# ECNU at SemEval-2016 Task 6: Relevant or Not? Supportive or Not? A Two-step Learning System for Automatic Detecting Stance in Tweets

**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
51131201039@ecnu.cn, mlan@cs.ecnu.edu.cn*

## Abstract

This paper describes our submissions to Task 6, i.e., Detecting Stance in Tweets, in SemEval 2016, which aims at detecting the stance of tweets towards given target. There are three stance labels: *Favor* (directly or indirectly by supporting given target), *Against* (directly or indirectly by opposing or criticizing given target), and *None* (none of the above). To address this task, we present a two-step learning system, which performs two steps, i.e., relevance detection and orientation detection, in a pipeline-based processing procedure. Our system ranked the 5th among 19 teams.

## 1 Introduction

Social platforms, such as *Twitter*, *Facebook*, etc., have attracted hundreds of millions of people to share and express their opinions or standpoints in the past few years. Promoted by that growth, researchers have been enthusiastic about mining useful information in these abundant free texts from social platform, such as stance detection. Determining the stance expressed in a post written for certain target is a relatively new task in sentiment analysis. Classifying stance involves identifying the target of the post and determining its sentiment orientation. The general researches just focus on detecting the stance of posts where the provided posts are relevant to the given target (Somasundaran et al., 2007; Somasundaran and Wiebe, 2010). Besides, the previous work usually aims at the posts collected from forums which have co-posts as reference (Murakami et al., 2007; Agrawal et al., 2003). Some approaches were adopted to settle stance detection, for example, Murakami and Agrawal detected the stance in the posts collected from forums and adopted co-posts as reference (Murakami et al., 2007; Agrawal et al., 2003).

The task of Detecting Stance in Tweets (DST) in SemEval 2016 aims at classifying the provided tweets into three stance classes, i.e., *Favor* (directly or indirectly by supporting given target), *Against* (directly or indirectly by opposing or criticizing given target) and *None* (none of the above) refer to a given target. The DST task consists of two subtasks which could be summarized as *supervised* subtask (i.e., subtask A) and *weakly supervised* subtask (i.e., subtask B). The supervised subtask is to test the stances of certain tweets towards five predefined targets with labeled training data, while the weak supervised subtask is to detect the stances of tweets towards one target with the aid of a mass of unlabeled training data.

Somasundaran showed that the stance classifier trained on unigram is a relatively strong baseline (Somasundaran and Wiebe, 2010). Based on Somasundaran and Wiebe's work, Anand augmented the *n*-gram features with several linguistic features (Anand et al., 2011). Except for feature engineering, many researchers focused on other methods to improve performance. For example, Murakami and Sridhar took the forward posts of current post into consideration (Murakami and Raymond, 2010; Sridhar et al., 2014). The previous works usually processed the posts with co-posts or some additional information, such as its author, writing timeline (Faulkner, 2014; Rajadesingan and Liu, 2014;

*Proceedings of SemEval-2016*, pages 451–457,
San Diego, California, June 16-17, 2016. ©2016 Association for Computational Linguistics

Hasan and Ng, 2013). Differ from these works, the DST focuses on classifying the stance of tweets into three classes, i.e., *Favor*, *Against* and *None*, rather than two classes. Moreover, the organizers did not provide the related information of tweet, such as author information. Thus, to address this task, we decomposed the stance detecting model into two parts, i.e., relevance detection and orientation detection, which aim at determining whether the tweet is relevant or irrelevant to the given target and whether the tweet is in favor of or against the given target. Since the given 6 targets belong to different types, e.g., *Hillary Clinton* and *Donald Trump* are about people, *Climate Change is a Real Concern* is a environmental topic, etc., considering the diversity of different targets, we built unique model for each target with different features. To achieve high performance, we proposed various features, e.g., *Linguistic Features*, *Topic Features*, *Word Vector Features*, *Similarity Features*, etc., to perform stance detection.

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

To address these two subtasks, i.e., supervised framework and weakly supervised framework, we used the two-step model to classify certain tweets into 3 stance labels (i.e., *Favor*, *Against*, *None*). The first step (i.e., relevance detection) is to determine whether the tweet is relevant to the given target. The second step (i.e., orientation detection) aims at classifying whether the tweet is support for the given target. To improve the classification performance, we extract various types of features, such as linguistic features (e.g., *N-grams*, *N-chars*), similarity features (e.g., *cosine similarity*, *JSD similarity*), topic features (e.g., *sent2topic*, *top topic word*), sentiment lexicon features, etc. The difference of the methods to settle two subtasks is located in the different training data they used. For subtask A, we segmented the training data into 5 subsets according to the 5 predefined targets and trained two classifiers for each subset. Thus, for subtask A, our system consists of 10 classifiers and we conducted feature selection procedure for each classifier. As for subtask B, we combined all labeled data in subtask A as training data and constructed two classifiers to perform relevance detection and orientation detection.

### 2.1 Data Preprocessing

Due to the irregular writing form of tweets, we first convert the slangs or abbreviations to their formal forms with the aid of a pre-defined dictionary downloaded from Internet[1]. For example, we convert *"gooooood"* into *"good"*, *"gr8"* to *"great"*. The processed data is fed into *CMU TweetNLP tool* (Owoputi et al., 2013) to perform tokenization, POS tagging. Meanwhile, we employ *Stanford Parser tool* (Klein and Manning, 2003) and *LDA-C* (Blei et al., 2003) to implement dependency parsing and topic parsing respectively. Finally, the *NLTK tool* (Bird et al., 2009) is used to conduct lemmatization and stemming.

### 2.2 Feature Engineering

Since we decompose the stance detecting task into two steps, i.e., relevance detection and orientation detection, this task is related to similarity evaluation and stance orientation classification. Thus, we extract five types of features, i.e., *Traditional Linguistic Features*, *Similarity Features*, *Topic Features*, *Sentiment Lexicon Features*, *Tweet Specific Features* and *Word Vector Features*.

#### 2.2.1 Traditional Linguistic Feature

*N-grams:* *N-grams* features are widely used in many NLP tasks. In this task, we extract *unigram*, *bigram*, *trigram* and *4-gram*.

*N-chars:* We record presence or absence of contiguous sequences of 3, 4, and 5 characters as *N-chars* features, i.e., *3-char*, *4-char* and *5-char*.

*Pos*: There are total 23 types of pos tags collected in training data processed by *CMU TweetNLP tool*. We record the number of each pos tags as *Pos* features.

*Cluster:* The *CMU TweetParser tool* provides the token clusters produced with the Brown clustering algorithm on 56 million English language tweets. The 1,000 clusters are served as *Cluster* features.

[1] This dictionary and the following Internet resources are available at https://github.com/haierlord/resource.git

*Dependency*: The dependency tree is generated by *Stanford Parser tool* and each tweet is represented as several triple (i.e., *relation(government, dependent)*). We extract three types of *Dependency* features: *relation-government* (*Rel-Gov*), *government-dependent* (*Gov-Dep*), *relation-government-dependent* (*Rel-Gov-Dep*). The feature value is set as 1 or 0 if corresponding tuple is present or absent in tweet.

*Top Tfidf Word* (*TopTfidf*): We divide the labeled tweet datasets with *Favor* or *Against* stances into 5 subsets towards given targets and calculate the *tfidf* score for each word in 5 subsets separately. We collect the 20 words with top $tfidf$ scores in each subset and set binary feature value to indicate whether the corresponding word exists in current tweet as *TopTfidf* feature.

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

*Negation:* We collect 29 negations from Internet and designed binary feature to record if there is negation in tweet.

### 2.2.2 Topic Feature

We feed all training data into *LDA-C tool* to produce some topic-related information.

*Sent2Topic*: The *LDA* could generate the document distribution among predefined topics. We extract this distribution as *sent2topic* feature.

*Word2Topic*: Each word in input tweets could be expressed as the probabilities among topics. The *word2topic* feature is represented as the accumulation of the probability of corresponding topic of all word in single tweet.

*Top Topic Word (TopTopic)*: Since the topic probability of each word indicates the significance for corresponding topic, thus we collect the top 20 words in each topic to build *TopTopic* feature.

### 2.2.3 Similarity Feature

Since we first determine whether the tweet is relevant to the given target, some similarity features are extracted to model the relevance detect classifier.

*JSD Similarity (JSD):* For each target, we collect the *Favor* and *Against* tweets to construct word distribution for different targets. Specifically, for subtask B, the tweets with *"#DonaldTrump"* are regarded as relevant record. For each tweet, we calculate *Jensen-Shannon Divergence (JSD) similarity* of word distributions between current tweet and corresponding target, which denotes as follows:

$$\begin{aligned} JSD(S,T) &= \frac{1}{2}KL(P_S||Q) + \frac{1}{2}KL(P_T||Q) \\ Q\left(w\right) &= \frac{1}{2}\left(P_S\left(w\right) + P_T\left(w\right)\right) \\ KL\left(P \parallel Q\right) &= \sum_{x\in X} P\left(x\right) log \frac{P\left(x\right)}{Q\left(x\right)} \end{aligned} \tag{1}$$

where $KL(P||Q)$ means the Kullback-Leibler divergence between distribution $P$ and $Q$. Furthermore, we also take the distributions under lemmatization and stemming forms into consideration.

*Cosine Similarity (Cosine):* Similar with *JSD*, we obtain the word distributions among targets current tweet respectively. The cosine distance among two distributions is calculated as *Cosine Similarity* feature. For *Cosine Similarity*, we take lemmatization and stemming forms into account as well.

*Overlap Similarity (Overlap): Overlap Similarity* is a simple and effective similarity measure and calculated as follows:

$$Overlap\ Similarity = \frac{|A \cap B|}{|A|} \tag{2}$$

where where $|A \cap B|$ denotes the size of intersection of set $A$ and set $B$ and $|A|$ means the size of set $A$. Here, we treat the top $5/10/20$ most relevant words of corresponding target produced by the *LDA tool* as $|A|$ and the current tweet as $|B|$. Similar with *JSD* and *Cosine Similarity*, we also consider the lemmatization and stemming forms. Thus, final dimension of *Overlap Similarity* is 9.

*ContainTopic:* It indicates whether there is any intersection between target words and tweet.

### 2.2.4 Sentiment Lexicon Feature

We employ the following seven sentiment lexicons to extract sentimental lexicon (*SentiLexi*) features: *Bing Liu lexicon*[2], *General Inquirer lexicon*[3], *IMDB*[4], *MPQA*[5], *AFINN*[6], *NRC Hashtag Sentiment*

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

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

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

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

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

*Lexicon*[7], and *NRC Sentiment140 Lexicon*[8]. Generally, we transform 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 a tweet, we first convert it to lowercase. Then for each sentiment lexicon, we calculate the following five sentimental scores: (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. If one word does not exist in one sentiment lexicon, its corresponding score is set to zero.

#### 2.2.5 Tweet Specific Feature

*AllCaps:* It represents the number of words with uppercase letters.

*Hashtag-Ngrams:* The hashtag always carries significant information. Thus, we segment the hashtag to normal phrase and construct *Hash-unigram* and *Hash-bigram*.

*Elongated:* It indicates the number of words with one character repeated more than two times in raw tweet, e.g., *"goooood"*.

*Emoticon:* We collect 69 emoticons from Internet and this binary feature records whether the corresponding emoticon is present in tweet.

#### 2.2.6 Word Vector Feature

Word vector is a continuous-values representation of the word which usually carries important information. In this part, we utilize two types of word vector, i.e., general word vector, sentiment word vector.

*General Word Vector (GoogleW2V):* We used the publicly available *word2vec* tool[9] to get word vectors with dimensionality of 300, which is trained on 100 billion words from Google News as *general word vector*.

*Sentiment Word Vector (SWV):* Zhang proposed the *Combined-Sentiment Word Embedding Model* to settle sentiment analysis task (Zhang and Lan, 2015). In this work, we continue to use this model to train the *sentiment word vector* with the aid of *NRC140 tweet corpus*(Go et al., 2009).

Since one tweet contains more than one word, we adopted simple *min*, *max*, *average* pooling strategies to obtain the text vector. Thus the final text vector $V(t)$ is concatenated by $V_{max}(t)$, $V_{min}(t)$ and $V_{average}(t)$.

### 2.3 Evaluation Metrics

For both subtask, we adopt the macro-averaged *F* score of *Favor* and *Against* stances (i.e., $F_{macro} = \frac{F_{Favor}+F_{Against}}{2}$) to evaluate the performance, which considers a sense of effectiveness on small classes. To estimate the system performance on training data, we employ $F_{macro} = \frac{F_{Relevant}+F_{Irrelevant}}{2}$ for step1 (i.e., relevance detection) and $F_{macro} = \frac{F_{Favor}+F_{Against}}{2}$ for step2 (i.e., orientation detection).

## 3 Experiments

### 3.1 Datasets

| Target | Favor | Against | None | Total |
|---|---|---|---|---|
| subtask A: | | | | |
| train: | | | | |
| Hillary | 118(17%) | 393(57%) | 178(26%) | 689 |
| Abortion | 121(19%) | 355(54%) | 177(27%) | 653 |
| Atheism | 92(18%) | 304(59%) | 117(23%) | 513 |
| Climate | 212(54%) | 15(4%) | 168(42%) | 395 |
| Feminist | 210(32%) | 328(49%) | 126(19%) | 664 |
| all | 753(26%) | 1,395(48%) | 766(26%) | 2,914 |
| test: | | | | |
| Atheism | 32(14%) | 160(73%) | 28(13%) | 220 |
| Abortion | 46(16%) | 189(68%) | 45(16%) | 280 |
| Hillary | 45(15%) | 172(58%) | 78(27%) | 295 |
| Climate | 123(73%) | 11(6%) | 35(21%) | 169 |
| Feminist | 58(20%) | 183(64%) | 44(16%) | 285 |
| all | 304(24%) | 230(57%) | 230(19%) | 1,249 |
| subtask B: | | | | |
| train: | | | | |
| Donald | | - | | 68,984 |
| test: | | | | |
| Donald | 148(21%) | 299(42%) | 260(37%) | 707 |

**Table 1:** Statistics of data sets in training data for subtask A and B. (*Hillary*, *Abortion*, *Climate*, *Feminist* and *Donald* refer to *"Hillary Clinton"*, *"Legalization of Abortion"*, *"Climate Change is a Real Concern" "Feminist Movement"* and *"Donald Trump"* respectively.)

For subtask A, the organizer supplied all training data, which consists of 5 targets, i.e., *"Hillary*

[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

*Clinton"*, *"Legalization of Abortion"*, *"Atheism"*, *"Climate Change is a Real Concern"* and *"Feminist Movement"*. The statistics of datasets are listed in Table 1. For subtask B, the participants just received tweet ids and a script to collect data and the provided tweets have no labels. The distribution of test data of both subtasks is established in Table 1 as well.

## 3.2 Experiments on Training Data

### 3.2.1 Subtask A

To address this subtask A, we adopted a two-step method, which aims at determining whether the tweet is relevant to the given target and whether the tweet is support for the given target, respectively. Furthermore, considering the diversity of different targets, we separated the training data into 5 subsets according their targets and trained 5 models to to settle subtask A. For each target, we built two classifiers to perform stance detection. Thus, there are altogether 10 classifiers constructed for subtask A. In order to improve the performance of stance detection, we conducted feature selection procedure for every classifier. The 5-fold cross validation was performed for system development.

Since the majority of features are high dimensional and sparse, e.g., the dimensions of *4-char* and *5-char* in *Feminist* are $15,536$ and $26,658$ respectively, in our preliminary experiments for all five targets, we employed the *Logistic Regression* algorithm implemented in *liblinear tools*[10], which has good generalization for sparse data.

Table 2 shows the results of feature selection experiments for subtask A. For each target, the two columns, i.e., *Rel* and *Ori*, list the optimal feature sets for relevance detection step and orientation detection step, respectively. As for feature selection strategy, we adopted *hill climbing*: keeping adding one type feature at a time until no further improvement can be achieved. Due to page limitation, we only listed optimal feature types for each corresponding target.

From Table 2, it is interesting to find:

(1) The *Similarity Features* are effective to detect the stance regardless of the targets. Since almost half of training tweets are labeled as *None* records which are not relevant to the given target, the *Similarity Features* are more adept at determining whether the tweet is related to the given target.

(2) The *Sentiment Lexicon* and *SWV* features are not quite effective as expected. Based on the observation on training data, we found that the *Favor* stance tweets not always directly express positive emotion and so are the *Against* tweets. For example, many tweeters usually support or against the target by commenting statements opposed to the given target rather than explicitly express their own opinions to the given target. Thus, it is hard to classify the tweet stance according to its sentiment polarity expressed in tweets alone. Furthermore, we also compared the output of feature selection procedure and the results using all features is much poorer than using optimal feature subsets (e.g., 57.81% vs 65.09% in *Hillary Clinton*), which shows that not all features are suitable for stance detection.

(3) The *Tweet Specific* features are beneficial to this task. It may be that the tweeters often use the emoticons (i.e., *Emoticon*), emphatic words (i.e., *Elongated*, *AllCaps*) to express their attitudes. Besides, the *hashtag* usually carries the main stance of the corresponding tweet.

We also performed preliminary experiments to tune parameters of classifier , e.g., the *penalty coefficient C*. Finally, the optimized configurations listed in Table 3 are adopted for subtask A test data.

### 3.2.2 Subtask B

As for subtask B, we did not construct extra system and just continued to use the method as subtask A. All labeled data of 5 targets in subtask A were used as training data for *Donald Trump* and two classifiers were built. The last two column in Table 2 shows the experiment results on training data for subtask B. The submitted system configuration is shown in Table 3.

| Subtask | Target | Configuration | |
|---|---|---|---|
| | | Step1 | Step2 |
| A | Hillary | LR, c=1 | LR, c=2 |
| | Abortion | LR, c=1 | LR, c=2 |
| | Atheism | LR, c=1 | LR, c=0.5 |
| | Climate | LR, c=1 | LR, c=1 |
| | Feminist | LR, c=5 | LR, c=0.5 |
| B | Donald | LR, c=1 | LR, c=1 |

**Table 3:** System configurations for subtask A and B.

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

| Feature | | Subtask A | | | | | | | | | | Subtask B | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Hillary | | Abortion | | Atheism | | Climate | | Feminist | | All | |
| | | Rel | Ori | Rel | Ori | Rel | Ori | Rel | Ori | Rel | Ori | Rel | Ori |
| Linguistic | unigram | | | | | | √ | | | | | | |
| | bigram | | | | | √ | | √ | | √ | | | |
| | trigram | √ | √ | | | √ | | √ | | √ | √ | | |
| | 4-gram | | | | | √ | | √ | | √ | | | |
| | 3-char | | | | √ | | | √ | | √ | √ | | |
| | 4-char | | | | √ | | | | | | √ | | |
| | 5-char | | | | | | √ | | | | √ | | |
| | Pos | | | | | | | | | √ | | | √ |
| | Negation | √ | | √ | | | | √ | | √ | √ | √ | |
| | Cluster | | | | | | | √ | | | √ | | |
| | Rel-Gov | | | | | | √ | √ | √ | | √ | | |
| | Gov-Dep | √ | | √ | | √ | | √ | √ | √ | √ | | √ |
| | Rel-Gov-Dep | | | | | | | √ | | √ | | | √ |
| | TopTfidf | √ | | √ | | √ | | √ | | | √ | √ | √ |
| | Punctuation | | | | | √ | | √ | √ | √ | √ | √ | √ |
| Topic | Sent2Topic | | | | √ | | | √ | | √ | √ | | |
| | Word2Topic | | | | | | | | | | √ | | √ |
| | TopTopic | | | √ | √ | √ | √ | √ | √ | √ | | | √ |
| Similarity | JSD | √ | | √ | | √ | √ | √ | √ | | | √ | √ |
| | Cosine | | | √ | √ | √ | √ | √ | √ | | √ | √ | |
| | Overlap | √ | √ | | | √ | √ | √ | | √ | | √ | √ |
| | ContainTopic | √ | | √ | | | | √ | | √ | | √ | √ |
| Sentiment Lexicon | SentiLexi | | | | | | | | | | | | |
| Tweet | AllCaps | | √ | | √ | √ | | √ | | √ | √ | √ | |
| | Hashtag-unigram | | | | | √ | | √ | | √ | √ | | √ |
| | Hashtag-bigram | | | √ | | | | √ | | √ | √ | √ | |
| | Elongated | | | √ | | √ | | √ | | √ | √ | | √ |
| | Emoticon | | | √ | √ | √ | | √ | √ | √ | √ | | |
| Word Vector | GoogleW2V | | √ | | √ | √ | √ | √ | | | √ | | √ |
| | SWV | | | | | | | | √ | | | | |
| Steps Results ($F_{macro}\%$) | | 81.59 | 72.11 | 82.51 | 77.57 | 83.69 | 76.78 | 85.78 | 62.96 | 73.39 | 63.94 | 80.07 | 68.65 |
| Final Results ($F_{macro}\%$) | | 65.09 | | 71.42 | | 73.75 | | 56.22 | | 61.58 | | 63.34 | |

**Table 2:** Results of feature selection experiments for subtask A and subtask B. *Rel* and *Ori* stand for *relevance detection step* and *orientation detection step* respectively.

### 3.3 Results and Discussion

Using the optimum feature sets shown in Table 2 and configurations shown in Table 3, we constructed 10 classifiers for subtask A and 2 classifiers for subtask B and assessed them against the SemEval 2016 Task 6 test data. Table 4 lists the results of our systems and the top-ranked systems on test data provided by organizer for subtask A and B. In subtask A, our system ranked 5th out of 19 teams and in subtask B, the ranking is 5th/9.

| Subtask | TeamID | Target | $F_{macro}(\%)$ |
|---|---|---|---|
| A | ECNU(5) | Hillary | 57.84 |
| | | Abortion | 61.25 |
| | | Atheism | 61.96 |
| | | Climate | 41.32 |
| | | Feminist | 56.20 |
| | | All | 65.55 |
| | MITRE(1) | All | 67.82 |
| | pkudblab(2) | All | 67.33 |
| B | ECNU(5) | Donald | 34.08 |
| | pkudblab(1) | Donald | 56.28 |
| | LitisMind(2) | Donald | 44.66 |

**Table 4:** Performances of our systems and the top-ranked system for subtask A and B. *All* stands for the overall result for subtask A.

First, the results in Table 4 showed that our two-step system performed comparable to the best result in supervised framework (subtask A). It indicates that the proposed system and features are adept in detecting stance in tweet. However, compared with the results on training data, the results on test data are much poorer. The possible reason may be the difference between training data and test data. A further deep analysis will be done later.

Second, in subtask B, our system performed worse than the top results. The major reason lies in that we only adopted the same configuration tuned on training data and did not expanded the training data by adding unlabeled data.

## 4 Conclusion

In this paper, we decomposed the stance detection task into two steps, i.e., relevance detection and orientation detection, which aim at determining whether the tweet is relevant to the given target and whether the tweet is support for the given target. Considering the diversity of different targets, we built unique model for each target. Several types of features are proposed, for example, *Similarity Features*, *Linguistic Features*, *Topic Features*, etc. The experimental results on training and test data show that the proposed systems is suitable for stance detection. In future work, we consider to optimize the feature engineering to avoid overfitting.

## Acknowledgments

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

## References

Rakesh Agrawal, Sridhar Rajagopalan, Ramakrishnan Srikant, and Yirong Xu. 2003. Mining newsgroups using networks arising from social behavior. In *Proceedings of the 12th international conference on World Wide Web*, pages 529–535. ACM.

Pranav Anand, Marilyn Walker, Rob Abbott, Jean E Fox Tree, Robeson Bowmani, and Michael Minor. 2011. Cats rule and dogs drool!: Classifying stance in online debate. In *Proceedings of the 2nd workshop on computational approaches to subjectivity and sentiment analysis*, pages 1–9. Association for Computational Linguistics.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python*. " O'Reilly Media, Inc.".

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022.

Adam Faulkner. 2014. Automated classification of stance in student essays: An approach using stance target information and the wikipedia link-based measure. In *The Twenty-Seventh International Flairs Conference*.

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

Kazi Saidul Hasan and Vincent Ng. 2013. Stance classification of ideological debates: Data, models, features, and constraints. In *IJCNLP*, pages 1348–1356.

Dan Klein and Christopher D Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pages 423–430. Association for Computational Linguistics.

Akiko Murakami and Rudy Raymond. 2010. Support or oppose?: classifying positions in online debates from reply activities and opinion expressions. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pages 869–875. Association for Computational Linguistics.

Akiko Murakami, Tetsuya Nasukawa, Fusashi Nakamura, Hironori Takeuchi, Risa Nishiyama, Pnina Veisberg, and Hideo Watanabe. 2007. Innovation-jam: Analysis of online discussion records using text mining technology. In *International Workshop on Intercultural Collaboration 2007 (IWIC2007)*.

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.

Ashwin Rajadesingan and Huan Liu. 2014. Identifying users with opposing opinions in twitter debates. In *Social Computing, Behavioral-Cultural Modeling and Prediction*, pages 153–160. Springer.

Swapna Somasundaran and Janyce Wiebe. 2010. Recognizing stances in ideological on-line debates. In *Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*, pages 116–124. Association for Computational Linguistics.

Swapna Somasundaran, Josef Ruppenhofer, and Janyce Wiebe. 2007. Detecting arguing and sentiment in meetings. In *Proceedings of the SIGdial Workshop on Discourse and Dialogue*, volume 6.

Dhanya Sridhar, Lise Getoor, and Marilyn Walker. 2014. Collective stance classification of posts in online debate forums. *ACL 2014*, page 109.

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*.