Building Large-Scale Twitter-Specific Sentiment Lexicon : A Representation Learning Approach

Duyu Tang[♮*], **Furu Wei**[‡], **Bing Qin**[♮†], **Ming Zhou**[‡], **Ting Liu**[♮]
[♮]Research Center for Social Computing and Information Retrieval,
Harbin Institute of Technology, China
[‡]Microsoft Research, Beijing, China
{dytang, qinb, tliu}@ir.hit.edu.cn
{fuwei, mingzhou}@microsoft.com

Abstract

In this paper, we propose to build large-scale sentiment lexicon from Twitter with a representation learning approach. We cast sentiment lexicon learning as a phrase-level sentiment classification task. The challenges are developing effective feature representation of phrases and obtaining training data with minor manual annotations for building the sentiment classifier. Specifically, we develop a dedicated neural architecture and integrate the sentiment information of text (e.g. sentences or tweets) into its hybrid loss function for learning sentiment-specific phrase embedding (**SSPE**). The neural network is trained from massive tweets collected with positive and negative emoticons, without any manual annotation. Furthermore, we introduce the Urban Dictionary to expand a small number of sentiment seeds to obtain more training data for building the phrase-level sentiment classifier. We evaluate our sentiment lexicon (**TS-Lex**) by applying it in a supervised learning framework for Twitter sentiment classification. Experiment results on the benchmark dataset of SemEval 2013 show that, TS-Lex yields better performance than previously introduced sentiment lexicons.

1 Introduction

A sentiment lexicon is a list of words and phrases, such as "*excellent*", "*awful*" and "*not bad*", each of which is assigned with a positive or negative score reflecting its sentiment polarity and strength. Sentiment lexicon is crucial for sentiment analysis (or opining mining) as it provides rich sentiment information and forms the foundation of many sentiment analysis systems (Pang and Lee, 2008; Liu, 2012; Feldman, 2013). Existing sentiment lexicon learning algorithms mostly utilize propagation methods to estimate the sentiment score of each phrase. These methods typically employ parsing results, syntactic contexts or linguistic information from thesaurus (e.g. WordNet) to calculate the similarity between phrases. For example, Baccianella et al. (2010) use the glosses information from WordNet; Velikovich et al. (2010) represent each phrase with its context words from the web documents; Qiu et al. (2011) exploit the dependency relations between sentiment words and aspect words. However, parsing information and the linguistic information from WordNet are not suitable for constructing large-scale sentiment lexicon from Twitter. The reason lies in that WordNet cannot well cover the colloquial expressions in tweets, and it is hard to have reliable tweet parsers due to the informal language style.

In this paper, we propose to build large-scale sentiment lexicon from Twitter with a representation learning approach, as illustrated in Figure 1. We cast sentiment lexicon learning as a phrase-level classification task. Our method contains two part: (1) a representation learning algorithm to effectively learn the continuous representation of phrases, which are used as features for phrase-level sentiment classification, (2) a seed expansion algorithm that enlarge a small list of sentiment seeds to collect training data for building the phrase-level classifier. Specifically, we learn sentiment-specific phrase embedding (**SSPE**), which is a low-dimensional, dense and real-valued vector, by encoding the sentiment information and

[*]This work was partly done when the first author was visiting Microsoft Research.
[†]Corresponding author.

Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers, pages 172–182, Dublin, Ireland, August 23-29 2014.

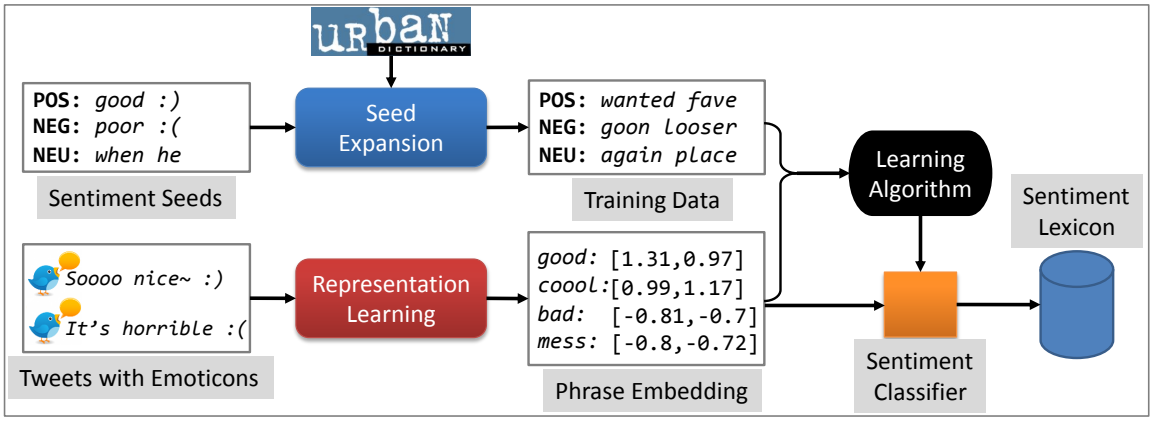


Figure 1: The representation learning approach for building Twitter-specific sentiment lexicon.

syntactic contexts into the continuous representation of phrases [1]. As a result, the nearest neighbors in the embedding space of SSPE are favored to have similar semantic usage as well as the same sentiment polarity. To this end, we extend the existing phrase embedding learning algorithm (Mikolov et al., 2013b), and develop a dedicated neural architecture with hybrid loss function to incorporate the supervision from sentiment polarity of text (e.g. tweets). We learn SSPE from tweets, leveraging massive tweets containing positive and negative emoticons as training set without any manual annotation. To obtain more training data for building the phrase-level sentiment classifier, we exploit the similar words from Urban Dictionary [2], which is a crowd-sourcing resource, to expand a small list of sentiment seeds. Finally, we utilize the classifier to predict the sentiment score of each phrase in the vocabulary of SSPE, resulting in the sentiment lexicon.

We evaluate the effectiveness of our sentiment lexicon (**TS-Lex**) by applying it in a supervised learning framework (Pang et al., 2002) for Twitter sentiment classification. Experiment results on the benchmark dataset of SemEval 2013 show that, TS-Lex yields better performance than previously introduced lexicons, including two large-scale Twitter-specific sentiment lexicons, and further improves the top-performed system in SemEval 2013 by feature combination. The quality of SSPE is also evaluated by regarding SSPE as the feature for sentiment classification of the items in existing sentiment lexicons (Hu and Liu, 2004; Wilson et al., 2005). Experiment results show that SSPE outperforms existing embedding learning algorithms. The main contributions of this work are as follows:

- To our best knowledge, this is the first work that leverages the continuous representation of phrases for building large-scale sentiment lexicon from Twitter;
- We propose a tailored neural architecture for learning the sentiment-specific phrase embedding from massive tweets selected with positive and negative emoticons;
- We report the results that our lexicon outperforms existing sentiment lexicons by applying them in a supervised learning framework for Twitter sentiment classification.

2 Related Work

In this section, we give a brief review about building sentiment lexicon and learning continuous representation of words and phrases.

2.1 Sentiment Lexicon Learning

Sentiment lexicon is a fundamental component for sentiment analysis, which can be built manually (Das and Chen, 2007), through heuristics (Kim and Hovy, 2004) or using machine learning algorithms (Turney, 2002; Li et al., 2012; Xu et al., 2013). Existing studies typically employ machine learning methods

[1] Word/unigram is also regarded as phrase in this paper.
[2] http://www.urbandictionary.com/

and adopt the propagation method to build sentiment lexicon. In the first step, a graph is built by regarding each item (word or phrase) as a node and their similarity as the edge. Then, graph propagation algorithms, such as pagerank (Esuli and Sebastiani, 2007), label propagation (Rao and Ravichandran, 2009) or random walk (Baccianella et al., 2010), are utilized to iteratively calculate the sentiment score of each item. Under this direction, parsing results, syntactic contexts or linguistic clues in thesaurus are mostly explored to calculate the similarity between items. Wiebe (2000) utilize the dependency triples from an existing parser (Lin, 1994). Qiu et al. (2009; 2011) adopt dependency relations between sentiment words and aspect words. Esuli and Sebastiani (2005) exploit the glosses information from Wordnet. Hu and Liu (2004) use the synonym and antonym relations within linguistic resources. Velikovich et al. (2010) represent words and phrases with their syntactic contexts within a window size from the web documents. Unlike the dominated propagation based methods, we explore the classification framework based on representation learning for building large-scale sentiment lexicon from Twitter.

To construct the Twitter-specific sentiment lexicon, Mohammad et al. (2013) use pointwise mutual information (PMI) between each phrase and hashtag/emoticon seed words, such as *#good*, *#bad*, *:)* and *:(*. Chen et al. (2012) utilize the Urban Dictionary and extract the target-dependent sentiment expressions from Twitter. Unlike Mohammad et al. (2013) that only capture the relations between phrases and sentiment seeds, we exploit the semantic and sentimental connections between phrases through phrase embedding and propose a representation learning approach to build sentiment lexicon.

2.2 Learning Continuous Representation of Word and Phrase

Continuous representation of words and phrases are proven effective in many NLP tasks (Turian et al., 2010). Embedding learning algorithms have been extensively studied in recent years (Bengio et al., 2013), and are dominated by the syntactic context based algorithms (Bengio et al., 2003; Collobert et al., 2011; Dahl et al., 2012; Huang et al., 2012; Mikolov et al., 2013a; Lebret et al., 2013; Sun et al., 2014). To integrate the sentiment information of text into the word embedding, Maas et al. (2011) extend the probabilistic document model (Blei et al., 2003) and predict the sentiment of a sentence with the embedding of each word. Labutov and Lipson (2013) learn task-specific embedding from an existing embedding and sentences with gold sentiment polarity. Tang et al. (2014) propose to learn sentiment-specific word embedding from tweets collected by emoticons for Twitter sentiment classification. Unlike previous trails, we learn sentiment-specific phrase embedding with a tailored neural network. Unlike Mikolov et al. (2013b) that only use the syntactic contexts of phrases to learn phrase embedding, we integrate the sentiment information of text into our method. It is worth noting that we focus on learning the continuous representation of words and phrases, which is orthogonal with Socher et al. (2011; 2013) that learn the compositionality of sentences.

3 Methodology

In this section, we describe our method for building large-scale sentiment lexicon from Twitter within a classification framework, as illustrated in Figure 1. We leverage the continuous representation of phrases as features, without parsers or hand-crafted rules, and automatically obtain the training data by seed expansion from Urban Dictionary. After the classifier is built, we employ it to predict the sentiment distribution of each phrase in the embedding vocabulary, resulting in the sentiment lexicon. To encode the sentiment information into the continuous representation of phrases, we extend an existing phrase embedding learning algorithm (Mikolov et al., 2013b) and develop a tailored neural architecture to learn sentiment-specific phrase embedding (**SSPE**), as described in subsection 3.1. To automatically obtain more training data for building the phrase-level sentiment classifier, we use the similar words from Urban Dictionary to expand a small list of sentiment seeds, as described in subsection 3.2.

3.1 Sentiment-Specific Phrase Embedding

Mikolov et al. (2013b) introduce Skip-Gram to learn phrase embedding based on the context words of phrases, as illustrated in Figure 2(a).

Given a phrase w_i, Skip-Gram maps it into its continuous representation e_i. Then, Skip-Gram utilizes

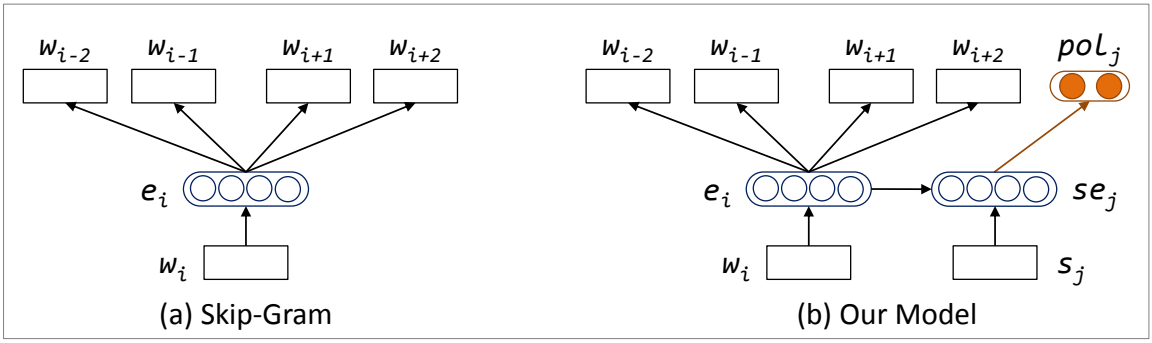


Figure 2: The traditional Skip-Gram model and our neural architecture for learning sentiment-specific phrase embedding (SSPE).

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. Hierarchical softmax (Morin and Bengio, 2005) is leveraged to accelerate the training procedure because the vocabulary size of phrase table is typically huge. The objective of Skip-Gram is to maximize the average log probability:

$$f_{syntactic} = \frac{1}{T}\sum_{i=1}^{T}\sum_{-c\leq j\leq c, j\neq 0} log\ p(w_{i+j}|e_i) \tag{1}$$

where T is the occurrence of each phrase in the corpus, c is the window size, e_i is the embedding of the current phrase w_i, w_{i+j} is the context words of w_i, $p(w_{i+j}|e_i)$ is calculated with hierarchical softmax. The basic $softmax$ unit is calculated as $softmax_i = exp(z_i)/\sum_k exp(z_k)$. We leave out the details of hierarchical softmax (Morin and Bengio, 2005; Mikolov et al., 2013b) due to the page limit. It is worth noting that, Skip-Gram is capable to learn continuous representation of words and phrases with the identical model (Mikolov et al., 2013b).

To integrate sentiment information into the continuous representation of phrases, we develop a tailored neural architecture to learn SSPE, as illustrated in Figure 2(b). Given a triple $\langle w_i, s_j, pol_j\rangle$ as input, where w_i is a 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 (Huang et al., 2012). The objective of the sentiment part is to maximize the average of log sentiment probability:

$$f_{sentiment} = \frac{1}{S}\sum_{j=1}^{S} log\ p(pol_j|se_j) \tag{2}$$

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. Our final training objective is to maximize the linear combination of the syntactic and sentiment parts:

$$f = \alpha \cdot f_{syntactic} + (1-\alpha)\cdot f_{sentiment} \tag{3}$$

where α weights the two parts. Accordingly, the nearest neighbors in the embedding space of SSPE are favored to have similar semantic usage as well as the same sentiment polarity.

We train our neural model with stochastic gradient descent and use AdaGrad (Duchi et al., 2011) to update the parameters. We empirically set embedding length as 50, window size as 3 and the learning rate of AdaGrad as 0.1. Hyper-parameter α is tuned on the development set. To obtain large-scale training corpus, we collect tweets from April, 2013 through TwitterAPI. After filtering the tweets that are too short ($<$ 5 words) and removing *@user* and *URLs*, we collect 10M tweets (5M positive and 5M negative) with positive and negative emoticons [3], which is are utilized as the training data to train our neural model. The vocabulary size is 750,000 after filtering the 1$\sim$4 grams through frequency.

[3]We use the emoticons selected by Hu et al. (2013), namely :) :) :-) :D =) as positive and :(: (:-(as negative ones.

3.2 Seed Expansion with Urban Dictionary

Urban Dictionary is a web-based dictionary that contains more than seven million definitions until March, 2013 [4]. It was intended as a dictionary of slang, cultural words or phrases not typically found in standard dictionaries, but it is now used to define any word or phrase. For each item in Urban Dictionary, there is a list of similar words contributed by volunteers. For example, the similar words of "*cooool*" are "*cool*", "*awesome*", "*coooool*", et al [5] and the similar words of "*not bad*" are "*good*", "*ok*" and "*cool*", et al [6]. These similar words are typically semantically close to and have the same sentiment polarity with the target word. We conduct preliminary statistic on the items of Urban Dictionary from "*a*" to "*z*", and find that there are total 799,430 items containing similar words and each of them has about 10.27 similar words on average.

We utilize Urban Dictionary to expand little sentiment seeds for collecting training data for building the phrase-level sentiment classifier. We manually label the top frequent 500 words from the vocabulary of SSPE as positive, negative or neutral. After removing the ambiguous ones, we obtain 125 positive, 109 negative and 140 neutral words, which are regarded as the sentiment seeds [7]. Afterwards, we leverage the similar words from Urban Dictionary to expand the sentiment seeds. We first build a k-nearest neighbors (KNN) classifier by regarding the sentiment seeds as gold standard. Then, we employ the KNN classifier on the items of Urban Dictionary containing similar words, and predict a three-dimensional discrete vector $[knn_{pos}, knn_{neg}, knn_{neu}]$ for each item, reflecting the hits numbers of sentiment seeds with different sentiment polarity in its similar words. For example, the vector value of "*not bad*" is $[10, 0, 0]$, which means that there are 10 positive seeds, 0 negative seeds and 0 neutral seeds occur in its similar words. To ensure the quality of the expanded words, we set threshold for each category to collect the items with high quality as expanded words. Take the positive category as an example, we keep an item as positive expanded word if it satisfies $knn_{pos} > knn_{neg} + threshold_{pos}$ and $knn_{pos} > knn_{neu} + threshold_{pos}$ simultaneously. We empirically set the thresholds of positive, negative and neutral as 6,3,2 respectively by balancing the size of expanded words in three categories. After seed expansion, we collect 1,512 positive, 1,345 negative and 962 neutral words, which are used as the training data to build the phrase-level sentiment classifier. We also tried the propagation methods to expand the sentiment seeds, namely iteratively added the similar words of sentiment seeds from Urban Dictionary into the expanded word collection. However, the quantity of expanded words is less than the KNN-based results and the quality is relatively poor.

After obtaining the training data and feature representation of phrases, we build the phrase-level classifier with $softmax$, whose length is two for the positive *vs* negative case:

$$\mathbf{y}(w) = softmax(\theta \cdot e_i + b) \tag{4}$$

where θ and b are the parameters of classifier, e_i is the embedding of the current phrase w_i, $\mathbf{y}(w)$ is the predicted sentiment distribution of item w_i. We employ the classifier to predict the sentiment distribution of each phrase in the vocabulary of SSPE, and save the phrases as well as their sentiment probability in the positive (negative) lexicon if the positive (negative) probability is larger than 0.5.

4 Experiment

In this section, we conduct experiments to evaluate the effectiveness of our sentiment lexicon (**TS-Lex**) by applying it in the supervised learning framework for Twitter sentiment classification, as given in subsection 4.1. We also directly evaluate the quality of SSPE as it forms the fundamental component for building sentiment lexicon. We use SSPE as the feature for sentiment classification of items in existing sentiment lexicons, as described in subsection 4.2.

[4] http://en.wikipedia.org/wiki/Urban_Dictionary
[5] http://www.urbandictionary.com/define.php?term=cooool
[6] http://www.urbandictionary.com/define.php?term=not+bad
[7] We will publish the sentiment seeds later.

4.1 Twitter Sentiment Classification

Experiment Setup and Dataset We conduct experiments on the benchmark Twitter sentiment classification dataset (message-level) from SemEval 2013 (Nakov et al., 2013). The training and development sets were completely released to task participants. However, we were unable to download all the training and development sets because some tweets were deleted or not available due to modified authorization status. The statistic of the positive and negative tweets in our dataset are given in Table 1(b). We train positive *vs* negative classifier with LibLinear (Fan et al., 2008) with default settings on the training set, tune parameters *-c* on the dev set and evaluate on the test set. The evaluation metric is Macro-F1.

(a) Sentiment Lexicons

Lexicon	Positive	Negative	Total
HL	2,006	4,780	6,786
MPQA	2,301	4,150	6,451
NRC-Emotion	2,231	3,324	5,555
TS-Lex	**178,781**	**168,845**	**347,626**
HashtagLex	216,791	153,869	370,660
Sentiment140Lex	480,008	260,158	740,166

(b) SemEval 2013 Dataset

	Positive	Negative	Total
Train	2,642	994	3,636
Dev	408	219	627
Test	1,570	601	2,171

Table 1: Statistic of sentiment lexicons and Twitter sentiment classification datasets.

Results and Analysis We compare TS-Lex with *HL*[8] (Hu and Liu, 2004), *MPQA*[9] (Wilson et al., 2005), *NRC-Emotion*[10] (Mohammad and Turney, 2012), *HashtagLex* and *Sentiment140Lex*[11] (Mohammad et al., 2013). The statistics of TS-Lex and other sentiment lexicons are illustrated in Table 1(a). *HL*, *MPQA* and *NRC-Emotion* are traditional sentiment lexicons with a relative small lexicon size. *HashtagLex* and *Sentiment140Lex* are Twitter-specific sentiment lexicons. We can find that, TS-Lex is larger than the traditional sentiment lexicons.

We evaluate the effectiveness of TS-Lex by applying it as the features for Twitter sentiment classification in the supervised learning framework (Pang et al., 2002). We conduct experiments in two settings, namely only utilizing the lexicon features (*Unique*) and appending lexicon feature to existing feature sets (*Appended*). In the first setting, we design the lexicon features as same as the top-performed Twitter sentiment classification system in SemEval2013 [12] (Mohammad et al., 2013). For each sentiment polarity (positive *vs* negative), the lexicon features are:

- total count of tokens in the tweet with score greater than 0;
- the sum of the scores for all tokens in the tweet;
- the maximal score;
- the non-zero score of the last token in the tweet;

In the second experiment setting, we append the lexicon features to the existing basic feature. We use the feature sets of Mohammad et al. (2013) excluding the lexicon feature as the basic feature, including bag-of-words, pos-tagging, emoticons, hashtags, elongated words, etc. Experiment results of the *Unique* features and *Appended* features from different sentiment lexicons on Twitter sentiment classification are given in Table 2(a).

From Table 2(a), we can find that TS-Lex yields best performance in both *Unique* and *Appended* feature sets among all sentiment lexicons, including two large-scale Twitter-specific sentiment lexicons. The reason is that the classifier for building TS-Lex utilize (1) the well developed feature representation of phrases (SSPE), which captures the semantic and sentiment connections between phrases, and (2) the enlarged sentiment words through web intelligence as training data. *HashtagLex* and *Sentiment140Lex*

[8] http://www.cs.uic.edu/ liub/FBS/sentiment-analysis.html#lexicon
[9] http://mpqa.cs.pitt.edu/lexicons/subj_lexicon/
[10] http://www.saifmohammad.com/WebPages/ResearchInterests.html
[11] We utilize the unigram and bigram lexicons from *HashtagLex* and *Sentiment140Lex*.
[12] http://www.saifmohammad.com/WebPages/Abstracts/NRC-SentimentAnalysis.htm

(a)

Lexicon	Unique	Appended
HL	60.49	79.40
MPQA	59.15	76.54
NRC-Emotion	54.81	76.79
HashtagLex	65.30	76.67
Sentiment140Lex	72.51	80.68
TS-Lex	**78.07**	**82.36**

(b)

Lexicon	Unique
Seed	57.92
Expand	60.69
Lexicon(seed)	74.64
TS-Lex	**78.07**

Table 2: Macro-F1 on Twitter sentiment classification with different lexicon features.

only utilize the relations between phrases and hashtag/emoticon seeds, yet do not well capture the connections between phrases. In the *Unique* setting, the performances of the traditional lexicons (*HL*, *MPQA* and *NRC-Emotion*) are lower than large-scale Twitter-specific lexicons (*HashtagLex*, *Sentiment140Lex* and our lexicon). The reason is that, tweets have the informal language style and contain slangs and diverse multi-word phrases, which are not well covered by the traditional sentiment lexicons with a small size. After incorporating the lexicon feature of TS-Lex into the top-performs system (Mohammad et al., 2013), we further improve the macro-F1 from 84.70% to 85.65%.

Effect of Seed Expansion with Urban Dictionary To verify the effectiveness of seed expansion through Urban Dictionary, we conduct experiments by applying (1) sentiment seeds (*Seed*), (2) words after expansion (*Expand*), (3) sentiment lexicon generated from the classifier only utilizing sentiment seeds as training data (*Lexicon(seed)*), (4) the final lexicon (*TS-Lex*) exploiting the expanded words as training data to build sentiment classifier, to produce lexicon features, and only use them for Twitter sentiment classification (*Unique*). From Table 2(b), we find that the performance of sentiment seeds and expanded words are relatively poor due to their low coverage. Under this scenario, seed expansion yields 2.77% improvement (from 57.92% to 60.69%) on macro-F1. By utilizing the expanded words as training data to build the phrase-level sentiment classifier, TS-Lex obtains 3.43% improvements on Twitter sentiment classification (from 74.64% to 78.07%), which verifies the effectiveness of seed expansion through Urban Dictionary. In addition, we find that only using a small number of sentiment seeds as the training data, we can obtain superior performance (74.64%) than all baseline lexicons. This indicates that the representation learning approach effectively capture the semantic and sentimental connections between phrases through SSPE, and leverage them for building the sentiment lexicon.

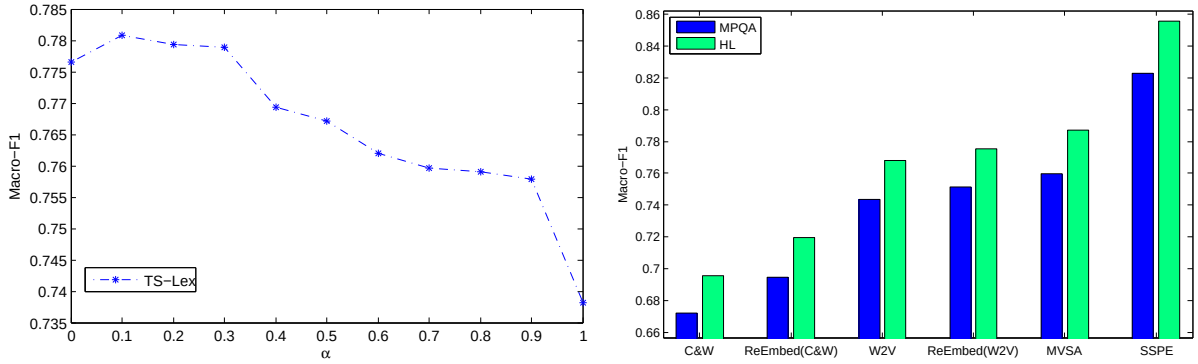
Effect of α in SSPE We tune the hyper-parameter α of SSPE on the development set of SemEval 2013, and study its influence on the performance of Twitter sentiment classification by applying the generated lexicon as features. We utilize the expanded words as training data to train $softmax$ and only utilize the lexicon features (*Unique*) for Twitter sentiment classification. Experiment results with different α are illustrated in Figure 3(a).

From Figure 3(a), we can see that that SSPE performs better when α is in the range of [0.1, 0.3], which is dominated by the sentiment information. The model with $\alpha = 1$ stands for Skip-Gram model. The sharp decline at $\alpha = 1$ indicates the importance of sentiment information in learning sentiment-specific phrase embedding for building sentiment lexicon.

Discussion In the experiment, we do not apply TS-Lex into the unsupervised learning framework for Twitter sentiment classification. The reason is that the lexicon-based unsupervised method typically require the sentiment lexicon to have high precision, yet our task is to build large-scale lexicon (TS-Lex) with broad coverage. We leave this as the future work, although we may set higher threshold (e.g. larger than 0.5) to increase the precision of TS-Lex and loose the recall.

4.2 Evaluation of Different Representation Learning Methods

Experiment Setup and Dataset We conduct sentiment classification of items in two traditional sentiment lexicons, *HL* (Hu and Liu, 2004) and *MPQA* (Wilson et al., 2005), to evaluate the effective of the



(a) SSPE with different α on the development set for Twitter sentiment classification.

(b) Sentiment classification of lexicons with different embedding learning algrithms.

Figure 3: Experiment results with different settings.

sentiment-specific phrase embedding (SSPE). We train the positive *vs* negative classifier with LibLinear (Fan et al., 2008). The evaluation metric is the macro-F1 of 5-fold cross validation. The statistics of *HL* and *MPQA* are listed in Table 1(a).

Baseline Embedding Learning Algorithms We compare SSPE with the following embedding learning algorithms:

(1) *C&W*. C&W is one of the most representative embedding learning algorithms (Collobert et al., 2011) for learning word embedding, which has been proven effective in many NLP tasks.

(2) *W2V*. Mikolov et al. (2013a) introduce Word2Vec for learning the continuous vectors for words and phrases. We utilize Skip-Gram as it performs better than CBOW in the experiments.

(3) *MVSA*. Maas et al. (2011) learn word vectors for sentiment analysis with a probabilistic model of documents utilizing the sentiment polarity of documents.

(4) *ReEmbed*. Lebret et al. (2013) learn task-specific embedding from existing embedding and task-specific corpus. We utilize the training set of Twitter sentiment classification as the labeled corpus to re-embed words. *ReEmbed(C&W)* and *ReEmbed(W2V)* stand for the use of different embedding results as the reference word embedding.

The embedding results of the baseline algorithms and SSPE are trained with the same dataset and parameter sets.

Results and Analysis Experiment results of the baseline embedding learning algorithms and SSPE are given in Figure 3(b). We can see that SSPE yields best performance on both lexicons. The reason is that SSPE effectively encode the sentiment information of tweets as well as the syntactic contexts of phrases from massive data into the continuous representation of phrases. The performances of *C&W* and *W2V* are relatively low because they only utilize the syntactic contexts of items, yet ignore the sentiment information of text, which is crucial for sentiment analysis. *ReEmbed(C&W)* and *ReEmbed(W2V)* achieve better performance than *C&W* and *W2V* because the sentiment information of sentences are incorporated into the continuous representation of phrases. There is a gap between *ReEmbed* and SSPE because SSPE leverages more sentiment supervision from massive tweets collected by positive and negative emoticons.

5 Conclusion

In this paper, we propose building large-scale Twitter-specific sentiment lexicon with a representation learning approach. Our method contains two parts: (1) a representation learning algorithm to effectively learn the embedding of phrases, which are used as features for classification, (2) a seed expansion algorithm that enlarge a small list of sentiment seeds to obtain training data for building the phrase-level sentiment classifier. We introduce a tailored neural architecture and integrate the sentiment information of tweets into its hybrid loss function for learning sentiment-specific phrase embedding (**SSPE**). We learn SSPE from the tweets collected by positive and negative emoticons, without any manual annota-

tion. To collect more training data for building the phrase-level classifier, we utilize the similar words from Urban Dictionary to expand a small list of sentiment seeds. The effectiveness of our sentiment lexicon (**TS-Lex**) has been verified through applied in the supervised learning framework for Twitter sentiment classification. Experiment results on the benchmark dataset of SemEval 2013 show that, TS-Lex outperforms previously introduced sentiment lexicons and further improves the top-perform system in SemEval 2013 with feature combination. In future work, we plan to apply TS-Lex into the unsupervised learning framework for Twitter sentiment classification.

Acknowledgements

We thank Nan Yang, Yajuan Duan and Yaming Sun for their great help. This research was partly supported by National Natural Science Foundation of China (No.61133012, No.61273321, No.61300113).

References

Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. 2010. Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *LREC*, volume 10, pages 2200–2204.

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

Yoshua Bengio, Aaron Courville, and Pascal Vincent. 2013. Representation learning: A review and new perspectives. *IEEE Trans. Pattern Analysis and Machine Intelligence*.

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

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*.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12:2493–2537.

George E Dahl, Ryan P Adams, and Hugo Larochelle. 2012. Training restricted boltzmann machines on word observations. *ICML*.

Sanjiv R Das and Mike Y Chen. 2007. Yahoo! for amazon: Sentiment extraction from small talk on the web. *Management Science*, 53(9):1375–1388.

John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *The Journal of Machine Learning Research*, pages 2121–2159.

Andrea Esuli and Fabrizio Sebastiani. 2005. Determining the semantic orientation of terms through gloss classification. In *Proceedings of the 14th ACM international conference on Information and knowledge management*, pages 617–624. ACM.

Andrea Esuli and Fabrizio Sebastiani. 2007. Pageranking wordnet synsets: An application to opinion mining. In *ACL*, volume 7, pages 442–431.

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. Liblinear: A library for large linear classification. *The Journal of Machine Learning Research*, 9:1871–1874.

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

Ming Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 168–177.

Xia Hu, Jiliang Tang, Huiji Gao, and Huan Liu. 2013. Unsupervised sentiment analysis with emotional signals. In *Proceedings of the International World Wide Web Conference*, pages 607–618.

Eric H Huang, Richard Socher, Christopher D Manning, and Andrew Y Ng. 2012. Improving word representations via global context and multiple word prototypes. In *ACL*, pages 873–882. ACL.

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

Igor Labutov and Hod Lipson. 2013. Re-embedding words. In *Annual Meeting of the Association for Computational Linguistics*.

Rémi Lebret, Joël Legrand, and Ronan Collobert. 2013. Is deep learning really necessary for word embeddings? *NIPS workshop*.

Fangtao Li, Sinno Jialin Pan, Ou Jin, Qiang Yang, and Xiaoyan Zhu. 2012. Cross-domain co-extraction of sentiment and topic lexicons. In *Proceedings of the 50th ACL*, pages 410–419. ACL, July.

Dekang Lin. 1994. Principar: an efficient, broad-coverage, principle-based parser. In *Proceedings of the 15th conference on COLING*, pages 482–488. Association for Computational Linguistics.

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

Andrew L Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *ICLR*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013b. Distributed representations of words and phrases and their compositionality. *The Conference on Neural Information Processing Systems*.

Saif M Mohammad and Peter D Turney. 2012. Crowdsourcing a word–emotion association lexicon. *Computational Intelligence*.

Saif M Mohammad, Svetlana Kiritchenko, and Xiaodan Zhu. 2013. Nrc-canada: Building the state-of-the-art in sentiment analysis of tweets. *Proceedings of the International Workshop on Semantic Evaluation*.

Frederic Morin and Yoshua Bengio. 2005. Hierarchical probabilistic neural network language model. In *Proceedings of the international workshop on artificial intelligence and statistics*, pages 246–252.

Preslav Nakov, Sara Rosenthal, Zornitsa Kozareva, Veselin Stoyanov, Alan Ritter, and Theresa Wilson. 2013. Semeval-2013 task 2: Sentiment analysis in twitter. In *Proceedings of the International Workshop on Semantic Evaluation*, volume 13.

Bo Pang and Lillian Lee. 2008. Opinion mining and sentiment analysis. *Foundations and trends in information retrieval*, 2(1-2):1–135.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 79–86.

Guang Qiu, Bing Liu, Jiajun Bu, and Chun Chen. 2009. Expanding domain sentiment lexicon through double propagation. In *IJCAI*, volume 9, pages 1199–1204.

Guang Qiu, Bing Liu, Jiajun Bu, and Chun Chen. 2011. Opinion word expansion and target extraction through double propagation. *Computational linguistics*, 37(1):9–27.

Delip Rao and Deepak Ravichandran. 2009. Semi-supervised polarity lexicon induction. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, pages 675–682. Association for Computational Linguistics.

Richard Socher, J. Pennington, E.H. Huang, A.Y. Ng, and C.D. Manning. 2011. Semi-supervised recursive autoencoders for predicting sentiment distributions. In *Conference on Empirical Methods in Natural Language Processing*, pages 151–161.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642.

Yaming Sun, Lei Lin, Duyu Tang, Nan Yang, Zhenzhou Ji, and Xiaolong Wang. 2014. Radical-enhanced chinese character embedding. *arXiv preprint arXiv:1404.4714*.

Duyu Tang, Furu Wei, Nan Yang, Ming Zhou, Ting Liu, and Bing Qin. 2014. Learning sentiment-specific word embedding for twitter sentiment classification. In *Procedding of the 52th Annual Meeting of Association for Computational Linguistics.*

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

Peter D Turney. 2002. Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In *Proceedings of Annual Meeting of the Association for Computational Linguistics*, pages 417–424.

Leonid Velikovich, Sasha Blair-Goldensohn, Kerry Hannan, and Ryan McDonald. 2010. The viability of web-derived polarity lexicons. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 777–785. Association for Computational Linguistics.

Janyce Wiebe. 2000. Learning subjective adjectives from corpora. In *AAAI/IAAI*, pages 735–740.

Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 347–354.

Liheng Xu, Kang Liu, Siwei Lai, Yubo Chen, and Jun Zhao. 2013. Mining opinion words and opinion targets in a two-stage framework. In *Proceedings of the 51st ACL*, pages 1764–1773. ACL.