Sentiment-Specific Representation Learning for Document-Level Sentiment Analysis

Duyu Tang

Research Center for Social Computing and Information Retrieval,
Harbin Institute of Technology, Harbin, China
dytang@ir.hit.edu.cn

ABSTRACT

In this paper, we propose a representation learning research framework for document-level sentiment analysis. Given a document as the input, document-level sentiment analysis aims to automatically classify its sentiment/opinion (such as *thumbs up* or *thumbs down*) based on the textural information. Despite the success of feature engineering in many previous studies, the hand-coded features do not well capture the semantics of texts. In this research, we argue that learning sentiment-specific semantic representations of documents is crucial for document-level sentiment analysis. We decompose the document semantics into four cascaded constitutes: (1) word representation, (2) sentence structure, (3) sentence composition and (4) document composition. Specifically, we learn sentiment-specific word representations, which simultaneously encode the contexts of words and the sentiment supervisions of texts into the continuous representation space. According to the principle of *compositionality*, we learn sentiment-specific sentence structures and sentence-level composition functions to produce the representation of each sentence based on the representations of the words it contains. The semantic representations of documents are obtained through document composition, which leverages the sentiment-sensitive discourse relations and sentence representations.

Categories and Subject Descriptors

I.2.7 [**ARTIFICIAL INTELLIGENCE**]: Natural Language Processing

Keywords

sentiment analysis; deep learning; natural language processing

1. INTRODUCTION

Sentiment analysis (also known as opinion mining [15, 9, 4]) that analyzes people's opinions/sentiments/emotions from texts is an active research field in natural language processing [10]. In this research, we target at the task of document-level sentiment analysis, which is a fundamental and most studied area in sentiment analysis. It aims to classify the sentiment/opinion (suc as *thumbs up* and *thumbs down*) of a document based on the text information.

Most previous studies follow Pang et al. [16] and regard document-level sentiment analysis as a special case of text categorization task. They typically employ machine learning algorithms, such as Supported Vector Machine, to build the sentiment classifier from the texts with accompanying sentiment labels in a supervised learning framework. Under this direction, most studies focus on designing effective features because the performance of a sentiment classifier is heavily dependent on the choice of feature representation of texts [3]. For example, Mohammad et al. [13] build the top-performed system in the Twitter sentiment classification track of SemEval 2013 [14] by using many sentiment lexicons and hand-crafted rules as features.

Despite the success of feature engineering, hand-coded features do not well capture the semantics of texts. Furthermore, it is desirable to discover the semantic representations of texts from the data and make the learning algorithms less dependent on laborious feature engineering. Recently, representation learning (or deep learning [1]) has been shown effective in many natural language processing tasks, such as word-segmentation, pos-tagging, named entity recognition and parsing, etc. However, we find that directly applying these fact-based approaches to document-level sentiment analysis is not effective enough. The reason lies in that they typically fail to capture the sentiment information of texts. Take word embedding[1] as an example, existing context-based learning algorithms [2, 11] only model the contexts of words. As a result, words with similar contexts but opposite polarity, such as *good* and *bad*, are mapped into neighboring vectors. It is meaningful for some tasks such as pos-tagging, but it becomes a disaster for sentiment analysis because they have the opposite sentiment polarity. We therefore focus on developing sentiment-specific representation learning methods for document-level sentiment analysis.

2. THE PROPOSED RESEARCH

In this research, we argue that learning sentiment-specific document semantic is vital for document-level sentiment analysis. According to the principle of *compositionality* that

[1]Word embedding is a dense, low-dimensional and real-valued vector for each word.

WSDM'15, February 2-6, 2015, Shanghai, China.

http://dx.doi.org/10.1145/2684822.2697035.

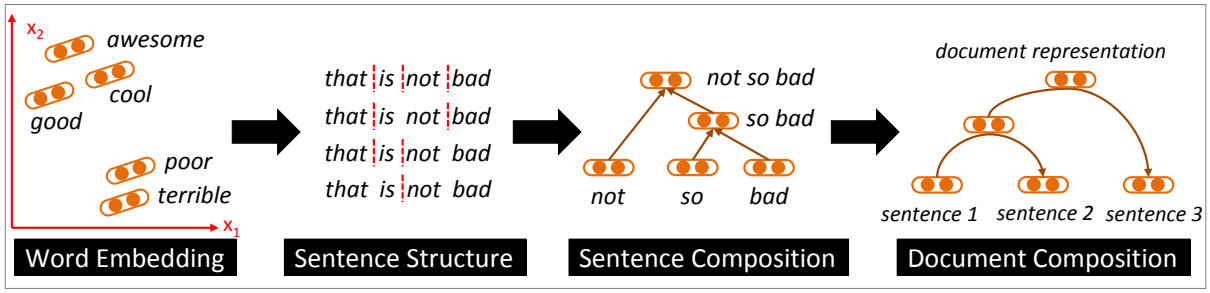

Figure 1: The sentiment-specific representation learning framework for document-level sentiment analysis.

the meaning of a longer expression (a sentence or a document) comes from the meanings of its words and the rules used to combine them [5], we decompose the document semantic into four cascaded constituents: (1) word embedding, (2) sentence structure, (3) sentence composition and (4) document composition. The work-flow of the proposed representation learning framework is illustrated in Figure 1. We learn word embeddings to capture the meanings of words, and then learn sentence structure and sentence composition to produce the representations of sentences based on the word embeddings. Afterwards, we calculate the document representations through document composition based on sentence representations and discourse analysis. We apply the learned document representations as features, and build the document-level sentiment classifier with existing machine learning methods such as SVM. We describe the research question of each constitute in this section.

2.1 Sentiment-Specific Word Embedding

Word embedding is the fundamental component of our representation learning framework because of the principle of *compositionality*. Word embedding is a low-dimensional, dense and real-valued vector for each word. After trained on a corpus, the words with similar grammatical usages and semantic meanings might be mapped into close vectors in the embedding space. Accordingly, the learned word embeddings can be easily utilized as features that capture semantic or clustering information of words for natural language processing tasks [2].

Existing embedding learning algorithms typically only capture the contexts of words but ignore the sentiment information of texts. As a result, words with similar contexts but opposite polarity, such as *good* and *bad*, are mapped into neighboring vectors. It is meaningful for some tasks such as pos-tagging, but it is problematic for sentiment analysis because they have the opposite sentiment polarity. To solve this problem, we propose to learn sentiment-specific word embedding (SSWE) that simultaneously encodes the contexts of words and sentiment information of texts in the continuous representation of words. Thus, the nearest neighbours of SSWE are also semantically similar while it favors words with the same sentiment polarity. We propose two methods based on existing embedding learning algorithms [2, 11] to learn SSWE, which are detailed in Section 3.1.

2.2 Sentiment-Specific Sentence Structure

Handling the complicated expressions delivering people's opinions is one of the most challenging problems in sentiment analysis. Traditional sentiment analysis algorithms typically simply employ bag-of-words to represent a sentence. As a result, they cannot handle the inconsistent sentiment polarity between a phrase and the words it contains, such as "*not bad*" and "*a great deal of*". Bag-of-ngram might cover high-order phrases, however, the use of it will dramatically increase the dimension of feature space.

We argue that learning the structure inherent in a sentence is important to help us understand its meaning. Traditional structure learning algorithms in fact-based NLP tasks (such as trunking and parsing) typically manually design linguistic-driven grammars or learn the patterns from the annotated treebank [10]. However, they cannot well handle the negation, intensification, and contrast phenomenons on the user-generated texts (e.g., reviews, tweets, etc.), which are the major focus of sentiment analysis algorithms. We therefore target at learning the sentiment-specific structures of sentences, which are optimized for sentiment analysis. Our sentiment-specific sentence segmentor [19] is described in Section 3.2.

2.3 Sentence Composition

Sentence representation is a pivot for document-level sentiment analysis because it links the word representations and document representations. Since regarding each sentence as a unique element makes the representation space to be extremely sparse, dominated previous studies investigate sentence composition methods [12] that calculate the representation of a sentence based on the representations of the words it contains. With the revival of interest in deep learning [1], neural network based methods including Recursive Neural Networks [17, 18] and Convolutional Neural Networks [7] have proven effective in sentence-level sentiment classification. However, to our knowledge, whether the compression based methods (such as autoencoder or restricted boltzmann machine) can learn meaningful sentence composition for sentiment analysis still remains unclear [6]. It is desirable to develop a sentiment-tailored composition approach that effectively handle the negation, intensification, and contrast phenomenons in sentiment analysis.

2.4 Document Composition

Learning meaningful and effective document representation is the major focus of the proposed representation learning framework. Given a document and the representation of each sentence it contains, we need to re-visit the principle of *compositionality* and calculate the document representation with document composition. We argue that sentiment-sensitive discourse relation is the key of document composition. A similar idea is given by Zhou et al. [22] that empir-

ically defines a discourse scheme with constraints on sentiment polarity based on Rhetorical Structure Theory (RST). Unlike their method, we want to learn these sentiment-sensitive discourse phenomenons from the data, and make the composition learning algorithm less dependent on feature engineering [8]. After that, the learned document representations will be considered as features for building the sentiment classifier with existing machine learning algorithms.

3. METHODOLOGY

We introduce our algorithms for learning sentiment-specific word embeddings and sentence structures in this part.

3.1 Sentiment-Specific Word Embedding

We extend two state-of-the-art neural network based methods [2, 11] tailored for learning word embeddings, and integrate the sentiment information of sentences (e.g. tweets) to learn the sentiment-specific word embedding (SSWE).

Our **first model** (SSWE$_u$ [21]) is an extension based on the C&W model [2], as illustrated in Figure 2). Collobert and Weston [2] introduce C&W model to learn word embedding based on the contexts of words. Given an ngram such as *"cat chills on a mat"*, C&W replaces the center word with a random word w^r and derives a **corrupted** ngram *"cat chills RANDOM a mat"*. The training objective is that the original ngram is expected to obtain a higher context score than the corrupted ngram by a margin of 1. The ranking objective can be optimized by a hinge loss,

$$loss_{cw}(t, t^r) = max(0; 1 - f^{cw}(t) + f^{cw}(t^r)) \quad (1)$$

where t is the original ngram, t^r is the corrupted ngram, $f^{cw}(\cdot)$ is a one-dimensional scalar representing the context score of the input ngram. During training, the context score $f^{cw}(\cdot)$ is obtained with a feed-forward neural network as shown in Figure 2.

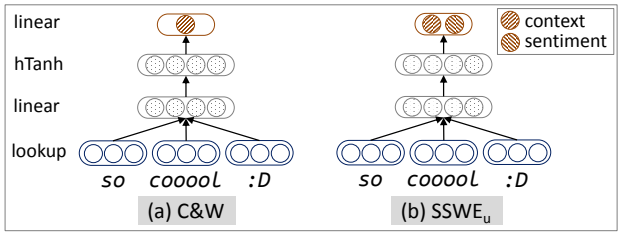

Figure 2: The traditional C&W model and our neural network (SSWE$_u$ [21]) for learning sentiment-specific word embedding.

Formally, $f^{cw}(\cdot)$ is calculated as given in Equation 2, where L is the lookup table of word embedding, w_1, w_2, b_1, b_2 are the parameters of linear layers. The original and corrupted ngrams are treated as inputs of the feed-forward neural network, respectively.

$$f^{cw}(t) = w_2 \cdot a + b_2 \quad (2)$$

$$a = hTanh(w_1 L_t + b_1) \quad (3)$$

We develop a neural network (**SSWE$_u$**) that captures the sentiment information of sentences as well as the contexts of words. Given an original (or corrupted) ngram and the sentiment polarity of a sentence as the input, SSWE$_u$ predicts a two-dimensional vector for each input ngram. The two scalars ($\boldsymbol{f}_0^u$, $\boldsymbol{f}_1^u$) stand for context score and sentiment score of the input ngram, respectively. The training objectives of SSWE$_u$ are that (1) the original ngram should obtain a higher context score $\boldsymbol{f}_0^u(t)$ than the corrupted ngram $\boldsymbol{f}_0^u(t^r)$, and (2) the sentiment score of original ngram $\boldsymbol{f}_1^u(t)$ should be more consistent with the gold polarity annotation of sentence than corrupted ngram $\boldsymbol{f}_1^u(t^r)$. The loss function of SSWE$_u$ is the linear combination of two hinge losses,

$$loss_u(t, t^r) = \alpha \cdot loss_{cw}(t, t^r) + (1 - \alpha) \cdot loss_{us}(t, t^r) \quad (4)$$

where $loss_{cw}(t, t^r)$ is the context loss as given in Equation 1, $loss_{us}(t, t^r)$ is the sentiment loss as described in Equation 5. The hyper-parameter α weighs the two parts.

$$loss_{us}(t, t^r) = max(0, 1 - \delta_s(t)\boldsymbol{f}_1^u(t) + \delta_s(t)\boldsymbol{f}_1^u(t^r)) \quad (5)$$

where $\delta_s(t)$ is an indicator function that reflects the gold sentiment polarity of a sentence,

$$\delta_s(t) = \begin{cases} 1 & \text{if } t \textit{ is positive} \\ -1 & \text{if } t \textit{ is negative} \end{cases} \quad (6)$$

Our **second model** (SSPE [20]) is an extension based on the context-based method (SkipGram) proposed by Mikolov et al. [11]. Given a word (or phrase) w_i as the input, SkipGram maps it into its embedding representation e_i, and utilizes e_i to predict the context words of w_i, namely w_{i-2}, w_{i-1}, w_{i+1}, w_{i+2}, et al. The training objective of SkipGram is to maximize the average log probability:

$$f_{context} = \frac{1}{T}\sum_{i=1}^{T}\sum_{-c \le j \le c, j \neq 0} log\ p(w_{i+j}|e_i) \quad (7)$$

where T is the occurrence of each phrase in the corpus, c is the window size, e_i is the embedding of the current word w_i, w_{i+j} is the context words of w_i.

We extend SkipGram and develop sentiment-specific phrase embedding (SSPE) to integrate the sentiment information of sentences, as illustrated in Figure 3. Given a triple $\langle w_i, s_j, pol_j \rangle$ as the input, where w_i is a word(or phrase) contained in the sentence s_j whose gold sentiment polarity is pol_j, our training objective is to (1) utilize the embedding of w_i to predict its context words, and (2) use the sentence representation se_j to predict the gold sentiment polarity of s_j, namely pol_j. We simply average the embedding of phrases contained in a sentence as its continuous representation se_j. The objective of the sentiment part is to maximize the average of log sentiment probability:

$$f_{sentiment} = \frac{1}{S}\sum_{j=1}^{S} log\ p(pol_j|se_j) \quad (8)$$

where S is the occurrence of each sentence in the corpus, $\sum_k pol_{jk} = 1$. For binary classification between positive and negative, the distribution of [0,1] is for positive and [1,0] is for negative. The final training objective is to maximize the linear combination of the context and sentiment parts:

$$f_{SSPE} = \alpha \cdot f_{context} + (1 - \alpha) \cdot f_{sentiment} \quad (9)$$

where α is a hyper-parameter that weights the two parts.

We collect massive tweets containing positive and negative emoticons to train the sentiment-specific word embeddings. We regard emoticon signals as the sentiment supervision of

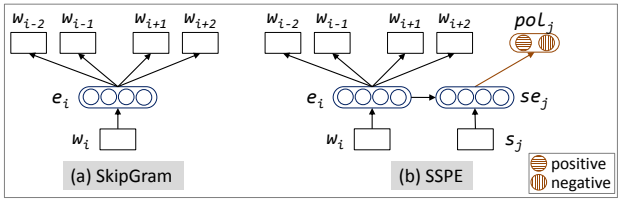

Figure 3: The traditional SkipGram model and our neural network (SSPE [20]) for learning sentiment-specific word embedding.

sentences during training. After using some heuristic filtering rules (detailed in [21, 20]), we collect 10M tweets, selected by 5M tweets with positive emoticons and 5M tweets with negative emoticons, as the training data.

3.2 Sentiment-Specific Sentence Structure

We describe our algorithm for learning sentiment-specific sentence segmentation [19] in this part. We develop a joint segmentation and classification framework, which simultaneously conducts sentence segmentation and sentence-level sentiment classification. The intuitions of the proposed joint model are two-folds:

• The segmentation results have a strong influence on the Sentiment classification performance, since they are the inputs of the sentiment classification model.

•The usefulness of a segmentation can be judged by whether the sentiment classifier can use it to predict the correct sentence polarity.

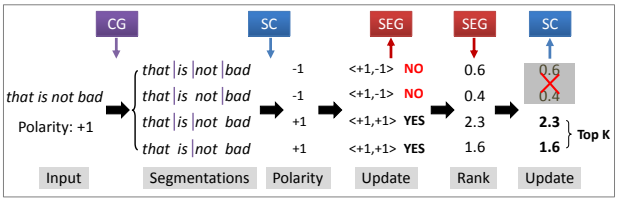

Figure 4: The proposed framework ([19]) for learning sentiment-specific sentence segmentation.

Specifically, we use a log-linear model to score each segmentation candidate, and exploit the phrasal information of top-ranked segmentations as features to build the sentiment classifier. A marginal log-likelihood objective function (as given in Equation 10) is devised for the segmentation model, which is optimized for enhancing the sentiment classification performance.

$$loss = -\sum_{i=1}^{|T|} log(\frac{\sum_{j \in H_i} \phi_{ij}}{\sum_{j' \in A_i} \phi_{ij'}}) + \lambda ||w||_2^2 \qquad (10)$$

where T is the training data; A_i represents all the segmentation candidates of sentence s_i; H_i means the hit candidates of s_i; λ is the weight of the L2-norm regularization factor (detailed in [19]).

4. EMPIRICAL RESULTS

We conducts experiments to evaluate the effectiveness of the proposed algorithms. In this part, we show the empirical results by applying sentiment-specific word embeddings and sentence structures for Twitter sentiment classification.

4.1 Sentiment-Specific Word Embedding

We conduct positive/negative Twitter sentiment classification on the benchmark dataset from SemEval 2013 [14]. In our method (SSWE$_u$), we directly apply the learned SSWE as features for building the sentiment classifier. We compare with the following baseline methods (more baselines are detailed in [21]). Results are given in Table 1.

(1) *DistSuper*: We use the 10 million tweets selected by positive and negative emoticons as training data, and build sentiment classifier with LibLinear based on bag-of-words.

(2) *SVM*: The bag-of-word features and SVM are widely used baseline methods to build sentiment classifiers [16]. We use LibLinear to train the classifier.

(3) *NRC*: NRC-Canada builds the top-performed system in SemEval 2013 Twitter sentiment classification track [13]. They use many lexicons and hand-crafted features.

Method	Macro-F1
DistSuper	61.74
SVM	74.50
NRC (**Top System** in SemEval 2013)	**84.73**
SSWE$_u$	**84.98**
SSWE$_u$+NRC	**86.58**

Table 1: Macro-F1 on positive/negative classification of tweets [21].

Our method (SSWE$_u$) achieves 84.98% by using only SSWE$_u$ as features without using any sentiment lexicons or hand-crafted rules. The results indicate that SSWE$_u$ automatically learns discriminative features from massive tweets and performs comparable with the state-of-the-art manually designed features. After concatenating SSWE$_u$ with the feature sets of *NRC*, the performance is further improved.

We also compare with other word embedding algorithms for Twitter sentiment classification. From Table 2, we find that our sentiment-specific word embeddings (SSWE$_h$, SSWE$_s$, SSWE$_u$) yield better performances.

Embedding	unigram	uni+bi	uni+bi+tri
C&W	74.89	75.24	75.89
Word2vec	73.21	75.07	76.31
ReEmb(C&W)	75.87	–	–
ReEmb(w2v)	75.21	–	–
WVSA	77.04	–	–
SSWE$_h$	81.33	83.16	83.37
SSWE$_r$	80.45	81.52	82.60
SSWE$_u$	**83.70**	**84.70**	**84.98**

Table 2: Macro-F1 on positive/negative classification of tweets with different word embeddings [21].

4.2 Sentiment-Specific Sentence Structure

We evaluate our sentiment-specific sentence segmentor by applying it for Twitter sentiment classification on the benchmark dataset from SemEval 2013. We compare the proposed model (JSC) with two pipelined methods. *Pipeline 1* use the bag-of-word segmentation. *Pipeline 2* use the segmentation

with maximum phrase number. The tick *[A, B]* on x-axis means the different features used for classification.

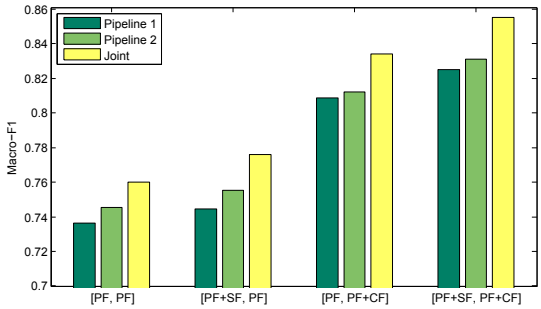

Figure 5: Macro-F1 for positive/negative classification of tweets with joint and pipelined models [19].

From Figure 5, we find that the proposed model consistently outperforms pipelined baseline methods in all feature settings. The reason lies in that our method uses the sentiment information to optimize the sentence segmentor, which in turns produces benefits to the sentiment classifier.

5. RESEARCH ISSUES FOR DISCUSSION

We briefly describe the potential research issues for discussion in this part. For **word embedding**, the issues are which kind of sentiment information (positive/negative or fine-grained sentiment; sentence-level or document-level) might be used, and how to incorporate them for learning better SSWEs. For **sentence structure**, the issue is whether we can manually design a hybrid grammar that integrates both linguistic and sentiment schemes. If not, how can we learn the grammar effectively with minor manually annotations. For **sentence composition**, the issue is how to develop reasonable and effective composition functions to effectively handle the negation, intensification, and contrast phenomenons. For **document composition**, how to carefully define or automatically learn the sentiment-sensitive discourse relations, and then leverage them for document composition are the issues to be solved.

Acknowledgments

This research was partly supported by National Natural Science Foundation of China (No.61133012, No.61273321, No.61300113).

6. REFERENCES

[1] Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. *IEEE Trans. PAMI*, 2013.

[2] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12:2493–2537, 2011.

[3] P. Domingos. A few useful things to know about machine learning. *Communications of the ACM*, 55(10):78–87, 2012.

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

[5] G. Frege. On sense and reference. *Ludlow (1997)*, pages 563–584, 1892.

[6] E. Grefenstette, K. M. Hermann, G. Dinu, and P. Blunsom. New directions in vector space models of meaning. In *Proceedings of the ACL: Tutorials*, 2014.

[7] N. Kalchbrenner, E. Grefenstette, and P. Blunsom. A convolutional neural network for modelling sentences. In *Proceedings of ACL*, pages 655–665, 2014.

[8] J. Li, R. Li, and E. Hovy. Recursive deep models for discourse parsing. In *Procedding of EMNLP*. 2014.

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

[10] C. Manning, P. Raghavan, and H. Schutze. *Introduction to information retrieval*, volume 1. Cambridge University Press Cambridge, 2008.

[11] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *ICLR*, 2013.

[12] J. Mitchell and M. Lapata. Composition in distributional models of semantics. *Cognitive Science*, 34(8):1388–1429, 2010.

[13] S. M. Mohammad, S. Kiritchenko, and X. Zhu. Nrc-canada: Building the state-of-the-art in sentiment analysis of tweets. *Proceedings of SemEval*, 2013.

[14] P. Nakov, S. Rosenthal, Z. Kozareva, V. Stoyanov, A. Ritter, and T. Wilson. Semeval-2013 task 2: Sentiment analysis in twitter. In *SemEval*, 2013.

[15] B. Pang and L. Lee. Opinion mining and sentiment analysis. *Foundations and trends in information retrieval*, 2(1-2):1–135, 2008.

[16] B. Pang, L. Lee, and S. Vaithyanathan. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the EMNLP*, 2002.

[17] R. Socher, J. Pennington, E. Huang, A. Ng, and C. Manning. Semi-supervised recursive autoencoders for predicting sentiment distributions. In *Proceeding of EMNLP*, pages 151–161, 2011.

[18] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Ng, and C. Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceeding of EMNLP*, 2013.

[19] D. Tang, F. Wei, B. Qin, L. Dong, T. Liu, and M. Zhou. A joint segmentation and classification framework for sentiment analysis. In *Procedding of EMNLP*. 2014.

[20] D. Tang, F. Wei, B. Qin, M. Zhou, and T. Liu. Building large-scale twitter-specific sentiment lexicon: A representation learning approach. In *Proceedings of COLING 2014*, pages 172–182, 2014.

[21] D. Tang, F. Wei, N. Yang, M. Zhou, T. Liu, and B. Qin. Learning sentiment-specific word embedding for twitter sentiment classification. In *Procedding of Annual Meeting of ACL.*, pages 1555–1565. 2014.

[22] L. Zhou, B. Li, W. Gao, Z. Wei, and K.-F. Wong. Unsupervised discovery of discourse relations for eliminating intra-sentence polarity ambiguities. In *Proceeding of EMNLP*, pages 162–171, 2011.