A Lexicon-Based Supervised Attention Model for Neural Sentiment Analysis

Yicheng Zou, Tao Gui, Qi Zhang, Xuanjing Huang
Shanghai Key Laboratory of Intelligent Information Processing, Fudan University
School of Computer Science, Fudan University
825 Zhangheng Road, Shanghai, China
rowitzou@gmail.com, {tgui16,qz,xjhuang}@fudan.edu.cn

Abstract

Attention mechanisms have been leveraged for sentiment classification tasks because not all words have the same importance. However, most existing attention models did not take full advantage of sentiment lexicons, which provide rich sentiment information and play a critical role in sentiment analysis. To achieve the above target, in this work, we propose a novel lexicon-based supervised attention model (LBSA), which allows a recurrent neural network to focus on the sentiment content, thus generating sentiment-informative representations. Compared with general attention models, our model has better interpretability and less noise. Experimental results on three large-scale sentiment classification datasets showed that the proposed method outperforms previous methods.

1 Introduction

Sentiment analysis has the goal of analyzing people's sentiments or opinions and has been well explored (Turney, 2002; Tang et al., 2014b; Chen et al., 2016). In order to improve sentiment analysis results, large sentiment lexicons have been built (Wilson et al., 2005; Baccianella et al., 2010; Tang et al., 2014a). A sentiment lexicon is a set of words such as *excellent*, *terrible* and *ordinary*, each of which is assigned a fixed positive or negative score that presents its sentiment polarity and strength (Tang et al., 2014a). Such information can serve as sentiment-informative features and significantly boost the classification performance (Agarwal et al., 2013; Teng et al., 2016; Qian et al., 2016).

In sentiment classification tasks, sentiment words such as *great* and *terrible* tend to play a more critical role than other words in texts (Liu, 2010). One attribute of words annotated in sentiment lexicons is their sentiment strength, which is intuitively associated with a word's contribution to the sentiment representation of a sentence. It is similar to the basic idea of the attention mechanism that not all words have the same importance. However, very few studies have focused on a method that combines an attention mechanism with such sentiment information. Yang et al. (2016) and Chen et al. (2016) proposed a hierarchical RNN model to learn attention weights based on the local context using an unsupervised method. However, their method may induce much noise and suffers from a lack of interpretability because it tends to capture some domain-specific words (Mudinas et al., 2012) instead of real sentiment information. For example, in movie reviews, the name of a movie with a good reputation tends to be regarded as positive words, and is thereby assigned higher weights, which does not work in other domains. Long et al. (2017) incorporated the reading time of human beings into the attention mechanism, but their method also has much noise because during the reading process, people tend to spend more time on intricate contents (Goodman, 1988) than on real sentiment words like *good* and *bad*. Other methods employed external information such as users and products to guide attention weights (Ma et al., 2017; Chen et al., 2016), which can boost the classification performance by a large margin. However, most of the time, we have no access to such external information.

In this paper, we propose a novel Lexicon-Based Supervised Attention model (LBSA) that combines sentiment lexicons and an attention mechanism to better extract sentiment information and form

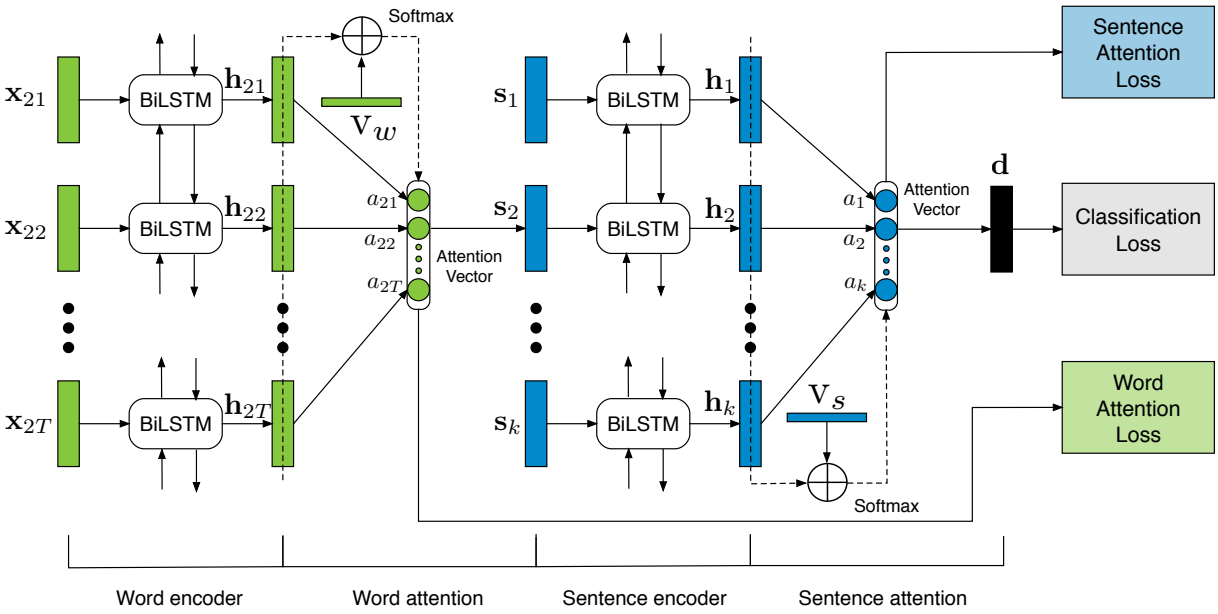

Figure 1: LBSA sentiment classification model.

sentiment-informative representations. We hypothesized that regardless of the polarities, sentiment words should be given more emphasis than other words in texts. For example, in the phrases *great film* and *terrible weather*, we should pay more attention to the sentiment words *great* and *terrible* than the words *film* and *weather*. Moreover, we define the *sentiment degree* to describe the intensity of the word sentiment, which can be attained from existing sentiment lexicons. The sentiment degree can then be used as the criterion for attention weights in a recurrent neural network model. Under the guidance of a supervised mechanism, sentiment words will be assigned higher attention weights, which reduces the influence of domain-specific words and generates sentiment-informative representations. We conducted experiments on three sentiment analysis benchmark datasets (IMDB, Yelp13, Yelp14). The experimental results showed that our model outperformed other state-of-the-art methods.

To summarize, our main contributions are as follows:

- We propose a novel lexicon-based attention model. It combines an attention mechanism with sentiment lexicons based on the *sentiment degree*. As a result, our model can precisely identify the sentiment contents in texts and capture accurate sentiment information.
- Benefiting from the guidance of supervised methods, our attention model has better interpretability and less noise than other attention models. Thus, it is better at summarizing and analyzing the sentiment of texts in an RNN fashion.
- Our model outperforms state-of-the-art methods on three sentiment analysis datasets (IMDB, Yelp13, Yelp14). This work validated the effectiveness of combining the sentiment lexicons and attention mechanism.

2 Approach

The overall architecture of our model is shown in Figure 1. It incorporates a word-level supervised attention layer and sentence-level supervised attention layer into a hierarchical bidirectional LSTM sentiment classification model. Formally, let D be a set of documents, and L be a sentiment lexicon. A document $d_m \in D$ contains k sentences S_1, S_2, ..., S_i, ..., S_k. A sentence S_i is a sequence of words w_{i1} w_{i2}...w_{it}, $t \in [1, T]$, where T denotes the length of S_i. For each word w_{it}, we define $SD^L(w_{it})$ as the *sentiment degree* (see Section 2.2.1) of w_{it} according to lexicon L. The sentiment degree is a positive real number, which can be attained from lexicons and indicates the strength of word sentiment.

As the figure shows, the word w_{it} is represented by its word embedding $\mathbf{x}_{it}$, which is fed into a bidirectional LSTM model (Graves and Schmidhuber, 2005; Graves et al., 2013) to extract sequential information. The word attention layer aggregates the representation of sentiment-informative words to form a sentence representation $\mathbf{s}_i$, with the supervision of the sentiment degree. Similarly, we can obtain the document representation $\mathbf{d}$ through a sentence-level attention layer to finally conduct sentiment classification. The concrete design is introduced in the following subsections.

2.1 Bidirectional LSTM

We adopt a Long Short-Term Memory (LSTM) network (Hochreiter and Schmidhuber, 1997) for the word-level and sentence-level feature extraction. As an example, consider the word-level case. For each word token w_t in a sentence, the model calculates a hidden state vector $\mathbf{h}_t$. The basic LSTM model has five internal vectors for node t. They control the information flow from the history $\mathbf{x}_1$ ...$\mathbf{x}_t$ and $\mathbf{h}_1$ ...$\mathbf{h}_{t-1}$ to the current state $\mathbf{h}_t$. Formally, $\mathbf{h}_t$ is calculated as follows:

$$
\begin{aligned}
\mathbf{i}_t &= \sigma(\mathbf{W_i}\mathbf{x}_t + \mathbf{U_i}\mathbf{h}_{t-1} + \mathbf{V_i}\mathbf{c}_{t-1} + \mathbf{b_i}) \\
\mathbf{f}_t &= \mathbf{1.0} - \mathbf{i}_t \\
\mathbf{c}'_t &= tanh(\mathbf{W_{c'}}\mathbf{x}_t + \mathbf{U_{c'}}\mathbf{h}_{t-1} + \mathbf{b_{c'}}) \\
\mathbf{c}_t &= \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \mathbf{c}'_t \\
\mathbf{o}_t &= \sigma(\mathbf{W_o}\mathbf{x}_t + \mathbf{U_o}\mathbf{h}_{t-1} + \mathbf{V_o}\mathbf{c}_t + \mathbf{b_o}) \\
\mathbf{h}_t &= \mathbf{o}_t \odot tanh(\mathbf{c}_t)
\end{aligned}
$$

Each word is embedded into a low-dimensional semantic space. Namely, each word w_t is mapped to its embedding $\mathbf{x}_t \in \mathbb{R}^n$, where n is the word embedding size. $\odot$ denotes element-wise multiplication, and σ is the sigmoid function. $\mathbf{W_i}$, $\mathbf{U_i}$, $\mathbf{V_i}$, $\mathbf{b_i}$, $\mathbf{W_{c'}}$, $\mathbf{U_{c'}}$, $\mathbf{b_{c'}}$, $\mathbf{W_o}$, $\mathbf{U_o}$, $\mathbf{V_o}$, and $\mathbf{b_o}$ represent LSTM parameters.

We use a bidirectional version of LSTM (BiLSTM) (Graves and Schmidhuber, 2005; Graves et al., 2013) to obtain information from both directions for words. The bidirectional LSTM contains the forward LSTM $\overrightarrow{f}$, which reads the sentence S_i from w_{i1} to w_{it}, and the backward LSTM $\overleftarrow{f}$, which reads from w_{it} to w_{i1}. The BiLSTM model maps each word embedding $\mathbf{x}_{it}$ to a pair of hidden vectors $\overrightarrow{\mathbf{h}_{it}}$ and $\overleftarrow{\mathbf{h}_{it}}$. We use different parameters for the forward LSTM and backward LSTM. These hidden vectors are the composition of sentence embedding $\mathbf{s}_i$, and used as features for calculating attention weights. On the sentence level, we also feed sentence embeddings $[\mathbf{s}_1, \mathbf{s}_2, ..., \mathbf{s}_k]$ into BiLSTM to obtain pairs of hidden vectors in a similar way.

2.2 Lexicon-based Supervised Attention

In this subsection, we introduce supervised attention to explicitly use sentiment lexicon information to aggregate sentiment-informative texts. Our basic idea is simple: sentiment words should be given more attention than others. To achieve this goal, we construct gold attention vectors using the lexicon information. These are employed as criteria to supervise the attention weights.

2.2.1 Gold Attention Vectors

Each word in the sentiment lexicon is annotated or assigned a fixed positive or negative score that reflects its sentiment polarity and strength. We map all of them into the interval $[-1, 1]$ as normalized sentiment scores, where a score in $(0, 1]$ represents various scales of *good* and one close to 1 means *very good*, with negations showing the opposite. Intuitively, a word with a strong positive or negative polarity requires more attention because it contains more sentiment information than a neutral word. In order to quantify the intensity of the sentiment, for each word w_{it} in sentence s_i, we define *sentiment degree* $SD^L(w_{it})$ as follows:

$$SD^L(w_{it}) = |score^L(w_{it})|, \tag{1}$$

where $score^L(w_{it})$ is the normalized sentiment score according to lexicon L. $SD^L(w_{it})$ is the absolute value of $score^L(w_{it})$, which means its range is $[0, 1]$. For words not in the lexicon, their sentiment

degrees are assigned values of 0. For sentence S_i, whose length is T, we compute word-level gold attention vectors $\mathbf{a}_i^\star$ by:

$$a_{it}^\star = \frac{exp(\lambda_w SD^L(w_{it}))}{\sum_{t=1}^{T} exp(\lambda_w SD^L(w_{it}))}. \tag{2}$$

λ_w is a positive hyper-parameter that adjusts the variance of the sentiment degree. A larger λ_w means a sharper distinction between sentiment terms and neutral ones in gold attention vectors. $a_{it}^\star$ denotes the t^{th} dimension of $\mathbf{a}_i^\star$.

We likewise construct sentence-level gold attention vectors. Given a document with k sentences, S_i denotes the i^{th} sentence in it. Empirically, a sentence tends to achieve a higher sentiment degree if it contains a higher proportion of sentiment words. Hence, we calculate the sentiment degree of S_i using a simple averaging method:

$$SD^L(S_i) = \frac{\sum_{t=1}^{T} SD^L(w_{it})}{T}. \tag{3}$$

Let λ_s be the sentence-level hyper-parameter. We then obtain sentence-level gold attention vectors $\mathbf{a}^\star$ in a similar way:

$$a_i^\star = \frac{exp(\lambda_s SD^L(S_i))}{\sum_{i=1}^{k} exp(\lambda_s SD^L(S_i))}. \tag{4}$$

2.2.2 Learned Attention Vectors

As described in the previous subsection, we obtain pairs of hidden states $\overrightarrow{\mathbf{h}_{it}}$ and $\overleftarrow{\mathbf{h}_{it}}$ from the word-level BiLSTM model. They can be employed to calculate a weight value π_{it}, which presents the sentiment strength of word w_{it}. Formally, π_{it} is defined as follows:

$$\pi_{it} = tanh(\mathbf{W}_{wf}\overrightarrow{\mathbf{h}_{it}} + \mathbf{W}_{wb}\overleftarrow{\mathbf{h}_{it}} + \mathbf{b}_w) \cdot \mathbf{v}_w^\top, \tag{5}$$

where $\mathbf{W}_{wf}$ and $\mathbf{W}_{wb}$ denote word-level weight matrices, and $\mathbf{b}_w$ is the bias vector. It is worth noting that $\mathbf{v}_w$ is a weight vector that can record historical sentiment information. It is just like the query vector employed by memory networks (Sukhbaatar et al., 2015). It is jointly learned during the training process. $\mathbf{v}_w^\top$ represents the transpose of $\mathbf{v}_w$.

After calculating the weight value π_{it}, we use a softmax function to generate the attention vector $\mathbf{a}_i$. The t^{th} dimension a_{it} denotes the attention weight of the t^{th} word w_{it} in sentence S_i. Specifically, a_{it} is computed as follows:

$$a_{it} = \frac{exp(\pi_{it})}{\sum_{t=1}^{T} exp(\pi_{it})}. \tag{6}$$

The learned attention vector $\mathbf{a}_i$ contains different weights for the corresponding words in sentence S_i. We then compute the sentence vector $\mathbf{s}_i$ as a weighted sum of the hidden vectors according to the weights, where $\mathbf{h}_{it}$ means a concatenation of $\overrightarrow{\mathbf{h}_{it}}$ and $\overleftarrow{\mathbf{h}_{it}}$:

$$\mathbf{s}_i = \sum_{t=1}^{T} a_{it}\mathbf{h}_{it}. \tag{7}$$

After obtaining sentence vectors $\mathbf{s}_i$, $1 \leq i \leq k$, we feed them into the sentence-level BiLSTM model to output hidden vector pairs. Similarly, with the attention mechanism, the hidden vector pairs $\overrightarrow{\mathbf{h}_i}$ and $\overleftarrow{\mathbf{h}_i}$, and their concatenation $\mathbf{h}_i$, document vector $\mathbf{d}$ can be induced as follows:

$$\pi_i = tanh(\mathbf{W}_{sf}\overrightarrow{\mathbf{h}_i} + \mathbf{W}_{sb}\overleftarrow{\mathbf{h}_i} + \mathbf{b}_s) \cdot \mathbf{v}_s^\top, \tag{8}$$

$$a_i = \frac{exp(\pi_i)}{\sum_{i=1}^{k} exp(\pi_i)}, \tag{9}$$

$$\mathbf{d} = \sum_{i=1}^{k} a_i\mathbf{h}_i. \tag{10}$$

Data	Class	Train Size	Dev. Size	Test Size	Sents/doc	Words/sent
IMDB	10	67,426	8,381	9,112	16.08	24.54
Yelp14	5	183,019	22,745	25,399	11.41	17.26
Yelp13	5	62,522	7,773	8,671	10.89	17.38

Table 1: Statistical information for three datasets. Sents/doc is the average number of sentences in a document, while Words/sent denotes the average length of a sentence.

2.3 Jointly Supervised Classification and Attention

The final document vector $\mathbf{d}$ is used to conduct sentiment classification, which requires the probability of labeling a document with sentiment polarity c, $c \in [1, C]$, where C represents the total number of classes. The probability is computed by a softmax function:

$$\mathbf{p} = softmax(\mathbf{W}_c\mathbf{d} + \mathbf{b}_c). \tag{11}$$

To jointly supervise the classification and attention, we introduce a soft constraint method that is employed by other tasks with supervised attention (Liu et al., 2016). Specifically, it is defined as follows:

$$\mathcal{L} = -\sum_{c=1}^{C} \mathbf{g}_c log(\mathbf{p}_c) + \mu_w \cdot \sum_{i=1}^{T} \triangle(\mathbf{a}_i^\star, \mathbf{a}_i) + \mu_s \cdot \triangle(\mathbf{a}^\star, \mathbf{a}), \tag{12}$$

where the classification loss function is cross entropy. $\mathbf{g}_c \in \mathbb{R}^C$ denotes the ground truth of the sentiment classification, presented by a one-hot vector. $\mathbf{p}_c \in \mathbb{R}^C$ is the predicted probability for each class. $\mathbf{a}_i^\star$ and $\mathbf{a}^\star$ are gold attention vectors for the word and sentence levels, respectively. $\mathbf{a}_i$ and $\mathbf{a}$ are learned attention vectors for the word and sentence levels, respectively. $\triangle$ is a loss function that indicates the disagreement between two vectors. Because the learned attention vector is a distribution, we naturally use cross entropy (CE) as the metric:

$$\triangle(\mathbf{a}^\star, \mathbf{a}) = -\sum_i \mathbf{a}_i^\star log(\mathbf{a}_i). \tag{13}$$

μ_w and μ_s are the coefficients for attention loss functions, which can balance the preference between classification and attention disagreements, to alleviate the overfitting problem.

3 Experimental Settings

In this section, we introduce the experimental settings in detail, including the datasets and lexicons, hyper-parameter settings, and baseline models.

3.1 Datasets and Lexicons

We conduct experiments to evaluate the effectiveness of our method on three document-level review datasets: IMDB, Yelp 2013 and Yelp 2014, which are developed by Tang et al. (2015). We split the datasets into training, development and testing sets in the proportion of 8:1:1, using pre-processing in the same way as Tang et al. (2015). Table 1 summarizes the statistics of the datasets. All of these datasets can be publicly accessed[1].

The sentiment lexicon contains four parts. During the process of constructing our lexicon, we only use the sentiment scores for unigrams. The first part comes from SentimentWordNet3.0 (SWN) (Baccianella et al., 2010). The original scores in SWN are the probabilities of positive, negative, or neutral polarities for words. We take the maximum sum of the positive and negative scores as the sentiment degree for different part-of-speech tags. The second part is extracted from MPQA (Wilson et al., 2005), which tags polarity ratings for sentiment words. To a word whose polarity is positive or negative, we assign 1 as its sentiment degree. Otherwise, the word's sentiment degree is 0. The third part consists of the leaf nodes of the Stanford Sentiment Treebank (SST) dataset (Socher et al., 2013). The sentiment scores of

[1] http://ir.hit.edu.cn/dytang/paper/acl2015/dataset.7z

SST are in the range of [0, 1]. We scale it to [-1, 1] and keep the absolute value as the sentiment degree. The last part is the sentiment lexicon of the Hownet Knowledge Database (HKD)[2], which contains a list of positive or negative words. We conduct the pre-processing in the same way as for MPQA. Finally, we combine the four parts and average the sentiment degree for words appearing in two or more lexicon parts. All of the neutral words in different parts are included in our lexicon because of the ignorance of polarity. Hence, the final lexicon is huge, containing 156,242 tokens and covering 96.8% of the words in three review datasets.

The sentiment lexicon is of great significance in our experiments. A lexicon of high quality ensures the effectiveness of the supervised mechanism. Empirically, if the overlapping of the dataset and lexicon is poor, the performance will be limited. The availability of lexicons is also not a given for any domain or language. In such scenarios, several existing methods can be used to construct or extend sentiment lexicons based on large corpora in a semi-supervised way to cover the dataset (Lu et al., 2011; Hamilton et al., 2016). As the above potential problems are not the main concern of our work, we just briefly touch on this and will explore it in future work.

3.2 Hyper-parameter Settings

Following Tang et al. (2015) and Chen et al. (2016), we set the dimension of the word embeddings to 200. We use the word2vec tool (Mikolov et al., 2013) to pre-train the word embeddings. We set the dimensions of the word-level bidirectional LSTM hidden states to 100. All the weight matrices are initialized by a uniform distribution in [-0.1, 0.1]. The hyper-parameter λ_w and λ_s which adjust the variance of the gold attention vectors, are both set to 3.0. The hyper-parameter μ_w which adjusts the proportion of classification and attention loss, is set to 0.001 while μ_s is set to 0.05. We use Adam (Kingma and Ba, 2014) as the optimizer, which adopts a self-adaptive learning rate to optimize parameters, and its initial learning rate is set to 0.001. The batch size is set to 32 for efficiency. The model achieving best results on the developing set is chosen for the final evaluation of the test set.

3.3 Baseline models

We separate baseline models into four groups. In the first group, all of the methods are recently developed and achieve good performances on three review benchmarks. **SSWE + SVM** (Tang et al., 2014b) first generates sentiment-specific word embeddings to compose document presentations and then trains a SVM classifier. **Paragraph + Vector** (Le and Mikolov, 2014) learns distributed representations of a document for classification. **RNTN + RNN** (Socher et al., 2013) represents sentences with the Recursive Neural Tensor Network (RNTN) and feeds them into a recurrent neural networks (RNN) to obtain document representations. **UPNN** (Tang et al., 2015) uses a text preference matrix and vector for each user and product as extra information to train a CNN sentiment classifier. *UPNN(noUP)* only uses CNN without considering user and product information.

The models in Group 2 are based on sentiment lexicons. **BiLSTM** (Xu et al., 2016) is a bidirectional LSTM baseline with a simple average pooling layer. **AveLex** naively averages sentiment scores of words in the document to measure the overall sentiment polarities according to our sentiment lexicon. **BiLSTM + Lex** (Teng et al., 2016) takes the local context into consideration, which leverages a bidirectional LSTM to capture context information and calculates the weighted sum of the sentiment scores. The two lexicon-based methods only work on the word level because sentences do not have a gold sentiment score. For a fair comparison, **BiLSTM + LBSA** is our model, which removes the hierarchical structure.

The models in Group 3 are based on attention mechanisms. **H-LSTM** (Chen et al., 2016) is a baseline utilizing an average pooling layer at both the word and sentence levels. **H-LSTM + LA** (Chen et al., 2016) uses local context to capture semantic information as an attention mechanism. **H-LSTM + CBA** (Long et al., 2017) adds cognitive information from external reading time materials. For a fair comparison, we simplify our method from bidirectional LSTMs to basic ones, which is denoted as **H-LSTM + LBSA**.

[2]http://www.keenage.com/html/e_index.html

Model	IMDB		Yelp 2013		Yelp 2014	
	Acc.	RMSE	Acc.	RMSE	Acc.	RMSE
SSWE + SVM (Tang et al., 2014b)	0.312	1.973	0.549	0.849	0.557	0.851
Paragraph + Vector (Le and Mikolov, 2014)	0.314	1.814	0.554	0.832	0.564	0.802
RNTN + RNN (Socher et al., 2013)	0.401	1.764	0.574	0.804	0.582	0.821
UPNN(noUP) (Tang et al., 2015)	0.405	1.629	0.577	0.812	0.585	0.808
AveLex	0.223	2.388	0.312	1.393	0.334	1.202
BiLSTM (Xu et al., 2016)	0.433	1.494	0.584	0.764	0.592	0.733
BiLSTM + Lex (Teng et al., 2016)	0.441	1.466	0.588	**0.758**	0.598	0.726
BiLSTM + LBSA	**0.443**	**1.457**	**0.590**	0.761	**0.603**	**0.720**
H-LSTM (Chen et al., 2016)	0.443	1.465	0.627	0.701	0.637	0.686
H-LSTM + LA (Chen et al., 2016)	0.487	1.381	0.631	0.706	0.631	0.715
H-LSTM + CBA (Long et al., 2017)	**0.489**	1.365	0.638	0.697	0.641	0.678
H-LSTM + LBSA	0.488	**1.360**	**0.640**	**0.694**	**0.647**	**0.670**
H-BiLSTM	0.484	1.399	0.640	0.700	0.646	0.672
H-BiLSTM + LA	0.492	1.359	0.647	0.698	0.648	**0.665**
H-BiLSTM + LBSA	**0.494**	**1.322**	**0.650**	**0.691**	**0.651**	0.668

Table 2: Sentiment classification performance on different models. Acc. (Accuracy) and RMSE are the evaluation metrics. The best results are in bold in the last three groups, respectively.

In Group 4, **H-BiLSTM** is a hierarchical bidirectional LSTM model with an average pooling layer. **H-BiLSTM + LA** is a BiLSTM version of the local attention model based on Chen et al. (2016).

4 Results

This section compares our model (LBSA) with other baseline methods. Two common performance evaluation metrics are used, including Accuracy (Acc.) and Root Mean Squared Error (RMSE). They are defined as follows:

$$Accuracy = \frac{T}{N}, \tag{14}$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{N}}. \tag{15}$$

Accuracy is a standard metric to measure the overall sentiment classification performance, where N is the number of overall samples, and T denotes the number of predicted outputs that are identical with gold labels. RMSE measures the divergences between the predicted sentiment classes and ground truth ones, where $\hat{y}_i$ and y_i present the gold labels and predicted outputs, respectively. Table 2 illustrates the comparison results, which are separated into four groups. Some results of the baseline methods are directly taken from Long et al. (2017) and Xu et al. (2016), because we conducted experiments on the same datasets.

We can observe from Group 2 that the performances are limited without a hierarchical structure because the texts are very long in document-level sentiment classification. **AveLex** is the worst method because it naively averages the sentiment scores of words without considering any textual semantics. Compared with **BiLSTM + Lex**, our method achieves a relatively better result. The main reason is that our method ignores the prior polarity of the sentiment words in lexicons. LBSA is able to lead neural networks to concentrate on sentimental contents and learn a more accurate polarity according to historical information. To illustrate the results more specifically, we investigate the sentiment word *long* in our datasets. *Long* has a sentiment degree of 0.42 in our constructed lexicon, which indicates that it plays a relatively critical role in sentiment classification. The sentiment polarity of *long* in MPQA and SST is negative. A random sentence sample *'The several tables were available, so I didn't understand*

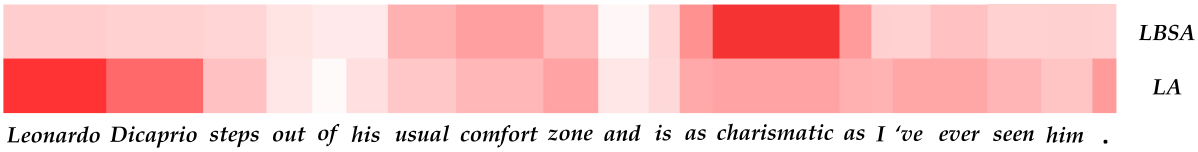

Figure 2: Sentence example from IMDB. Local attention (LA) focuses more on the domain-specific words *Leonardo Dicaprio*, while our model (LBSA) can exactly capture the real sentiment word *charismatic*.

*the **long** wait.'* is taken from a one-star review in Yelp 2013, which meets our expectation. However, in other domains or contexts, *long* is likely to be positive. For instance, the sentence *'Its appeal to me was immediate and **long** lasting.'* is extracted from a 10-star movie review in IMDB. It is obviously a positive sentence, but the prior polarity of *long* in the sentiment lexicon misleads the judgment of general methods that directly use sentiment scores.

In Group 3, we observe that our method significantly outperforms other attention mechanisms with the Yelp 2013 and Yelp 2014 datasets. The evaluation on the RMSE metric is also better than other methods with the IMDB dataset. This demonstrates that LBSA is effective at capturing sentiment information, which can be a crucial factor in sentiment classification. Compared with **H-LSTM + CBA**, our model achieves better results, which means lexicon information is more suitable for attention supervision in sentiment analysis tasks. In the forth group, the bidirectional LSTM improves the basic performance, which indicates that both forward and backward sequences indispensably contribute to information extraction. Compared with local context-based attention, the basic performance of our method is outstanding and has a relatively small improvement when a bidirectional mechanism is added. This indicates that our method lays emphasis on contents containing rich sentiment information, which is robust in a simpler structure. In contrast, local context-based attention relies heavily on contexts in both directions. The characteristic of being dedicated to the contexts may induce domain-specific words and ignore real sentiment information. Here, we use a sentence from a movie review in IMDB as an example: *'Leonardo Dicaprio steps out of his usual comfort zone and is as charismatic as I've ever seen him.'* Figure 2 illustrates the main difference between two attention mechanisms.

The LA model gives higher weights to the name *Leonardo Dicaprio* because it usually appears in a positive movie review. The words are domain-specific and attract attention from real sentiment words. In contrast, our method tends to extract specific sentiment information. As a result, it assigns a very high weight to the word *charismatic*, which indeed plays a decisive role as a positive adjective.

5 Related Work

Sentiment lexicons are widely used in sentiment analysis and opinion mining tasks because they contain rich sentiment information. Turney (2002) employed the sum of the sentiment scores of all the words in the texts that appeared in a sentiment lexicon. This is still the standard practice for specific domains with some carefully constructed domain-specific lexicons. Agarwal et al. (2013) proposed a tree-structured model leveraging non-polarity features such as POS-tags and capitalized words, adding the summation of the prior polarity scores of words, which achieved high performance on the twitter sentiment analysis benchmark. Teng et al. (2016) presented a context-sensitive lexicon-based method that uses a bidirectional LSTM to extract context information. It calculates the weighted sum of the sentiment scores of words to measure the sentiment value of a sentence. Qian et al. (2016) proposed a linguistically regularized LSTM for sentiment analysis. The model combines linguistic resources such as sentiment lexicons, negation words and intensity words with the basic LSTM model. It is able to capture the semantic information and sentiment effects in sentences more accurately. Unlike most previous studies, we incorporate the features of lexicon words into an attention mechanism, which was proven to be effective in our experiments.

Recently, the attention mechanism has been widely studied in sentiment classification. Yang et al. (2016) proposed two attention-based bidirectional GRUs to enforce the neural networks to attend to

the related part of a sentence or document. Apart from local contexts, Chen et al. (2016) incorporated extra information such as users and products in review datasets into a hierarchical attention mechanism. Ma et al. (2017) cascaded multiway of user and product information to enhance the effects of different aspects. Zhou et al. (2016) proposed a LSTM network based on an attention mechanism for cross-lingual sentiment classification at the document level. The model consists of two attention-based hierarchical LSTMs for bilingual representation. Long et al. (2017) took cognition into consideration, leveraging extra resources including the reading time of human beings. They proposed a mutimodel that learns to predict the reading time to construct the cognitive attention. The main difference between our method and others is that we supervise the attention weights using sentiment information. As a result, it is able to capture real sentiment texts and achieves a better result.

6 Conclusion

In this paper, we propose a novel lexicon-based supervised attention model. We combine an attention mechanism and sentiment lexicons to guide the neural network to focus on the sentiment content. It has better interpretability and less noise compared with other attention models. Experiments on three large review datasets validated the effectiveness of our model, showing that it can capture real sentiment information and generate sentiment-informative representations to improve the classification performance.

Acknowledgements

The authors wish to thank the anonymous reviewers for their helpful comments. This work was partially funded by National Natural Science Foundation of China (No. 61532011, 61473092, and 61472088) and STCSM (No. 16JC1420401).

References

Apoorv Agarwal, Boyi Xie, Ilia Vovsha, Owen Rambow, and Rebecca Passonneau. 2013. Sentiment analysis of twitter data. In *The Workshop on Languages in Social Media*, pages 30–38.

Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. 2010. Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *LREC 2010, 17-23 May 2010, Valletta, Malta*, pages 83–90.

Huimin Chen, Maosong Sun, Cunchao Tu, Yankai Lin, and Zhiyuan Liu. 2016. Neural sentiment classification with user and product attention. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1650–1659.

Kenneth Goodman. 1988. The reading process. *Interactive approaches to second language reading*, 6.

Alex Graves and Jürgen Schmidhuber. 2005. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural Networks*, 18(5-6):602–610.

Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. 2013. Speech recognition with deep recurrent neural networks. In *Acoustics, speech and signal processing (icassp), 2013 ieee international conference on*, pages 6645–6649. IEEE.

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

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *International Conference on Machine Learning*, pages 1188–1196.

Lemao Liu, Masao Utiyama, Andrew Finch, and Eiichiro Sumita. 2016. Neural machine translation with supervised attention. *arXiv preprint arXiv:1609.04186*.

Bing Liu. 2010. Sentiment analysis and subjectivity. *Handbook of natural language processing*, 2:627–666.

Yunfei Long, Lu Qin, Rong Xiang, Minglei Li, and Chu-Ren Huang. 2017. A cognition based attention model for sentiment analysis. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 462–471.

Yue Lu, Malu Castellanos, Umeshwar Dayal, and ChengXiang Zhai. 2011. Automatic construction of a context-aware sentiment lexicon: an optimization approach. In *Proceedings of the 20th international conference on World wide web*, pages 347–356. ACM.

Dehong Ma, Sujian Li, Xiaodong Zhang, Houfeng Wang, and Xu Sun. 2017. Cascading multiway attentions for document-level sentiment classification. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, volume 1, pages 634–643.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Andrius Mudinas, Dell Zhang, and Mark Levene. 2012. Combining lexicon and learning based approaches for concept-level sentiment analysis. In *Proceedings of the first international workshop on issues of sentiment discovery and opinion mining*, page 5. ACM.

Qiao Qian, Minlie Huang, Jinhao Lei, and Xiaoyan Zhu. 2016. Linguistically regularized lstms for sentiment classification. *arXiv preprint arXiv:1611.03949*.

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 *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642.

Sainbayar Sukhbaatar, Jason Weston, Rob Fergus, et al. 2015. End-to-end memory networks. In *Advances in neural information processing systems*, pages 2440–2448.

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 COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, 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 (Volume 1: Long Papers)*, volume 1, pages 1555–1565.

Duyu Tang, Bing Qin, and Ting Liu. 2015. Learning semantic representations of users and products for document level sentiment classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, volume 1, pages 1014–1023.

Zhiyang Teng, Duy Tin Vo, and Yue Zhang. 2016. Context-sensitive lexicon features for neural sentiment analysis. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1629–1638.

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

Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the conference on human language technology and empirical methods in natural language processing*, pages 347–354. Association for Computational Linguistics.

Jiacheng Xu, Danlu Chen, Xipeng Qiu, and Xuangjing Huang. 2016. Cached long short-term memory neural networks for document-level sentiment classification. *arXiv preprint arXiv:1610.04989*.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489.

Xinjie Zhou, Xiaojun Wan, and Jianguo Xiao. 2016. Attention-based lstm network for cross-lingual sentiment classification. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 247–256.