Features:* We recorded the number of nouns (the corresponding POS tags in *CMU parser* are $N, O, \hat{}, S, Z$), verbs (i.e., V, L, M), adjectives (i.e., A), and adverbs (i.e., R) in pending words.

- *Negation Features:* Usually, the sentiment orientation of a message or phrase can be reversed by a modified negation. Thus, we collected 29 negations[8] from Internet and this binary feature is set as 1 or 0 if corresponding negation is present or absent in pending words.

Tweet Specific Features (PAHE):

- *Emoticon:* We gathered 69 emoticons from Internet and this binary feature records whether the corresponding emoticon is present or absent in pending words.

- *Punctuation:* The numbers of exclamations (!) and questions (?) are also noted.

- *All-caps:* It indicates the number of words with uppercase letters.

- *Hashtag:* It is the number of hashtags in the sentence or phrase.

- *Elongated:* It represents the number of words with one character repeated more than two times, e.g., *"gooooood"*.

Word Embedding Features: Word embedding is a continuous-valued representation of the word which usually carries syntactic and semantic information (Zeng et al., 2014). Since a phrase or sentence contains more than one word, usually there are two strategies to convert the words vectors into a sentence vector: (1) summing up all words vectors; (2) rolling up the sequential words to obtain a vector that contains context information (i.e., convolution neutral network). The convolution neural network (*CNN*) is usually employed in image recognition, while many researchers have adopted it in Natural Language Processing (Kim, 2014; dos Santos and Gatti, 2014) and achieved good performance. For subtask A and B, we adopted the *CNN tools* in (Kim, 2014) and extracted the penultimate hidden layer content as the sentence word embedding features to perform classification. For subtask E, we simply adopted the first strategy to sum up the word vectors in the given phrase.

Specifically, in this work we used the publicly available *word2vec* vectors to get the word embedding with dimensionality of 300, which is trained on 100 billion words from Google News (Mikolov et al., 2013b). If a word is not in *word2vec* list, we initialize its vector values to random values.

2.3 Evaluation Metrics

For subtask A, B and C, 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, which considers a sense of effectiveness on small classes. For subtask D, the averaged absolute difference (i.e., $avgAbsDiff = \frac{1}{n}\sum_{i=1}^{n}|x_i - \bar{x}|$) is employed, which is a common measure of how much a set of observations differ from the average. Since the subtask E aims at predicting the sentiment score for target term, in order to make the comparison of predicted strength of different terms reasonable, the Kendall rank correlation coefficient (usually measures the association between two measured quantities) and Spearman rank correlation (a nonparametric measure of statistical dependence between two variables) are adopted in this subtask, where the Kendall rank correlation coefficient is the official evaluation criteria.

3 Experiments

3.1 Datasets

The organizers provided tweet ids and a script for all participants to collect data. Table 1 shows the statistics of the data sets we used in our experiments.

For subtask A and B, the training data set is composed of SemEval 2013 Task 2 training and development data (Nakov et al., 2013) and the development

[8]The 29 negations and other following manually collected data are available upon request.

data set is made up of the test sets from the same tasks in previous two years. For subtask C and D, this data is divided into many topic sets.

With regard to subtask E, the organizers provided 200 terms labeled with a decimal in the range of 0 to 1. We observed that among these 200 given terms, 22% are hashtags and 15% contain negator. In consideration of the lack of training data, we expanded it with 1,346 terms collected from following sources: 916 terms which are present in all above mentioned 7 sentiment lexicons, 230 terms with hashtag and 200 terms with negator extracted from *NRC Hashtag sentiment lexicon* randomly. The provided 200 terms were used as development data. To predict the strength values of the extended data, we used the MPQA sentiment lexicon label as reference. There are 6 polarity types in MPQA, i.e., strong positive, weak positive, both strong, both weak, weak negative and strong negative. We converted them to numeric score as $1, 0.75, 0.5, 0.5, 0.25, 0$ respectively. By doing so, if a target term is present in this expanded lexicon, the output is its corresponding score. Otherwise we split the term to several words and calculated their averaged sentiment score as output.

dataset	Positive	Negative	Neutral	Total
subtask A:				
train	5,738(62%)	3,097(33%)	456(5%)	9,291
dev	10,159(58%)	6,416(37%)	875(5%)	17,450
test				
LiveJournal	660(50%)	511(39%)	144(11%)	1,315
SMS2013	1,071(46%)	1,104(47%)	159(7%)	2,334
Twitter2013	2,734(62%)	1,541(35%)	160(3%)	4,435
Twitter2014	1,807(73%)	578(23%)	88(4%)	2,473
Twitter2014S	82(66%)	37(30%)	5(4%)	124
offical2015	1,896(61%)	1,006(33%)	190(6%)	3,092
all	8,250(60%)	4,777(35%)	746(6%)	13,773
subtask B:				
train	3,774(37%)	1,598(16%)	4,842(47%)	10,214
dev	5,570(37%)	2,536(17%)	6,788(46%)	14,894
test				
LiveJournal	427(37%)	304(27%)	411(36%)	1,142
SMS2013	492(24%)	394(19%)	1,207(57%)	2,093
Twitter2013	1,572(41%)	601(16%)	1,640(43%)	3,813
Twitter2014	982(53%)	202(11%)	669(36%)	1,853
Twitter2014S	33(38%)	40(47%)	13(15%)	86
offical2015	1,038(43%)	365(15%)	987(41%)	2,390
all	5,411(39%)	2,166(16%)	6,183(45%)	13,760
subtask C and D:				
train	142(29%)	56(11%)	288(59%)	489
dev	65(35%)	34(18%)	85(46.%)	184
test	867(36%)	260(11%)	1256(53%)	2383

Table 1: Statistics of data sets in training (train), development (dev), test (test) set for subtask A, B, C and D. Twitter2014S stands for Twitter2014Sarcasm.

3.2 Experiments on Training Data

3.2.1 Subtask A and B

To address subtask A and B, we conducted a series of experiments to examine the effects of different traditional features. Table 2 describes the experiments of various traditional features on subtask A and B. From Table 2, it is interesting to find that: (1) SentiLexi and unigram are the most effective feature types to detect the polarities; (2) POS feature makes contribution to improve the performance for subtask B but no improvement for A. It may be because the neutral instances in subtask B (i.e., 45.58%) are much more than that in subtask A (i.e., 5.01%); (3) The emoticons features are not as effective as expected since most emoticons are already present in unigram.

Besides, following (Kim, 2014) we adopted sentence modeling and extracted the penultimate hidden layer content as novel word embedding feature to build another classifier. Furthermore, we combined the intermediate results (i.e., the distances between point to multiple hyperplanes returned from SVM) of two classifiers. The experimental results of using word embedding features in isolation and in combination are shown in Table 3. From Table 3, we find that the word embedding alone performs a bit worse than the traditional features. This may be because the traditional features are dozens of times more than word embedding features and as a result the effectiveness of word embeddings is impaired. However, when we combined the two experimental results, we find that the combination result of two classifiers achieves the best performances in both subtasks. This indicates that although the size of word embeddings is small, it still makes contribution to performance improvement.

Features	Subtask A	Subtask B
Traditional	86.65%	66.81%
Word embedding	83.80%	64.85%
Combination	87.68%	67.80%

Table 3: Results of subtask A and B using traditional features, word embedding features and their combination in terms of F_{macro} on training data.

Besides, in our preliminary experiments, we examined several supervised machine learning classification algorithms with different parameters imple-

Features	Subtask A	Features	Subtask B	Features	Subtask C
SentiLexi	81.83	SentiLexi	60.99	unigram	32.87
.+unigram	85.32(+3.49)	.+unigram	64.60(+3.61)	.+PAHE	33.51(+0.64)
.+Negation	86.20(+0.88)	.+pairgram	65.76(+1.16)	.+SentiLexi	34.37(+0.86)
.+pairgram	86.52(+0.32)	.+POS	66.19(+0.43)	.+POS	35.45(+1.03)
.+PAHE	86.57(+0.05)	.+Negation	66.68(+0.49)	.+Emoticon	**36.03**(+0.58)
.+bigram	**86.65**(+0.08)	.+PAHE	**66.81**(+0.13)	.+Negation	34.94(-1.09)
.+POS	86.53(-0.12)	.+Emoticon	66.76(-0.05)	-	-
.+Emoticon	86.50(-0.03)	.+bigram	66.21(-0.55)	-	-

Table 2: Results of feature selection experiments for subtask A, B and C in terms of F_{macro} on the training data. The numbers in the brackets are the performance increments compared with the previous results. *PAHE* stands for Punctuation&All-caps&Hashtag&Enlongated features. ".+" means to add current feature to the previous feature set.

mented in *scikit-learn* tools (Pedregosa et al., 2011) (e.g., SVM with *kernel*={*linear, rbf*}, *c=0.1 ,1 ,10*, SGD with *loss*={*hinge, log*}, RandomForestClassifier with *n*={*10, 50, 100*}, etc). Table 4 shows the configuration of classifiers with best performance. Thus, in subsequential experiments, we adopted the configurations listed in Table 4.

Task	Features	Configuration
Subtask A	traditional	SVM, kernel=linear,c=0.1
	word embedding	SVM, kernel=rbf,c=0.1
Subtask B	traditional	SVM, kernel=linear,c=0.1
	word embedding	SVM, kernel=rbf,c=0.1

Table 4: System configurations for subtask A and B.

3.2.2 Subtask C and D

Table 2 lists the experimental results using several traditional features on subtask C. Since the sentiment trend of given topic in subtask D is calculated from the results of subtask C (i.e., $sentiment\ trend = positive/(positive + negative)$), we have not conducted additional experiments for subtask D.

Similar with the first two subtasks, we adopted the SVM classification algorithm with *kernel=linear, c=0.1* as system configurations for follow-up experiments.

3.2.3 Subtask E

We transformed the informal terms to their normal forms and used the sentiment lexicons mentioned in Section 2.2 except *MPQA* to extract sentiment lexicon feature. If the target term contained more than one word, we averaged their scores as its final sentiment lexicon feature. Besides, the word embedding features were also adopted in this subtask.

To explore the effectiveness of different feature types, we conducted several feature combination experiments shown in Table 5.

Features	Kendall Rank	Spearman Rank
SentiLexi	48.24%	66.17%
Word embedding	52.97%	70.90%
SentiLexi + Word embedding	56.73%	75.56%

Table 5: Results of feature section experiments for subtask E on training data.

From Table 5, we find that: (1) The combination of SentiLexi and word embedding is the most effective feature type for sentiment score prediction; (2) The word embedding features achieved better result than SentiLexi features about 4.7% improvement in terms of Kendall measure, which indicates that word embedding feature preserves the sentiment information and semantic relationship between words.

We also performed a series of experiments to optimize the parameters of SVM classifiers. Similarly, we found that SVM classifier with *kernel=linear* and *c=1* obtained the best performance. Thus, in following experiments on test data, we adopted this configuration with SentiLexi and word embedding features together.

3.3 Results on Test Data

Using the optimum feature set and configurations described in Section 3.2, we trained separate models for each subtasks and evaluated them against the SemEval-2015 Task 10 test set.

Table 6 shows the results of our systems and the top-ranked systems on subtask A, B, C and D. From

Subtask	Systems	LiveJournal	SMS2013	Twitter2013	Twitter2014	Twitter2014S	Offical2015	Twitter2015
A	ECNU	82.49(6)	84.70(4)	85.28(4)	82.09(7)	70.96(7)	81.08(7)	-
	unitn	84.46(2)	88.60(2)	90.10(1)	87.12(1)	73.65(3)	84.79(1)	-
	KLUEless	83.94(4)	88.62(2)	88.56(2)	84.99(3)	75.59(3)	84.51(2)	-
B	ECNU	74.40(3)	68.49(1)	65.25(22)	66.37(20)	45.87(24)	59.72(19)	-
	Webis	71.64(14)	63.92(14)	68.49(10)	70.86(6)	49.33(11)	64.84(1)	-
	unitn	72.48(12)	68.37(2)	72.79(3)	73.60(2)	55.44(4)	64.59(2)	-
C/D	ECNU	-	-	-	-	-	-	25.38(5)/0.300(5)
	TwitterHawk	-	-	-	-	-	-	50.51(1)/0.214(3)
	KLUEless	-	-	-	-	-	-	45.48(2)/0.202(1)

Table 6: Performances of our systems and top-ranked systems for subtask A, B, C ($F_{macro}(\%)$) and D (*avgAbsDiff*) on test data. The numbers in the brackets are the rankings on corresponding data set.

the Table 6, we observe the following findings.

Firstly, in accordance with previous work (Rosenthal et al., 2014), the results of subtask B is much worse than those of subtask A. On one hand, the text in message-level task is long and contains multiple/mixed sentiments with different strength and the text in expression-level usually contain a single sentiment orientation. On the other hand, the polarity distributions of subtask A and B are significantly different (i.e., about 6.14% instances in expression-level are neutral while 41.30% in message-level).

Secondly, the performances on LiveJournal and SMS are comparable to the results on Twitter2013 and Twitter2014 in both subtasks, which means the Twitter, SMS and LiveJournal have similar characteristics and then we may consider to use SMS as training data when the available tweet data is insufficient.

Thirdly, the submissions of subtask C and D only adopted traditional linguistic features rather than the combination of word embeddings, which may result in the poor performance in subtask C and D.

Our systems ranked $7th$ out of 11 submissions for subtask A, $19th$ out of 40 submissions for subtask B and performed well on LiveJournal and SMS2013 data sets. For subtask C and D, our systems ranked $5th$ out of 7 submissions and $5th$ out of 6 submissions respectively.

Team ID	Kendall Rank	Spearman Rank
ECNU	59.07%(3)	78.61%(3)
INESC-ID	62.51%(1)	81.72%(2)
lsislif	62.11%(2)	82.02%(1)

Table 7: Performances of our systems and the top-ranked systems for subtask E. The numbers in the brackets are the official ranking.

Table 7 shows the results of our system and the top ranked system provided by organizer for subtask E. Our system ranked 3rd out of 10 submissions. Although the word embedding features obtained from large amount of contexts are believed to contain semantic information, they contain sentiment information more or less induced from context. As a consequence, with the aid of sentiment lexicon and word embedding, our system is promising.

4 Conclusion

In this paper, we combined the results of two classifiers (adopting traditional features and word embedding features respectively) to detect the sentiment polarity towards expression-level and message-level (i.e., subtask A, B), adopted several basic feature types to settle topic-level task (i.e., subtask C, D) and built regression model with the aid of sentiment lexicon features and word embedding features to predict degree of polarity strength on term-level (i.e., subtask E). Using word embedding features alone may not perform good results, but it makes contribution to performance improvement in combination with traditional linguistic features. In future work, we consider to construct the word representations bearing sentiment information to address sentiment analysis.

5 Acknowledgements

This research is supported by grants from Science and Technology Commission of Shanghai Municipality under research grant no. (14DZ2260800 and 15ZR1410700) and Shanghai Collaborative Innovation Center of Trustworthy Software for Internet of Things (ZF1213).

References

Lu Chen, Wenbo Wang, Meenakshi Nagarajan, Shaojun Wang, and Amit P Sheth. 2012. Extracting diverse sentiment expressions with target-dependent polarity from Twitter. In *ICWSM*.

Cícero Nogueira dos Santos and Maíra Gatti. 2014. Deep convolutional neural networks for sentiment analysis of short texts. In *Proceedings of the 25th International Conference on Computational Linguistics (COLING), Dublin, Ireland*.

Stefan Evert, Thomas Proisl, Paul Greiner, and Besim Kabashi. 2014. SentiKLUE: Updating a polarity classifier in 48 hours. *SemEval 2014*, page 551.

Long Jiang, Mo Yu, Ming Zhou, Xiaohua Liu, and Tiejun Zhao. 2011. Target-dependent Twitter sentiment classification. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 151–160.

Soo-Min Kim and Eduard Hovy. 2004. Determining the sentiment of opinions. In *Proceedings of the 20th international conference on Computational Linguistics*, page 1367.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.

Svetlana Kiritchenko, Xiaodan Zhu, Colin Cherry, and Saif Mohammad. 2014. NRC-Canada-2014: Detecting aspects and sentiment in customer reviews. pages 437–442, August.

Efthymios Kouloumpis, Theresa Wilson, and Johanna Moore. 2011. Twitter sentiment analysis: The good the bad and the omg! *ICWSM*, 11:538–541.

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 *Advances in Neural Information Processing Systems*, pages 3111–3119.

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.

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.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in Python. *The Journal of Machine Learning Research*, 12:2825–2830.

Sara Rosenthal, Preslav Nakov, Alan Ritter, and Veselin Stoyanov. 2014. Semeval-2014 task 9: Sentiment analysis in Twitter. *Proc. SemEval*, pages 73–80.

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, Denver, Colorado, June.

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.

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.

Hong Yu and Vasileios Hatzivassiloglou. 2003. Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences. In *Proceedings of the 2003 conference on Empirical methods in natural language processing*, pages 129–136.

Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, and Jun Zhao. 2014. Relation classification via convolutional deep neural network. In *Proceedings of COLING*, pages 2335–2344.

Jiang Zhao, Man Lan, and Tian Tian Zhu. 2014. ECNU: Expression-and message-level sentiment orientation classification in Twitter using multiple effective features. *SemEval 2014*, page 259.