The Thirty-Second AAAI Conference
on Artificial Intelligence (AAAI-18)

Learning Sentiment-Specific Word Embedding via Global Sentiment Representation

Peng Fu,[1,2] **Zheng Lin,**[1*] **Fengcheng Yuan,**[1,2] **Weiping Wang,**[1] **Dan Meng**[1]
Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China[1]
School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China[2]
{fupeng, linzheng, yuanfengcheng, wangweiping, mengdan}@iie.ac.cn

Abstract

Context-based word embedding learning approaches can model rich semantic and syntactic information. However, it is problematic for sentiment analysis because the words with similar contexts but opposite sentiment polarities, such as *good* and *bad*, are mapped into close word vectors in the embedding space. Recently, some sentiment embedding learning methods have been proposed, but most of them are designed to work well on sentence-level texts. Directly applying those models to document-level texts often leads to unsatisfied results. To address this issue, we present a sentiment-specific word embedding learning architecture that utilizes local context information as well as global sentiment representation. The architecture is applicable for both sentence-level and document-level texts. We take global sentiment representation as a simple average of word embeddings in the text, and use a corruption strategy as a sentiment-dependent regularization. Extensive experiments conducted on several benchmark datasets demonstrate that the proposed architecture outperforms the state-of-the-art methods for sentiment classification.

Introduction

Continuous word representation, commonly called word embedding, attempt to represent each word as a continuous, low-dimensional and real-valued vector. Since they can capture various dimensions of semantic and syntactic information and group words with similar grammatical usages and semantic meanings, they have less susceptible to data sparsity. Therefore, word embeddings are widely used for many natural language processing tasks, such as sentiment analysis (Wang et al. 2015), machine translation (Ding et al. 2017) and question answering (Hao et al. 2017).

Existing word embedding learning approaches mostly represent each word by predicting the target word through its context (Collobert and Weston 2008; Mikolov et al. 2013) and map words of similar semantic roles into nearby points in the embedding space. For example, '*good*' and '*bad*' on the left of the Figure 1 are mapped into close vectors in the embedding space. However, it is confusing for sentiment analysis, because these two words actually have opposite sentiment polarities. Therefore, it is desired to propose models that can not only capture the contexts of words but also model sentiment information of texts, like word embeddings on right of the Figure 1.

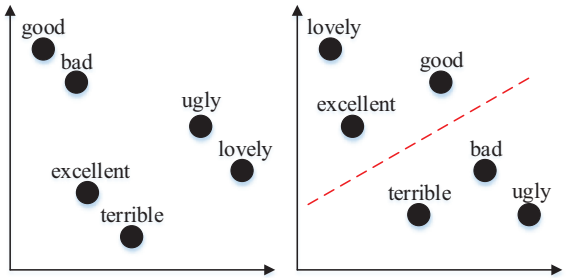


Figure 1: Illustrative normal word embedding (left) and sentiment-specific word embedding (right) in embedding space.

To achieve this goal, Tang et al. proposed two models based on the C&W(Collobert and Weston 2008) model that learn sentiment-specific word embedding by sentiment polarity labels for twitter sentiment classification. They also extended their work with several customized loss functions (Tang et al. 2016b). These models predict or rank sentiment polarity based on word embeddings in a fixed window of words across a sentence. In addition, based on the Skip-Gram (Mikolov et al. 2013), Zhang et al.(2015) integrated the sentiment information by using the semantic word embeddings in the context to predict the sentiment polarity through a $softmax$ layer, and Yang et al.(2017) proposed a model that predicted the target word and its label simultaneously. Both of them took sentiment information as a part of the local context. Due to the limitation of the design of these training methods, they could only be used in specific task and is less efficient for document-level text. Therefore, the integration of sentiment polarity into semantic word embeddings is still a major challenge for sentiment analysis.

In this paper, we will introduce a sentiment-specific word embedding learning architecture that incorporates local context with global sentiment representation. In general, the local context can be regarded as a representation of the target word, while the global sentiment representation is the averaged vector of the words in the text through a corruption

*Corresponding author

strategy. The strategy is a biased randomly sampling process. Thus, the local and the global representations could be regarded as semantic and sentiment information respectively. In order to learn sentiment-specific word embedding, the global sentiment representation could integrate into the local context by modeling jointly.

Based on the proposed architecture, we develop two neural network models to learn the sentiment-specific word embeddings, which are the extension to the Continuous Bag-of-Words (CBoW) model. The prediction model (SWPredict) takes sentiment prediction as a multi-class classification task, and it can be viewed as language modeling. The ranking model (SWRank) takes sentiment prediction as a ranking problem, and it penalizes relative distances among triplet global sentiment representations. Experiments demonstrate the effectiveness of our models, and empirical comparisons on sentence-level and document-level sentiment analysis tasks show that our architecture outperforms state-of-the-art methods.

The main contributions of this work are as follows:

- We propose a general architecture to learn sentiment-specific word embeddings, and use a global sentiment representation to model the interaction of words and sentiment polarity. The architecture is effective for both sentence-level and document-level texts.
- We develop two neural networks to learn sentiment-specific word embeddings. The prediction model takes the sentiment prediction as a classification task, and the ranking model takes sentiment prediction as a ranking problem among the triplets.
- To improve the efficiency of the model, we use a corruption strategy that favors informative words which have strong discrimination capability. It can be regarded as a sentiment-dependent regularization for global sentiment representation.

Background

Modeling Contexts of Words

Many methods can encode contexts of words into embeddings from a large collection of unlabeled data. Here we focus on the most relevant methods to our model. Bengio et al. proposed a neural language model and estimated the parameters of the network and these embeddings jointly. For this model is quite expensive to train, Mikolov et al.(2013) proposed the Word2vec, which contains CBoW and Skip-Gram models, to learn high-quality word embeddings.

CBoW is an effective framework for modeling contexts of words, which aims to predict the target word given its context in a sentence. It contains an input layer, a projection layer parameterized by the matrix U and an output layer parameterized by V. The probability of the target word w^t with its local context C^t can be calculated as:

$$P(w^t|\mathrm{C}^t) = \frac{\exp(\mathrm{V}_{w^t}^{\mathrm{T}}\mathrm{U}\mathrm{C}^t)}{\sum_{w' \in V} \exp(\mathrm{V}_{w'}^{\mathrm{T}}\mathrm{U}\mathrm{C}^t)} \quad (1)$$

Document Representation

Document representation is a fundamental problem for many natural language processing tasks. Many efforts have been done to generate concise document representation. Paragraph Vectors (Dai, Olah, and Le 2015) is an unsupervised method that explicitly learns a document representation with word embeddings. In the Paragraph Vectors model, a projection matrix D is introduced. Each column of matrix D is a document representation x. The model inserts x to the standard language model which aims at capturing the global semantic information of the document. With the document representation x, the probability of the target word w^t given its local context C^t is calculated as:

$$P(w^t|\mathrm{C}^t, \mathrm{x}) = \frac{\exp(\mathrm{V}_{w^t}^{\mathrm{T}}(\mathrm{U}\mathrm{C}^t + \mathrm{x}))}{\sum_{w' \in V} \exp(\mathrm{V}_{w'}^{\mathrm{T}}(\mathrm{U}\mathrm{C}^t + \mathrm{x}))} \quad (2)$$

However, the complexity of Paragraph Vectors grows with the size of vocabulary and training corpus, and it needs expensive inference to obtain the representations of unseen documents. To alleviate these problems, Chen (2017) proposed a model, called Doc2VecC, which simply represents a document as an average of word embeddings that are randomly sampled from the document. The randomly sampling process is a kind of drop-out corruption that can speed up the training. What's more, the corruption strategy is proved to be as a data-dependent regularization.

Given a document D contains word embeddings $\{\mathrm{w}_1, \ldots, \mathrm{w}_T\}$, its global representation is denoted as x and each word embedding is denoted as x. The corruption strategy randomly overwrites each word embedding of the original document x with probability q, and it sets the uncorrupted word embeddings to $\frac{1}{1-q}$ times the value of its originals. Formally,

$$\widetilde{x}_d = \begin{cases} 0, & \text{with probability } q \\ \frac{x_d}{1-q}, & \text{otherwise} \end{cases} \quad (3)$$

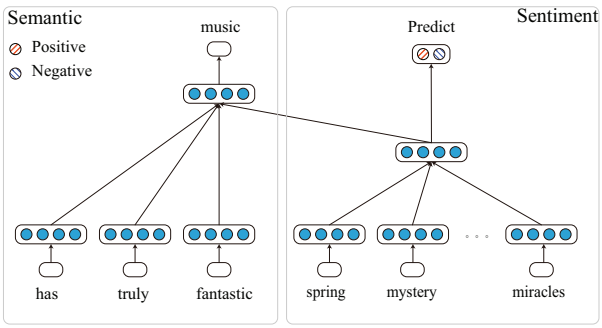
Thus, the corrupted document representation is denoted as $\mathrm{x} = \frac{1}{T}\sum_1^T \widetilde{x}_d$, where T is the length of the document. Finally, the calculation of the probability of the target word w^t is the same as Eq.2.

Approach

Architecture

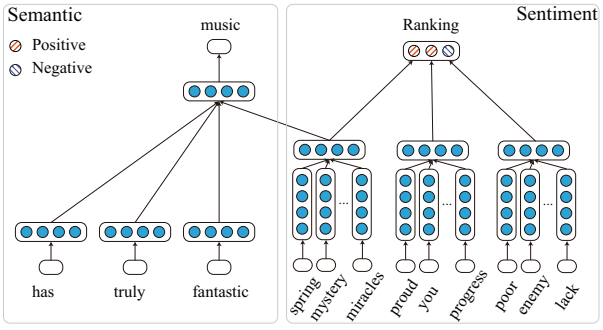
An intuitive solution to model the interaction of sentiment information and word embeddings is to predict the sentiment distribution of the global representation while modeling the contexts of words. The benefit of introducing global representation is that it can learn sentiment-specific word embeddings from variable-length texts. In this paper, we propose an architecture that is an extension of the CBoW model (Mikolov et al. 2013). Based on the architecture, we develop two neural networks to learn sentiment-specific word embeddings, including a prediction model and a ranking model. The architecture consists of two components:

- The semantic component is to learn semantic and syntactic information of words in an unsupervised way as shown in the background section.



(a)

Figure 2: The architecture of the prediction model. It takes sentiment prediction as a multi-class classification problem.



(b)

Figure 3: The architecture of the ranking model. It takes sentiment prediction as a ranking problem among triplets.

- The sentiment component is to model global sentiment representation of text in a supervised manner, which introduces a corruption strategy as sentiment-dependent regularization that will describe below.

Corruption as Sentiment-dependent Regularization

We observe that the frequencies of many sentiment words are lower than the commonly used words in most cases. Consequently, we employ a biased drop-out corruption, which can regarded as sentiment-dependent regularization (Chen 2017) for global text representation.

Specifically, we randomly overwrite each word embedding of the original text x with probability p. The probability p is calculated as a corruption by the frequency of word w_i.

$$p = 1 - (\sqrt{\frac{\alpha}{freq(w_i)}} + \frac{\alpha}{freq(w_i)}) \tag{4}$$

where α is a threshold, we use $1e-4$ in this paper, and $freq(w_i)$ is the frequency of word w_i. We set the uncorrupted word embeddings to $\frac{1}{1-q}$ times the value of its originals as (Chen 2017). Thus, the global text representation is calculated as:

$$\widetilde{x}_d = \begin{cases} 0, & \text{with probability } p > 0 \\ \frac{1}{1-q} \cdot x_d, & \text{otherwise} \end{cases} \tag{5}$$

and $\widetilde{\mathrm{x}} = \sum_1^T \widetilde{x}_d$. Therefore, the probability of the target word w^t, given its local context C^t as well as the text representation $\widetilde{\mathrm{x}}$, is calculated as:

$$P(w^t|\mathrm{C}^t, \widetilde{\mathrm{x}}) = \frac{\exp(\mathrm{V}_{w^t}^{\mathrm{T}}(\mathrm{UC}^t + \frac{1}{T}\mathrm{U}\widetilde{\mathrm{x}}))}{\sum_{w' \in V} \exp(\mathrm{V}_{w'}^{\mathrm{T}}(\mathrm{UC}^t + \frac{1}{T}\mathrm{U}\widetilde{\mathrm{x}}))} \tag{6}$$

where T is the length of the text. Given the training corpus $D = \{D_1, \ldots, D_n\}$, the parameters U and V are learned to minimize the loss:

$$J_S = -\sum_{i=1}^{n}\sum_{t=1}^{T_i} f(w_i^t, \mathrm{C}_i^t, \widetilde{\mathrm{x}}_i^t) \tag{7}$$

Prediction Model

The basic idea of the prediction model is to take sentiment prediction as a multi-class classification problem. It predicts positive or negative categorical probabilities of the global sentiment representation.

As illustrated in Figure 2, it consists of two components, and each component contains an input layer, a projection layer, and an output layer. The inputs of the semantic component are word tokens, while the inputs of the sentiment component are the sampled word tokens by the corruption strategy. In the semantic component, the probability of the target word is calculated as Eq.6. While in the sentiment component, the projection layer feeds the corrupted global text representation calculated as Eq.5 to a linear layer, and converts the vector length to category number, which is 2 in this paper. Then, the output layer generates conditional probabilities over positive and negative categories.

Given the gold sentiment polarity of the input texts in the corpus D, we use $f^g(t) = [1, 0]$ as the positive polarity, and $f^g(t) = [0, 1]$ as the negative polarity. The cross entropy error between gold sentiment distribution and predicted distribution of the output layer is:

$$J_{predit} = -\sum_{d}^{D}\sum_{k=\{0,1\}} f_k^g(t) \cdot log(f_k^{pred}(t)) \tag{8}$$

To get sentiment-specific word embeddings, we combine the losses from the semantic and the sentiment components together. The final loss function is Eq.9, where β weights the two components.

$$J_{SP} = \beta \cdot J_S + (1-\beta) \cdot J_{predict} \tag{9}$$

Ranking Model

Sentiment-specific word embedding learning needs a large scale of training data. However, it is hard to obtain abundant dataset with carefully labeled sentiment polarity. In view of well labeled texts, it is easier to obtain weakly labeled texts from ratings (movie reviews) or emoticons (tweets), without much manual work. However, the weakly labeled texts may contain wrong labels, which will influence the quality of the learned sentiment-specific word embedding. To alleviate the

influence, the texts belonging to the same sentiment polarity should be as close as possible, while the texts of different sentiments should be kept far away. Therefore, we propose a ranking model based on triplets.

As Figure 3 shows, it also consists of two components. The semantic component is the same as the prediction model, while the sentiment component generates valid ranking triplets.

A valid triplet is generated as follows: given the subset P contains positive texts and the subset N contains negative texts. Let's take P as focus, two global text representations $\widetilde{x}_1$ and $\widetilde{x}_2$ are sampled from P, and a global text representation $\widetilde{x}_3$ is sampled from N. The case for N as the focus is a mirror case. Before the output layer, we use a nonlinear layer to convert global representations to sentiment scores y_i.

$$\mathrm{y}_i = f(\mathrm{w}\widetilde{\mathrm{x}} + \mathrm{b}) \tag{10}$$

where w and b are parameters, $f(\cdot)$ is a sigmoid function.

The training objective is defined as the following ranking loss:

$$J_{rank} = \sum_{\mathrm{d}=<\widetilde{\mathrm{x}}_1,\widetilde{\mathrm{x}}_2,\widetilde{\mathrm{x}}_3>}^{D} \max(0, \lambda - dst(\widetilde{\mathrm{x}}_1, \widetilde{\mathrm{x}}_3) + dst(\widetilde{\mathrm{x}}_1, \widetilde{\mathrm{x}}_2)) \tag{11}$$

where λ is the margin parameter, $dst(\cdot)$ is the distance between global text representations, and $< \widetilde{\mathrm{x}}_1, \widetilde{\mathrm{x}}_2, \widetilde{\mathrm{x}}_3 >$ denotes a valid triplet. This means we require the distance between $< \widetilde{\mathrm{x}}_1, \widetilde{\mathrm{x}}_2 >$ to be closer than that between $< \widetilde{\mathrm{x}}_1, \widetilde{\mathrm{x}}_3 >$ by at least λ.

The distance is calculated as the difference scores between the two representations.

$$dst(\widetilde{\mathrm{x}}_\mathrm{i}, \widetilde{\mathrm{x}}_\mathrm{j}) = |\mathrm{y}_\mathrm{i} - \mathrm{y}_\mathrm{j}| \tag{12}$$

Compared with pairwise ranking, the triplet-based ranking can easily make representations with the same sentiment polarity close to one another, while keep representations with different sentiment polarity away. Because the majority of the representations are with correct labels, they would gather close to each other in the training process. While, representations with wrong labels would go towards the opposite cluster, but with slower speeds compared with pairwise ranking(Guan et al. 2016).

The final loss function is the combination of the losses from the semantic and the sentiment components:

$$J_{ST} = \beta \cdot J_S + (1 - \beta) \cdot J_{rank} \tag{13}$$

where β weights the two components.

Training

We use two datasets for training the sentiment-specific word embeddings separately. One is movie reviews that is considered as document-level texts, while the other is tweets that is regarded as sentence-level texts. The movie reviews are extracted from the SAR14(Nguyen et al. 2014) dataset that contains 233,600 IMDB reviews along with their associated ratings on a 1-10 scale. We use all reviews with scores ≤ 4 as the negative texts, and randomly select the same amount of reviews with scores ≥ 7 as the positive texts. We extract tweets from a collection of dataset[1] that labeled positive or negative, and filter the tweets that less than 7 words. The statistics of the datasets are given in Table 1.

Dataset	#Vocab.	Len$_{avg}$	#Pos.	#Neg.
Reviews	41,778	283.3	66,000	66,222
Tweets	34,165	16.7	637,728	665,432

Table 1: Statistics of the training datasets for sentiment-specific word embedding learning.

For our models, we use AdaGrad (John Duchi 2011) to update the parameters and the learning rate is 1.0. We empirically set the context window size as 3 and batch size as 128. We set the hyper-parameter $\alpha = 1e-4$ and $p = 0.9$ for corruption. We evaluate the effect of the embedding size and choose 150 for both models. Our models are implemented in tensorflow[2].

Experiments

We evaluate our learned embeddings by taking them as features for sentence-level and document-level sentiment classification. We use the LIBLINEAR (Fan et al. 2008) as the classifier.

Datasets

For document-level sentiment classification, we use the IMDB movie review dataset (Maas et al. 2011). It consists of 100,000 movie reviews, and half of them are labeled as either positive or negative. We use the default train/test split, and randomly select 10% of the training data as the development set.

For sentence-level sentiment classification, we use the Twitter dataset from the SemEval 2013 task 2 (Nakov et al. 2013). The data can be downloaded by running a script[3]. We build a 2-class Twitter sentiment classifier. Thus, we only use positive and negative data. The statistics of our datasets are given in Table 2.

Dataset	Subset	#Positive	#Negative	#Total
IMDB	Train	11,250	11,250	22,500
	Dev	1,250	1,250	2,500
	Test	12,500	12,500	25,000
Twitter	Train	2978	1162	4,140
	Dev	328	170	498
	Test	1306	485	1,791

Table 2: Statistics of the IMDB and the Twitter datasets for sentiment classification.

[1] thinknook.com/twitter-sentiment-analysis-training-corpus-datasets-2012-09-22

[2] www.tensorflow.org

[3] https://cs.york.ac.uk/semeval-2013/task2/index.html

IMDB Dataset						
Method	Word2vec	Glove	SE-Pred	SE-HyRank	SWPredict	SWRank
Avg.	86.51 ± 0.02	84.43 ± 0.01	83.62 ± 0.02	84.13 ± 0.03	88.31 ± 0.03	$\mathbf{88.55 \pm 0.05}$
Conv.	87.68 ± 0.02	86.0 ± 0.01	84.73 ± 0.04	85.82 ± 0.02	88.22 ± 0.04	$\mathbf{88.39 \pm 0.04}$
Twitter Dataset						
Method	Word2vec	Glove	SE-Pred	SE-HyRank	SWPredict	SWRank
Avg.	77.39 ± 0.02	79.23 ± 0.08	80.9 ± 0.02	81.2 ± 0.03	$\mathbf{84.05 \pm 0.04}$	84.02 ± 0.02
Conv.	77.53 ± 0.02	79.74 ± 0.13	81.8 ± 0.08	82.2 ± 0.02	$\mathbf{83.73 \pm 0.04}$	82.69 ± 0.03

Table 3: Accuracy of sentiment classification on the IMDB and the Twitter datasets. "Avg." denotes the average of the word embeddings in the text, which are features of the SVM classifier. "Conv." denotes the concatenation of vectors derived from $max, min, average$ convolutional layers, which are features of the SVM classifier. We mark the best results by boldface.

Setup

We apply word embeddings to sentiment classification under a supervised learning framework. Given the learned word embeddings matrix U, we represent each global sentiment representation as an average of the word embeddings in the text as Chen(2017) did:

$$z(x) = \frac{1}{T} \sum_{w \in D} \mathrm{U}_w$$

We also adopt the $max, min, average$ convolutional layers to represent the global sentiment representation (Socher et al. 2011; Tang et al. 2014). The representation is regarded as the concatenation of the vectors derived from different convolutional layers.

$$z(x) = [z_{max}(\widetilde{x}), z_{min}(\widetilde{x}), z_{avg}(\widetilde{x})]$$

where $z(x)$ is the representation of text x. Taking $z(x)$ as features, we use Support Vector Machine to build sentiment classifiers. We train them on the training set, and tune parameters and evaluate the models on the development and the test datasets respectively.

Baseline Methods

We compare our models with the following baseline word embedding learning methods:

- Word2vec[4]: Mikolov et al. developed a widely used toolkit "Word2vec" that contains CBoW and Skip-Gram algorithms to learn word embedding. We use CBoW in the experiments and train it with negative sampling.
- Glove[5]: Pennington et al.(2014) released a popular algorithm for obtaining word embedding. It is a log bilinear model that use AdaGrad (John Duchi 2011) to minimize a weighted square error on global co-occurrence counts. The method is comparable to Word2vec.
- SE-Pred: A sentiment-specific word embedding learning model based on C&W(Collobert and Weston 2008). It regards sentiment prediction as a multi-classification task which proposed by (Tang et al. 2016b). It has high performance for twitter sentiment classification.
- SE-HyRank: A sentiment-specific word embedding learning model based on ranking loss that proposed by Tang et al. It learns context information and sentiment information simultaneously through a neural network with a pairwise ranking loss. It is the latest model that learns sentiment word embeddings.

Here we focus on the comparisons of the model architectures. For a fair comparison, we train all competing methods on the same datasets using a context window of three words. For the baseline methods, we use default settings in the provided implementations or described as their papers.

Sentiment Classification Results

Table 3 shows the performance of the learned word embedding on both sentence-level and document-level sentiment classification tasks.

On the IMDB dataset, SWPredict and SWrank outperform all baselines significantly, especially SWRank achieves the best result. It denotes that our models learn a better word embeddings for document-level sentiment classification. Compared with SE-Pred and SE-HyRank, Word2vec and Glove get higher results. The underlying reason is that SE-Pred and SE-HyRank predict or rank sentiment polarity based on word embeddings in a fixed window of words across a sentence, which is not appropriate for document-level texts. We address this problem by encoding sentiment information from global text representation. SWRank achieves a better result than SWPredict indicates the effectiveness of ranking among triplets for document-level sentiment classification with weakly labeled texts.

On the Twitter dataset, sentiment-specific word embedding learning models obtain better performance, which shows the importance of sentiment information for sentiment classification. Our models get higher results than SE-Pred and SE-HyRank, which implies that our models can capture the most distinguishable information when learning sentiment-specific word embeddings. The main difference between our models and SE-Pred as well as SE-HyRank is that we encode sentiment information from global text representation and use a corruption strategy as a sentiment-dependent regularization. That makes our models can learn more discriminable word embeddings. One of the differences between SWRank and SE-HyRank is that we use a

[4]https:code.google.com/p/word2vec/

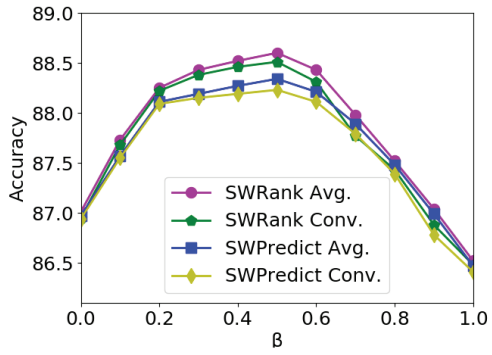
[5]http://nlp.stanford.edu/projects/glove/

triplet rather than a pairwise in the ranking loss. The results show that triplet-based ranking is more effective than pairwise ranking for sentence-level sentiment classification.

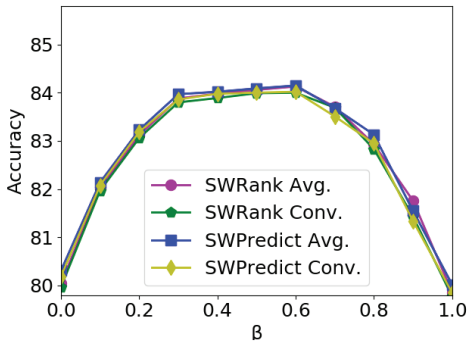
For our models, the accuracy of the average word embeddings as features is slightly higher than the concatenation of the outputs of convolutional layers as features. It indicates that our models can learn the most discriminative word embeddings and penalize the common or non-discriminative word embeddings through the corruption strategy.

Effect of β in the models

In this experiment, we investigate the effect of β on the classification performance with different β on the evaluation datasets. The parameter β weights the semantic and sentiment information. A small value of β means the models pay more attention on sentiment information.



(a) IMDB



(b) Twitter

Figure 4: Effect of the parameter β with different values on the sentiment classification performance on the IMDB and the Twitter datasets.

As shown in Figure 4, for the IMDB dataset, SWPredict and SWRank obtain best results when setting β as 0.5 on both global representation methods. While, for the Twitter dataset, both models get best results when β is 0.6. This means both models perform better when β is in the range of [0.5,0.6], which balances the semantic and sentiment information. We can see a sharp decline when β is 1, which indicates the essential role of sentiment information for sentiment classification.

Effect of λ in the Ranking Model

The margin parameter λ controls the distance between positive and negative texts. Here, we investigate the effect of λ on the classification performance. We change λ from 1 to 10, and the performance results are shown in Figure 5.

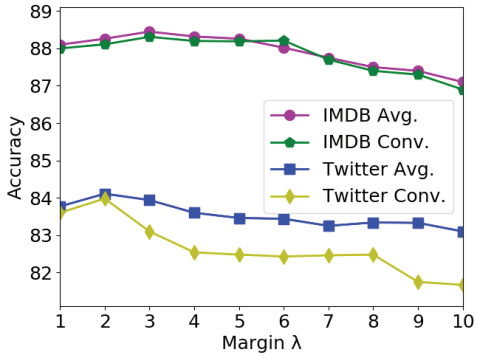


Figure 5: Effect of margin λ with different values on the sentiment classification performance on the IMDB and the Twitter datasets of the SWRank model.

For the IMDB dataset, both global representation methods obtain the best results when λ is 3, and drop steadily after that. For the Twitter dataset, the best results are achieved at $\lambda = 2$, and the performance of Conv. method drops quickly. Moreover, when λ is set to a large value, the networks are more easily to be trapped in saturating regions (Bengio et al. 2013) after long time training. In addition, the overall training time will increase when enlarging the margin λ. Therefore, we set $\lambda = 3$ on the movie reviews sentiment-specific word embedding training dataset and $\lambda = 2$ on the tweets sentiment-specific word embedding training dataset.

Comparison to Document Representation

Model	Accuracy %
Paragraph Vectors†	87.9
Skip-Thought Vectors†	82.6
Doc2VecC†	88.3
SWPredit	89.4
SWRank	89.6

Table 4: Comparison with previous work on document representation for sentiment classification. †denotes the results cited from (Chen 2017).

We compare the performance of our models to Paragraph Vectors (Dai, Olah, and Le 2015), Skip-Thought Vectors (Kiros et al. 2015) and Doc2VecC (Chen 2017). We directly cite the results from Chen(2017), for we both use SVM classifiers to evaluate our models on the same IMDB dataset (Maas et al. 2011). The main difference is that we only use the labeled training set that contains 25,000 reviews, while they also use the unlabeled set that contains 50,000 reviews. Table 4 shows that our models obtain slightly higher results

with less data. The reason is that the other document representation methods do not model the sentiment polarity information into the representation directly, while our approach incorporates the sentiment polarity information into the vectors in a supervised manner. In addition, a biased corruption process which favors informative sentiment words is used when calculating the document representation.

Related Work

Our work is inspired by two fields of research: integrating sentiment information into semantic word embedding and learning continuous feature representations for variable-sized texts.

Sentiment-specific Word Embedding

The common method to generate word embedding is based on language model. Mikolov et al.(2013) simplified the structure of a neural probabilistic language model (NNLM) (Bengio et al. 2003) and introduced two efficient models which called CBoW and Skip-Gram. While Collobert and Weston(2008) proposed the C&W model that train word embedding in a ranking fashion with hinge loss function. However, a key limitation of the above models for sentiment analysis is that they map words with similar contexts but opposite sentiment polarities into close word vectors in the embedding space. To avoid this limitation, Labutov et al.(2013) re-embedding existing word embeddings with logistic regression by regarding sentiment supervision of sentences as a regularization item. In contrast, Tang et al.(2014) introduced a neural network based on C&W model to incorporate the supervision from sentiment polarity of text for twitter sentiment classification. In addition, they extended their work and developed a number of neural networks with tailoring loss functions to learn sentiment-specific word embedding (Tang et al. 2016b). These models focus on Twitter sentiment classification and predict or rank sentiment polarity based on word embeddings in a fixed window of words across a sentence. Based on the Skip-Gram model, Zhang et al.(2015) proposed a model for word-level and sentence-level sentiment analysis, and Yang et al.(2017) proposed a model that predicted the target word and its label simultaneously. Both of them took sentiment information as a part of the local context. Unlike their work, we develop an architecture that takes sentiment information as global representation and learn sentiment-specific embedding for both sentence-level and document-level texts.

Feature Representations of Texts

There are two major ways to learn text representation: supervised and unsupervised. For supervised learning, researchers have explored different neural networks for sentence-level and document-level text representations for sentiment classification, such as Convolutional Neural Network (Kim 2014) and its variants (Nal, Edward, and Phil 2014; Yin and Schtze 2015), Recursive Neural Network models which learn compositionally vector representations for phrases and sentences(Socher et al. 2012), and Long Short-Term Memory (Kai Sheng Tai 2015; Tang et al. 2016a). For unsupervised learning, Skip-Thought vectors (Kiros et al. 2015) uses an encoder-decoder to model the surrounding sentences of the encoded passage and maps similar sentences into vectors, while Paragraph Vectors (Dai, Olah, and Le 2015) explicitly learns a document representation with the word embeddings. However, the complexity of these models grows with the size of the training corpus. To alleviate this problem, Chen(2017) proposed the Doc2VecC that simply represents a document as an average of word embeddings that are randomly sampled from the document. We borrow the idea and take the text representation as the global sentiment information, which jointly learned with local contexts. With the help of sentiment-dependent corruption strategy, our models can learn informative word embeddings which have strong discrimination capability for sentiment classification.

Conclusion

In this paper, we propose a sentiment-specific word embedding learning architecture via global sentiment representation. Different from the existing studies that learn sentiment-specific word embedding from sentence-level texts for the specific classification task, our architecture can learn sentiment-specific word embedding from both sentence-level and document-level texts. We utilize global sentiment representation as well as local context to learn word embeddings. We take global sentiment representation as a simple average of word embeddings in the text with a corruption strategy. The corruption strategy can be seen as a sentiment-dependent regularization for global sentiment representation. It makes our models favor informative sentiment words and reduce the complexity of our models. Based on the proposed architecture, we introduce two neural networks to effectively learn sentiment-specific word embedding. We evaluate the effectiveness of the learned sentiment-specific word embedding on sentence-level and document-level sentiment classification. Experimental results show that our architecture is adept in learning sentiment-specific word embeddings and outperforms the baseline methods.

Acknowledgements

This work was supported by National Natural Science Foundation of China (No. 61502478, No.61602467), National Key Research and Development Program of China (2016YFB1000604) and National HeGaoJi Key Project (2013ZX01039-002-001-001). The authors would like to thank the anonymous reviewers and the area chairs for their constructive comments.

References

Bengio, Y.; Ducharme, R.; Vincent, P.; and Janvin, C. 2003. A neural probabilistic language model. *Journal of Machine Learning Research* 1137–1155.

Chen, M. 2017. Efficient vector representation for documents through corruption. In *Proceedings of the 5th international conference on Learning Representations*.

Collobert, R., and Weston, J. 2008. A unified architecture for natural language processing: Deep neural networks

with multitask learning. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, 160–167. New York, NY, USA: ACM.

Dai, A. M.; Olah, C.; and Le, Q. V. 2015. Document embedding with paragraph vectors. In *Proceedings of Neural Information Processing Systems Deep Learning Workshop*.

Ding, Y.; Liu, Y.; Luan, H.; and Sun, M. 2017. Visualizing and understanding neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, 1150–1159. Association for Computational Linguistics.

Fan, R.-E.; Chang, K.-W.; Hsieh, C.-J.; Wang, X.-R.; and Lin, C.-J. 2008. Liblinear: A library for large linear classification. 1871–1874.

Guan, Z.; Chen, L.; Zhao, W.; Zheng, Y.; Tan, S.; and Cai, D. 2016. Weakly-supervised deep learning for customer review sentiment classification. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, IJCAI'16, 3719–3725. AAAI Press.

Hao, Y.; Zhang, Y.; Liu, K.; He, S.; Liu, Z.; Wu, H.; and Zhao, J. 2017. An end-to-end model for question answering over knowledge base with cross-attention combining global knowledge. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, 221–231. Association for Computational Linguistics.

Igor Labutov, H. L. 2013. Re-embedding words. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 489–493.

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

Kai Sheng Tai, Richard Socher, C. D. M. 2015. Improved semantic representations from tree-structured long short-term memory networks. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 1556–1566.

Kim, Y. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 1746–1751.

Kiros, R.; Zhu, Y.; Salakhutdinov, R.; R, R.; Zemel; Torralba, A.; Urtasun, R.; and Fidler, S. 2015. Skip-thought vectors. In *Proceedings of the Advances in Neural Information Processing Systems 28*, 3294–3302.

Liner Yang, Xinxiong Chen, Z. L. M. S. 2017. Improving word representations with document labels. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 25(4):863–870.

Maas, A. L.; Daly, R. E.; Pham, P. T.; Huang, D.; Ng, A. Y.; and Potts, C. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 142–150.

Mikolov, T.; Chen, K.; Corrado, G.; and Dean, J. 2013. Efficient estimation of word representations in vector space. In *Proceedings of the 1st international conference on Learning Representations*.

Nakov, P.; Rosenthal, S.; Kozareva, Z.; Stoyanov, V.; Ritter, A.; and Wilson, T. 2013. Semeval-2013 task 2: Sentiment analysis in twitter. In *Proceedings of the International Workshop on Semantic Evaluation*.

Nal, K.; Edward, G.; and Phil, B. 2014. A convolutional neural network for modelling sentences. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 655–665.

Nguyen, D. Q.; Nguyen, D. Q.; Vu, T.; and Pham, S. B. 2014. Sentiment Classification on Polarity Reviews: An Empirical Study Using Rating-based Features. In *Proceedings of the 5th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, 128–135. Baltimore, Maryland: Association for Computational Linguistics.

Pennington, J.; Socher, R.; and Manning, C. D. 2014. Glove: Global vectors for word representation. In *Proceedings of the Empirical Methods in Natural Language Processing*, 1532–1543.

Socher, R.; Huang, E. H.; Pennington, J.; Ng, A. Y.; and Manning, C. D. 2011. Dynamic pooling and unfolding recursive autoencoders for paraphrase detection. In *Proceedings of The Annual Conference on Neural Information Processing Systems*, 801–809.

Socher, R.; Huval, B.; Manning, C. D.; and Ng, A. Y. 2012. Semantic compositionality through recursive matrix-vector spaces. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 1201–1211.

Tang, D.; Wei, F.; Yang, N.; Zhou, M.; Liu, T.; and Qin, B. 2014. Learning sentiment-specific word embedding for twitter sentiment classification. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 1555–1565.

Tang, D.; Qin, B.; Feng, X.; and Liu, T. 2016a. Effective lstms for target-dependent sentiment classification. In *Proceedings of the International Conference on Computational Linguistics*, 3298–3307.

Tang, D.; Wei, F.; Qin, B.; Yang, N.; Liu, T.; and Zhou, M. 2016b. Sentiment embeddings with applications to sentiment analysis. *IEEE Transactions on Knowledge and Data Engineering* 28:496–509.

Wang, X.; Liu, Y.; Sun, C.; Wang, B.; and Wang, X. 2015. Predicting polarities of tweets by composing word embeddings with long short-term memory. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 1343–1353.

Yin, W., and Schtze, H. 2015. Multichannel variable-size convolution for sentence classification. In *Proceedings of The SIGNLL Conference on Computational Natural Language Learning*, 204–214.

Yoshua Bengio, Aaron C. Courville, P. V. 2013. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35:1798–1828.

Zhihua Zhang, M. L. 2015. Learning sentiment-inherent word embedding for word-level and sentence-level sentiment analysis. In *Proceedings of the 2015 International Conference on Asian Language Processing (IALP)*, 94–97.