Sentiment Lexicon Construction with Representation Learning Based on Hierarchical Sentiment Supervision

Leyi Wang and **Rui Xia***
School of Computer Science & Engineering
Nanjing University of Science & Technology, China
leyiwang.cn@gmail.com, rxia@njust.edu.cn

Abstract

Sentiment lexicon is an important tool for identifying the sentiment polarity of words and texts. How to automatically construct sentiment lexicons has become a research topic in the field of sentiment analysis and opinion mining. Recently there were some attempts to employ representation learning algorithms to construct a sentiment lexicon with sentiment-aware word embedding. However, these methods were normally trained under document-level sentiment supervision. In this paper, we develop a neural architecture to train a sentiment-aware word embedding by integrating the sentiment supervision at both document and word levels, to enhance the quality of word embedding as well as the sentiment lexicon. Experiments on the SemEval 2013-2016 datasets indicate that the sentiment lexicon generated by our approach achieves the state-of-the-art performance in both supervised and unsupervised sentiment classification, in comparison with several strong sentiment lexicon construction methods.

1 Introduction

Sentiment lexicon is a set of words (or phrases) each of which is assigned with a sentiment polarity score. Sentiment lexicon plays an important role in many practical sentiment analysis and opinion mining tasks. There were some manually annotated universal sentiment lexicons such as General Inquireer (GI) and HowNet. However, due to the ubiquitous domain diversity and absence of domain prior knowledge, the automatic construction technique for domain-specific sentiment lexicons has become a challenging research topic in the field of sentiment analysis and opinion mining (Wang and Xia, 2016).

The early work employed unsupervised learning for sentiment lexicon construction. They normally labelled a set of seed words at first, and then learned the polarity of each candidate word, based on either word conjunction relations (e.g., constellation and transition in texts) (Hatzivassiloglou and McKeown, 1997), or the word co-occurrence information (such as pointwise mutual information, PMI) (Turney, 2002), between the candidate word and the seed words. However, the unsupervised manner showed limited effect in sentiment prediction, and the performance greatly depends on the quality of the seed words.

To fully exploit the sentiment labeling information in texts, a series of supervised learning methods was further proposed to learn the sentiment lexicons. For example, Mohammad et al. (2013) proposed to construct sentiment lexicons by calculating PMI between the word and the distantly supervised sentiment labels (such as emoticons) in tweets and the word's sentiment orientation (SO). The resulting lexicons obtained the best results in SemEval 2013. More advanced representation learning models were also utilized, with the aim to construct the sentiment lexicons with efficient word embeddings (Tang et al., 2014a; Hamilton et al., 2016; Vo and Zhang, 2016). The traditional representation learning framework such as Word2Vec only captures the syntactic information in the texts, but ignores the sentiment relations between words. Therefore, some researchers attempted to add sentiment supervision into the network structure, in order to train a sentiment-aware word embedding. For example, Tang et al. (2014a) exploited a dedicated neural architecture to integrate document-level sentiment supervision and the syntactic knowledge for representation

*The corresponding author of this paper.

learning. The sentiment-aware word embedding is then used to construct a sentiment lexicon. Vo and Zhang (2016) proposed to learn a two-dimensional sentiment representation based on a simple neural network. The sentiment lexicons generated by their approach obtained better performance to predict the tweet sentiment labels, in comparison with the PMI-based method (Mohammad et al., 2013).

Although these supervised learning methods can to some extent exploit the sentiment labeling information in the texts and can learn a sentiment-aware word embedding, the manner of using document-level sentiment supervision suffers from some complex linguistic phenomena such as negation, transition and comparative degree, and hence unable to capture the fine-grained sentiment information in the text. For example, in the following tweet

> *"Four more fake people added me. Is this why people don't like Twitter? :(",*

the document-level sentiment label is negative, but there is a positive word *"like"* in the text. In representation learning, the embeddings of words are summed up to represent the document, and the word *"like"* will be falsely associated with the negative sentiment label. Such linguistic phenomena occur frequently in review texts, and makes sentiment-aware word representation learning less effective.

To address this problem, in this paper, we propose a new representation learning framework called HSSWE, to learn sentiment-aware word embeddings based on hierarchical sentiment supervision. In HSSWE, the learning algorithm is supervised under both document-level sentiment labels and word-level sentiment annotations (e.g., labeling *"like"* as a positive word). By leveraging the sentiment supervision at both document and word level, our approach can avoid the sentiment learning flaws caused by coarse-grained document-level supervision by incorporating fine-grained word-level supervision, and improve the quality of sentiment-aware word embedding. Finally, following Tang et al. (2014a), a simple classifier was constructed to obtain the domain-specific sentiment lexicon by using word embeddings as inputs.

The main contributions of this work are as follows:

1. To the best of our knowledge, this is the first work that learns the sentiment-aware word representation under supervision at both document and word levels.

2. Our approach supports several kinds of word-level sentiment annotations such as 1) predefined sentiment lexicon; 2) PMI-SO lexicon with hard sentiment annotation; 3) PMI-SO lexicon with soft sentiment annotation. By using PMI-SO dictionary as word-level sentiment annotation, our approach is totally corpus-based, without any external resource.

3. Our approach obtains the state-of-the-art performance in comparison with several strong sentiment lexicon construction methods, on the benchmark SemEval 2013-2016 datasets for twitter sentiment classification.

2 Related Work

In general, sentiment lexicons construction can be classified into two categories, dictionary-based methods and corpus-based methods.

Dictionary-based methods generally integrate predefined resources, such as WordNet, to construct sentiment lexicons. Hu and Liu (2004) exploited WordNet for sentiment lexicon construction. They first labelled two sets of seed words by polarities, then extended the sets by adding the synonyms for each word to the same set and antonyms to the other. For a given new word, Kim and Hovy (2004) introduced a Naive Bayes model to predict the polarities with .the synonym set obtained from WordNet as features. Kamps et al. (2004) investigated a graph-theoretic model of WordNet's synonymy relation and measured the sentiment orientation by distance between each candidate word and the seed words with different polarities. Heerschop et al. (2011) proposed a method to propagate the sentiment of seed set words through semantic relations of WordNet.

Corpus-based approaches originate from the latent relation hypothesis: "*Pairs of words that co-occur in similar patterns tend to have similar semantic and sentiment relations*" (Turney, 2008).

The primary corpus-based method made the use of PMI. Turney (2002) built a sentiment lexicon by calculating PMI between the candidate word and seed words. The difference of the PMI score between positive and negative seed words is finally used as the sentiment orientation (SO) of each candidate word (Turney, 2002). Many variants of

PMI were proposed afterwards, for example, positive pointwise mutual information (PPMI), second order co-occurrence PMI (SOC-PMI), etc. Hamilton et al. (2016) proposed to build a sentiment lexicon by a propagation method. The key of this method is to build a lexical graph by calculating the PPMI between words. Instead of calculating the PMI between words, Mohammad et al. (2013) proposed to use emoticons as distant supervision and calculate the PMI between words and the distant class labels, and obtained sound performance for tweet sentiment classification.

The latest corpus-based approaches normally utilize the up-to-date machine learning models (e.g. neural networks) to first learn a sentiment-aware distributed representation of words, based on which the sentiment lexicon is then constructed. There were many word representation learning methods such as NNLM (Bengio et al., 2003) and Word2Vec (Mikolov et al., 2013). However, they mainly consider the syntactic relation of words in the context but ignore the sentiment information. Some work were later proposed to deal with this problem by incorporating the sentiment information during representation learning. For example, Tang et al. (2014a) adapted a variant of skip-gram model, which can learn the sentiment information based on distant supervision. Furthermore, Tang et al. (2014b) proposed a new neural network approach called SSWE to train sentiment-aware word representation. Vo and Zhang (2016) exploited a simple and fast neural network to train a 2-dimensional representation. Each dimension is explicitly associated with a sentiment polarity.

The sentiment-aware word representation in these methods was normally trained based on only document-level sentiment supervision. In contrast, the learning algorithm in our approach is supervised under both document-level and word-level sentiment supervision.

3 Our Approach

Our approach is comprised of three base modules: (1) Word-level sentiment learning and annotation; (2) Sentiment-aware word embedding learning; (3) Sentiment lexicon construction.

Our approach depends on document-level sentiment labels. The tweet corpus provides a cheap way to get document-level sentiment annotation, owing to the distant sentiment supervision. But it should be noted that our approach is feasible for any corpus provided with document-level sentiment labels (not merely tweets).

The first module of our method aims to learn the pseudo sentiment distribution for each word and use it as word-level sentiment annotations to supervise word embedding learning.

In the second module, we learn the sentiment-aware embeddings for each word in corpus, based on hierarchical sentiment supervision.

In the last module, we construct a sentiment lexicon by using the sentiment-aware word embeddings as the basis.

3.1 Learning Word-Level Sentiment Supervision

In addition to use a pre-defined sentiment lexicon for word-level annotations, we also propose to learn the word-level sentiment supervision, based on PMI and SO.

(1) PMI and SO

Given a corpus with document-level class labels. We first compute the PMI score between each word t and two class labels

$$PMI(t,+) = \log \frac{p(+|t)}{p(+)}, \quad (1)$$

$$PMI(t,-) = \log \frac{p(-|t)}{p(-)}, \quad (2)$$

where $+$ and $-$ denote the positive and negative document-level class labels, respectively.

Second, we compute the SO score for each word t:

$$SO(t) = PMI(t,+) - PMI(t,-). \quad (3)$$

We call $\{t, SO(t)\}$ as PMI-SO dictionary. The PMI-SO dictionary was widely used as a corpus-based sentiment lexicon for sentiment classification. By contrast, in our approach, it is the first step to learn the sentiment-aware word representation. Our approach supports two kinds of word-level sentiment annotations: 1) PMI-SO dictionary with hard sentiment annotation; 2) PMI-SO dictionary with soft sentiment annotation.

The word-level sentiment annotation is represented as $[\hat{p}(-|t), \hat{p}(+|t)]$. We employ the following two ways to obtain $[\hat{p}(-|t), \hat{p}(+|t)]$.

(2) PMI-SO lexicon with hard sentiment annotation

Notations	Description
e_t	The embedding of word t
de	The document representation of d
b_t	The bias of word-level softmax layer
b_d	The bias of document-level softmax layer
θ_t	Weight of word-level softmax layer
θ_d	Weight of document-level softmax layer
$p(c\|e_t)$	The sentiment distribution of word t predicted by our model
$p(c\|de)$	The sentiment distribution of document d predicted by our model
$\hat{p}(c\|t)$	The word-level sentiment annotation of word t with respect to class c
$\hat{p}(c\|d)$	The document-level sentiment annotation of document d with respect to class c

Table 1: The parameters used in our neural network.

"Hard sentiment annotation" indicates that $[\hat{p}(-|t), \hat{p}(+|t)]$ is a two-dimensional one-hot representation, where the annotation of words is given by the class labels:

$$[\hat{p}(-|t), \hat{p}(+|t)] = \begin{cases} [0,1], & \text{if } SO(t) > 0 \\ [1,0], & \text{if } SO(t) < 0 \\ random\{[1,0] or [0,1]\}, & \text{otherwise} \end{cases} \quad (4)$$

(3) PMI-SO lexicon with soft sentiment annotation

"Soft sentiment annotation" means that the annotation is given by the probability of two sentiment polarities, rather than the class label. We first use the sigmoid function to map the SO score to the range of a probability, and then define

$$[\hat{p}(-|t), \hat{p}(+|t)] = [1 - \sigma(SO(t)), \sigma(SO(t))] \quad (5)$$

as the PMI-SO soft sentiment distribution of the word t.

3.2 Learning Sentiment-aware Word Representation under Hierarchical Sentiment Supervision

Till now we have obtained both document and word-level sentiment annotations, in the next step, we propose a neural network framework to learn the sentiment-aware word representation by integrating the sentiment supervision at both word and document granularities. We call it "hierarchical sentiment supervision". The architecture of our model is shown in Figure 1. We denote the corpus as $D = \{d_1, d_2, ..., d_N\}$ where N is the size of the corpus. Suppose d_k is k-th document in D, and t_i represents the i-th word in a document d. The parameters used in our neural network are described in Table 1.

We construct a embedding matrix $C \in R^{V \times M}$, of which each row represents the embedding of a word in the vocabulary, where V is the size of the vocabulary and M is the dimension of word embedding. We randomly initialize each element of matrix C with a normal distribution.

(1) Word-Level Sentiment Supervision

We use the word-level sentiment annotation $[\hat{p}(-|t), \hat{p}(+|t)]$ provided in Section 3.1 to supervise word representation learning at the word level.

For each word in document d, we map it to a continuous representation as $e \in C$ and feed e into our model to predict the sentiment distribution of the input word:

$$p(c|e) = softmax(\theta_t \cdot e + b_t). \quad (6)$$

The cost function is defined as the average cross entropy that measures the difference between the sentiment distribution predicted in our model and the sentiment annotations at the word level:

$$f_{word} = -\frac{1}{T} \sum_{k=1}^{N} \sum_{t \in d_k} \sum_{c \in \{+,-\}} \hat{p}(c|t) \log p(c|e_t) \quad (7)$$

where T is the number of words in corpus.

(2) Document-Level Sentiment Supervision

We use the document-level sentiment annotations to supervise word representation learning at the document level.

In order to obtain a continuous representation of a document d, we simply use the average embedding of words in d as de:

$$de = \frac{1}{|d|} \sum_{t \in d} e_t. \quad (8)$$

We feed de into our model to predict the sentiment probability:

$$p(c|de) = softmax(\theta_d \cdot de + b_d). \quad (9)$$

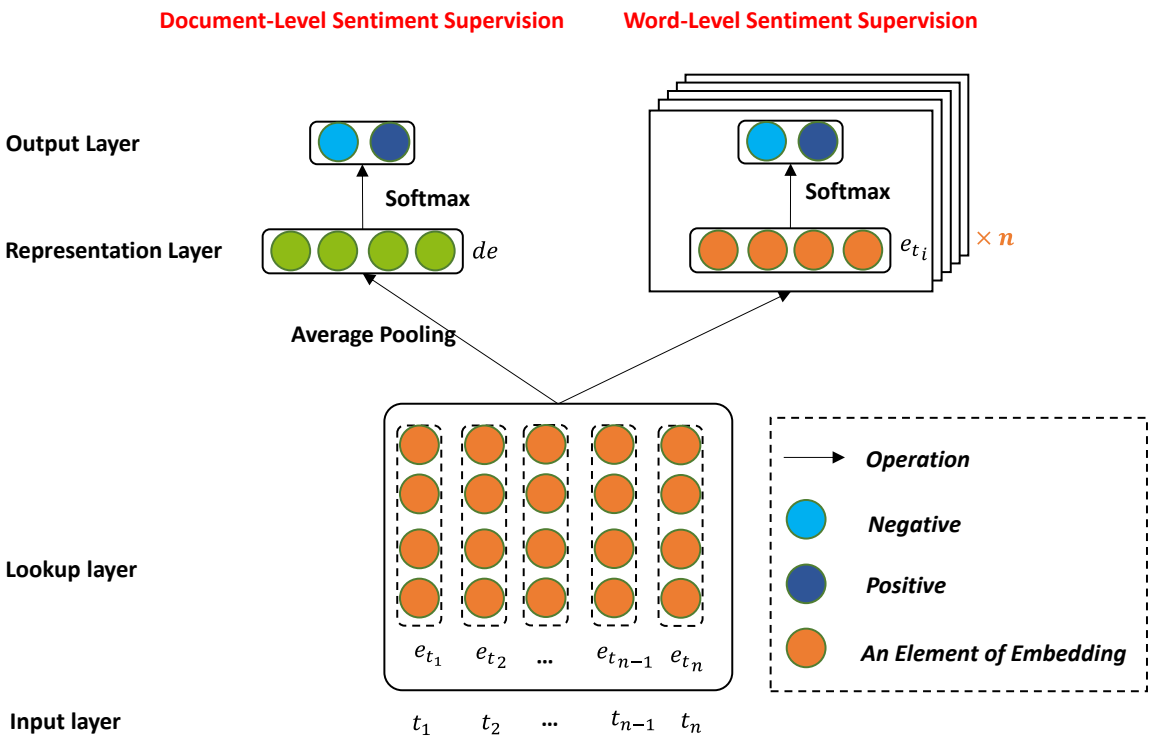

Figure 1: The Architecture of our Neural Network. Given a document d, represented as $[t_1, t_2, \ldots, t_n]$. t_i is the i-th word in d. And e_{t_i} represents the embedding of the word t_i. We take de, the average embedding of $[e_{t_1}, e_{t_2}, \ldots, e_{t_n}]$, as the representation of document d. We get each embedding of words in d as input to predict its sentiment polarities. We also take de as input to predict the sentiment for document d one time per epoch.

Similarly, the cost function is defined as average cross entropy that measures the difference between the sentiment distribution predicted in our model and the sentiment annotation at the document level:

$$f_{doc} = -\frac{1}{N}\sum_{k=1}^{N}\sum_{c\in\{+,-\}} \hat{p}(c|d_k)\log p(c|de_k) \tag{10}$$

where $\hat{p}(c|d_k)$ is the sentiment annotation of document d_k. $\hat{p}(c|d_k) = 1$ denotes the class label of d_k is positive, otherwise $\hat{p}(c|d_k) = 0$.

(3) Word and Document-Level Joint Learning

In order to learn the sentiment-aware word representation at both word and document levels, we integrate the cost function of two levels in a weighted combination way. The final cost function is defined as follows:

$$f = \alpha f_{word} + (1-\alpha) f_{doc} \tag{11}$$

where α is a tradeoff parameter$(0 \leq \alpha \leq 1)$. The weight of f_{word} can be increased by choosing a lager value of α.

We train our neural model with stochastic gradient descent and use AdaGrad (Duchi et al., 2011) to update the parameters.

3.3 From Sentiment Representation to Sentiment Lexicon

In this part, we follow the method proposed by Tang et al. (2014a) to build a classifier to convert the sentiment-aware word representation learned in Section 3.2 to a sentiment lexicon. The word representation is the input of the classifier and word sentiment polarity is the output.

Firstly, we utilize the embedding of 125 positive and 109 negative seed words manually labelled by Tang et al. (2014a) as training data[1].

Secondly, a variant-KNN classifier is also applied to extending the seed words on a web dictionary called Urban Dictionary. Unlike (Tang et al.,

[1]http://ir.hit.edu.cn/ dytang/paper/14coling/data.zip

Dataset	$\#pos$	$\#neg$	$Total$
SemEval2013-train	3632	1449	5081
SemEval2013-dev	482	282	764
SemEval2013-test	1474	559	2033
SemEval2014-test	982	202	1184
SemEval2015-test	1038	365	1403
SemEval2016-test	7059	3231	10290

Table 2: Statistics of Evaluation Datasets

2014a), we did not extend the neutral words.

Thirdly, a traditional logistic regression classifier is trained by using the embeddings of extended sentiment words as the inputs. The sentiment score of a word is the difference between its positive and negative probabilities.

Finally, the sentiment lexicon can be collected by using the classifier to predict the other words' sentiment score.

4 Experiment Study

4.1 Datasets and Settings

We utilize the public distant-supervision corpus[2] (Go et al., 2009) to learn our lexicons. We set M, the dimension of embedding, as 50. The learning rate is 0.3 for stochastic gradient descent optimizer. We tune the hyper-parameter α in the training process.

We evaluate the sentiment lexicons in both supervised and unsupervised sentiment classification tasks, on the SemEval 2013-2016 datasets. The statistics of evaluation datasets are shown in Table 2.

Supervised Sentiment Classification Evaluation: To evaluate the effect of the sentiment lexicon in supervised sentiment classification, we report the supervised sentiment classification performance by using some pre-defined lexicon features. We follow (Mohammad et al., 2013) to extract the lexicon features as follows:

- Total count of words in the tweet score of which is greater than 0;
- Total count of words in the tweet score of which is less than 0;
- The sum of scores for all word great than 0;
- The sum of scores for all word less than 0;
- The max score greater than 0;
- The min score less than 0;
- Non-zero score of the last positive word in the tweet;
- Non-zero score of the last negative word in the tweet.

We report the performance of SVM by using these lexicon features. The LIBSVM [3] toolkit is used with a linear kernel and the penalty parameter is set as the default value. The metric is F_1 score.

Unsupervised Sentiment Classification Evaluation: For unsupervised sentiment classification, we sum up the scores of all sentiment words in the document, according to the sentiment lexicon. If the sum is greater than 0, the document will be considered as positive, otherwise negative. The unsupervised learning evaluation metric is accuracy.

4.2 (External) Comparison with Public Lexicons

We compare our HSSWE method with four sentiment lexicons generated by the related work proposed in recent years:

- **Sentiment140** was constructed by Mohammad et al. (2013) on tweet corpus based on PMI between each word and the emoticons.
- **HIT** was constructed by Tang et al. (2014a) with a representation learning approach.
- **NN** was constructed by Vo and Zhang (2016) with a neural network method.
- **ETSL** refers to SemEval-2015 English Twitter Sentiment Lexicon[4] (Rosenthal et al., 2015; Kiritchenko et al., 2014), which is done using Best-Worst Scaling.

Note that Tang et al. (2014a), Vo and Zhang (2016) used incomplete dataset of SemEval2013 in their papers. For fair comparison, we conduct

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

[3]http://www.csie.ntu.edu.tw/ cjlin/libsvm

[4]http://www.saifmohammad.com/WebDocs/lexiconstoreleaseonsclpage/SemEval2015-English-Twitter-Lexicon.zip

Lexicon	Semeval2013	Semeval2014	Semeval2015	Semeval2016	Average
Sentiment140	0.7317	0.7271	0.6917	0.6809	0.7079
HIT	0.7181	0.6947	0.6797	0.6928	0.6963
NN	0.7225	0.7115	***0.6970***	0.6887	0.7049
ETSL	0.7104	0.7090	0.6650	0.6862	0.6926
HSSWE	***0.7550***	***0.7424***	0.6921	***0.7097***	***0.7248***

Table 3: Supervised Evaluation for External Comparison (F_1 Score)

Lexicon	Semeval2013	Semeval2014	Semeval2015	Semeval2016	Average
Sentiment140	0.7208	0.7416	0.6935	0.6928	0.7122
HIT	0.7566	0.7922	0.7128	0.7282	0.7474
NN	0.6903	0.7280	0.6507	0.6585	0.6819
ETSL	0.7675	0.8226	0.7505	***0.7365***	0.7693
HSSWE	***0.7734***	***0.8539***	***0.7669***	0.7206	***0.7787***

Table 4: Unsupervised Evaluation for External Comparison (Accuracy)

all the comparison experiments on the complete benchmark datasets.

Supervised Sentiment Classification: We first report the supervised sentiment classification F_1 score of five compared methods on the Semeval 2013-2016 datasets in Table 3. It can be seen that our HSSWE method gets the best result on all four datasets. It outperforms Sentiment140, HIT, NN and ETSL 1.7, 2.8, 1.9, and 3.2 percentages on the average of four datasets. The improvements are significant according to the paired t-test.

Unsupervised Sentiment Classification: We then report the unsupervised sentiment classification accuracy of five methods on the Semeval 2013-2016 datasets in Table 4. In can be seen that HSSWE obtains the best performance on Semeval 2013-2015. On the Semeval 2016 dataset, it is slightly lower than ETSL. Across four datasets, the average accuracy of HSSWE is 6.6, 3.1, 9.6 and 0.94 higher than Sentiment140, HIT, NN and ETSL, respectively.

4.3 (Internal) Comparison within the Model

In order to further verify the effectiveness of our method and analyze which part of our model contributes the most, we carried out the internal comparison within our model. We design the following two simplified versions of our model for comparison:

- **PMI-SO** denotes a PMI-SO based sentiment lexicon with soft sentiment annotation learned in Section 3.1.

- **Doc-Sup** denotes the neural network system with only document-level sentiment supervision. It equals to HSSWE when $\alpha = 0$.

Actually, HSSWE can be viewed as a "combination" of PMI-SO and Doc-Sup. In Tables 5 and 6, we report the comparison results on supervised and unsupervised sentiment classification respectively.

Supervised Sentiment Classification: As is shown in Table 5, two basic models PMI-SO and Doc-Sup show similar performance. They have distinct superiority across different datasets. But both are significantly lower than HSSWE. It shows that by combing the supervision at both document and word levels, it can indeed improve the quality of sentiment-aware word embedding and the subsequent sentiment lexicon.

Unsupervised Sentiment Classification: As is shown in Table 6, the conclusions are similar with that in supervised sentiment classification: HSSWE achieves the significantly better performance.

4.4 Word-level Sentimnt Annotation: Hard vs. Soft

In Section 3.1, we introduce two kinds of word-level sentiment annotation, i.e., soft and hard sentiment annotation. We now compare two methods. The results are reported in Tables 5 and 6. It can be seen that for supervised evaluation, HSSWE (soft) and HSSWE (hard) yield comparative performance. HSSWE (hard) has slight superiority over HSSWE (hard) in Semeval 2013, 2014 and 2016, but HSSWE (hard) is better on Semeval2015. In contrast, for unsupervised evaluation, HSSWE (soft) is significantly better than

Lexicon	Semeval2013	Semeval2014	Semeval2015	Semeval2016	Average
PMI-SO	0.7265	0.7333	0.7008	0.6858	0.7116
Doc-Sup	0.7326	0.7302	0.6814	0.6986	0.7107
HSSWE (soft)	***0.7550***	***0.7424***	0.6921	***0.7097***	***0.7248***
HSSWE (hard)	0.7503	0.7383	***0.7020***	0.7061	0.7242

Table 5: Supervised Evaluation for Internal Comparison (F_1 Score), where HSSWE (hard) and HSSWE (soft) utilize the PMI-SO lexicon with hard sentiment annotation and soft sentiment annotation at the word level, respectively.

Lexicon	Semeval2013	Semeval2014	Semeval2015	Semeval2016	Average
Doc-Sup	0.7252	0.8294	0.7391	0.6859	0.7449
HSSWE (soft)	***0.7734***	***0.8539***	***0.7669***	***0.7206***	***0.7787***
HSSWE (hard)	0.7418	0.8395	0.7633	0.7011	0.7614

Table 6: Unsupervised Evaluation for Internal Comparison (Accuracy)

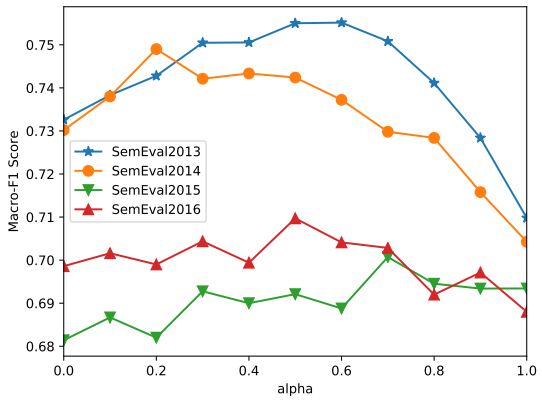

Figure 2: HSSWE (soft) with different α

HSSWE (hard). The average improvement is 1.7 percentage.

4.5 Tuning the Parameter α

In this section, we discuss the tradeoff between two parts of supervisions by turning the tradeoff parameter α. When α is 0, HSSWE only benefits from the document-level sentiment supervision and when α is 1, HSSWE benefits from only word-level sentiment supervision. We observe that HSSWE performs better when α is in the range of [0.45,0.55]. By integrating two component parts of sentiment supervision, HSSWE has significant superiority over that learned from either one.

4.6 Lexicon Analysis

In order to gain more insight of our model and observe the effectiveness of the sentiment lexicon, in Table 7 we extract the positive sentiment score of some representative words learned by different methods. The positive scores are supposed to be: *best*>*better*>*well*. HSSWE captures such comparative sentiment strength but PMI-SO does not. We further observe that in many cases where the results of PMI-SO and Doc-Sup are inconsistent (e.g., Doc-Sup incorrectly predicts *"unreasonable"*, *"boredddddd"* and *"sickkk"* as positive words, but PMI-SO predicts them correctly; PMI-SO incorrectly predicts *"fit"* but Doc-Sup predicts it correctly.), HSSWE often yield the correct results. It shows the advantages of hierarchical sentiment supervision. HSSWE can also correct the sentiment prediction where both PMI-SO and Doc-Sup are inefficient (e.g., *"overplayed"*).

Words	HSSWE	PMI-SO	Doc-Sup
well	0.5740	0.5430	0.7898
better	0.5837	*0.5358 (F)*	0.8440
best	0.9455	0.6823	0.9639
fit	0.2894	*-0.5594 (F)*	0.6076
unreasonable	-0.2441	-0.8421	*0.5275 (F)*
boredddddd	-0.1137	-0.7142	*0.4843 (F)*
sickkkk	-0.3892	-0.7692	*0.1323 (F)*
overplayed	-0.1390	*0.5000 (F)*	*0.8448 (F)*

Table 7: Sentiment Lexicon Analysis, where score with *(F)* means falsely predicted polarity or strength.

5 Conclusion

In this paper, we proposed to construct sentiment lexicons based on a sentiment-aware word representation learning approach. In contrast to traditional methods normally learned based on only the document-level sentiment supervision. We proposed word representation learning via hierarchical sentiment supervision, i.e., under the supervi-

sion at both word and document levels. The word-level supervision can be provided based on either predefined sentiment lexicons or the learned PMI-SO based sentiment annotation of words. A wide range of experiments were conducted on several benchmark sentiment classification datasets. The results indicate that our method is quite effective for sentiment-aware word representation, and the sentiment lexicon generated by our approach beats the state-of-the-art sentiment lexicon construction approaches.

Acknowledgements

The work was supported by the Natural Science Foundation of China (No. 61672288), and the Natural Science Foundation of Jiangsu Province for Excellent Young Scholars (No. BK20160085).

References

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

John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul):2121–2159.

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

William L. Hamilton, Kevin Clark, Jure Leskovec, and Dan Jurafsky. 2016. Inducing domain-specific sentiment lexicons from unlabeled corpora. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 595–605.

Vasileios Hatzivassiloglou and Kathleen R McKeown. 1997. Predicting the semantic orientation of adjectives. In *Proceedings of the eighth conference on European chapter of the Association for Computational Linguistics*, pages 174–181.

Bas Heerschop, Alexander Hogenboom, and Flavius Frasincar. 2011. Sentiment lexicon creation from lexical resources. In *International Conference on Business Information Systems*, pages 185–196.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the Tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177.

Jaap Kamps, Maarten Marx, Robert J Mokken, Maarten De Rijke, et al. 2004. Using wordnet to measure semantic orientations of adjectives. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation*, volume 4, pages 1115–1118.

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

Svetlana Kiritchenko, Xiaodan Zhu, and Saif M Mohammad. 2014. Sentiment analysis of short informal texts. *Journal of Artificial Intelligence Research*, 50:723–762.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *Proceedings of Workshop at 1st International Conference on Learning Representations (ICLR2013)*.

Saif M. Mohammad, Svetlana Kiritchenko, and Xiaodan Zhu. 2013. Nrc-canada: Building the state-of-the-art in sentiment analysis of tweets. In *Proceedings of the seventh international workshop on Semantic Evaluation Exercises (SemEval-2013)*, pages 321–327.

Sara Rosenthal, Preslav Nakov, Svetlana Kiritchenko, Saif Mohammad, Alan Ritter, and Veselin Stoyanov. 2015. Semeval-2015 task 10: Sentiment analysis in twitter. In *SemEval@ NAACL-HLT*, pages 451–463.

Duyu Tang, Furu Wei, Bing Qin, Ming Zhou, and Ting Liu. 2014a. Building large-scale twitter-specific sentiment lexicon : A representation learning approach. In *Proceedings of the 25th International Conference on Computational Linguistics (COLING 2014)*, pages 172–182.

Duyu Tang, Furu Wei, Nan Yang, Ming Zhou, Ting Liu, and Bing Qin. 2014b. Learning sentiment-specific word embedding for twitter sentiment classification. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 1555–1565.

Peter D Turney. 2002. Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 417–424.

Peter D Turney. 2008. The latent relation mapping engine: Algorithm and experiments. *Journal of Artificial Intelligence Research*, 33:615–655.

Duy Tin Vo and Yue Zhang. 2016. Dont count, predict! an automatic approach to learning sentiment lexicons for short text. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, volume 2, pages 219–224.

Ke Wang and Rui Xia. 2016. A survey on automatical construction methods of sentiment lexicons. *Acta Automatica Sinica*, 42(4):495–511.