Article

Lexicon-Enhanced Attention Network Based on Text Representation for Sentiment Classification

Wenkuan Li ^{1,2}, Dongyuan Li ³, Hongxia Yin ^{1,2}, Lindong Zhang ^{1,2}, Zhenfang Zhu ^{2,4,*} and Peiyu Liu ^{1,2,*}

¹ School of Information Science and Engineering, Shandong Normal University, Jinan 250358, China

² Shandong Provincial Key Laboratory of Information System and Network Information Security, Shandong Normal University, Jinan 250358, China

³ School of Microelectronics, Shandong University, Jinan 250100, China

⁴ School of Information Science and Electrical Engineering, Shandong Jiaotong University, Jinan 250357, China

* Correspondence: zhuzf@sdjtu.edu.cn (Z.Z.); liupy@sdnu.edu.cn (P.L.)

Received: 21 July 2019; Accepted: 4 September 2019; Published: 6 September 2019

Abstract: Text representation learning is an important but challenging issue for various natural language processing tasks. Recently, deep learning-based representation models have achieved great success for sentiment classification. However, these existing models focus on more semantic information rather than sentiment linguistic knowledge, which provides rich sentiment information and plays a key role in sentiment analysis. In this paper, we propose a lexicon-enhanced attention network (LAN) based on text representation to improve the performance of sentiment classification. Specifically, we first propose a lexicon-enhanced attention mechanism by combining the sentiment lexicon with an attention mechanism to incorporate sentiment linguistic knowledge into deep learning methods. Second, we introduce a multi-head attention mechanism in the deep neural network to interactively capture the contextual information from different representation subspaces at different positions. Furthermore, we stack a LAN model to build a hierarchical sentiment classification model for large-scale text. Extensive experiments are conducted to evaluate the effectiveness of the proposed models on four popular real-world sentiment classification datasets at both the sentence level and the document level. The experimental results demonstrate that our proposed models can achieve comparable or better performance than the state-of-the-art methods.

Keywords: text sentiment classification; sentiment linguistic knowledge; deep learning; attention mechanism

1. Introduction

With the rapid popularization of the Internet, the number of user-generated comments on social media applications has grown exponentially. In particular, social network services such as Twitter, provide a platform for specific users to easily share their personal preferences, opinions, and attitudes about various topics through short posts [1]. Understanding the abundant information of people's opinions expressed in these short posts can help companies to meet the emerging needs of downstream applications such as customer services [2], brand monitoring [3], and recommendation systems [4,5]. For instance, in order to gain more revenue, businesses can adjust their marketing strategies according to the attitudes of specific users towards products and services. Therefore, sentiment classification [6] has been an active research direction in the field of natural language processing (NLP), which aims to automatically classify the sentiment polarity of a given text into negative, positive, or more fine-grained classes.

An essential step of NLP is text representation learning, which is to encode the original text information into continuous vectors by constructing a projection from semantics to points in a

high-dimensional space. The performance of text sentiment analysis completely depends on the comprehensive and high-quality representation of the raw text through representation learning models. Earlier efforts in modeling text for sentiment analysis are mainly based on statistical machine learning methods. The statistical machine learning-based representation models treat the task of text sentiment classification as the word frequency statistics problem, focusing on designing effective sentiment features to achieve better performance, such as n-grams [7], bag-of-words [8], and TF-IDF [9]. Specifically, the bag-of-words feature is to summarize the frequency of co-occurrence sentiment-related words to express the original text as the one-hot vector representation for sentiment classification, where the sentiment polarity of text is largely predicted to be negative if the number of negative sentiment-related words is more than that of the positive ones. Similar to bag-of-words, the n-grams feature shifts focus from co-occurrence sentiment-related words to sentiment-related word pair. The TF-IDF feature returns the weight scores of each word according to certain sentiment lexicon in different documents, which can help the computational process to be more easily implemented in sentiment classification task. These traditional sentiment features show the fact that sentiment linguistic knowledge is essentially a valuable resource that deserves further exploited in the process of text sentiment analysis.

However, in real-world applications, sentiment linguistic knowledge is hardly useful without the information of context features. In general, the ambiguity and complexity of NLP make a great challenge for the simple frequency characteristics of words. Firstly, the concept of semantic composition does not exist in word frequency statistics approaches. For example, it will raise the problem about reverse polarity of sentiment when handling content word negation (e.g., ‘ameliorates fear’), flipping negation (e.g., ‘not sad’) or unbounded dependencies (e.g., ‘no one passed the selection’). Secondly, word frequency statistics methods cannot explicitly identify the single word sense, thus causing the word-level semantic confusion, e.g., unable to distinguish the meaning of word ‘hot’ in the phrases ‘a hot temper’ and ‘a hot, kind girl’. Lastly, simple summation over sentiment-related words can also cause the sentence-level semantic confusion because it does not have the ability to handle sentences with double contrasting polarities. For example, the sentiment polarity score of ‘Thanks for making this unpleasing party more pleasing’ is predicted to be negative because the word ‘unpleasing’ has a stronger sentiment intensity than the word ‘pleasing’ according to the general sentiment lexicon. Hence, the exploitation of sentiment linguistic knowledge based on the contextual features information is an inherently challenging task.

More recently, the deep learning-based representation models have demonstrated incredibly powerful competitiveness in most NLP tasks [10–12]. With no exception, deep neural networks have dominated the task of text sentiment classification due to its ability to effectively learning text context features, including convolutional neural network (CNN) [13] and recurrent neural network (RNN) based on long short-term memory (LSTM) [14]. As CNN and LSTM give priority to the locality and sequentiality of text respectively, these deep learning-based representation models can effectively capture syntactic and semantic information in original consecutive word sequences. However, they are limited by the quality of word embedding learning algorithms [15]. As the initialization module of deep neural network, the word-embedding algorithms can transform each word form raw text data into a dense low-dimensional vector space with the contextual information from the corpus, which contains the grammatical and semantic information of words. Nevertheless, the most serious disadvantage of word embedding algorithms is that they only focus on the contexts of words but ignore sentiment linguistic knowledge. As a result, words with opposite sentiment polarities, such as ‘sad’ and ‘happy’, are mapped into close vectors in the embedding space. To solve this problem, Tang et al. introduced a supervised objective function to optimize word embedding based on the sentiment polarity label marked by the training corpus [16]. However, it is difficult to collect large-scale manually labeled corpus to ensure the reliability of word vector training. In addition, the use of deep neural networks alone is insufficient to cope with the complexity of text sentiment classification. Accordingly, Fu et al. used sentiment lexicon as external knowledge pre-training a word sentiment classifier to

generate a sentiment-based word embedding and then combined the sentiment embedding with its original word embedding to enhance the quality of word representation [17]. But the performance of this word sentiment classifier depends more heavily on the quality of sentiment lexicon. Therefore, it is essential to explore how to integrate external knowledge into deep neural network to further enhance the text representation ability of deep neural network.

To alleviate the aforementioned limitations, in this paper, we propose a novel lexicon-enhanced attention network based on text representation for the task of sentiment classification, which bridges the gap between sentiment linguistic knowledge and deep learning methods. More specifically, we first leverage the sentiment lexicon as the supplemental information attending to context words in the word embedding module via an attention mechanism. With the aid of sentiment lexicon, the attention mechanism can automatically assign different sentiment weights to the original word embeddings according to the context semantics. This sentiment-based word embedding can not only solve the word-level semantic confusion through the interactive learning characteristic of attention mechanism, but also not completely rely on the quality of feature engineering. Then, we introduce a multi-head attention mechanism to settle the sentence-level semantic confusion by capturing the important contextual information from different representation subspaces at different positions. With this design, our model can utilize the attention mechanism associated with sentiment linguistic knowledge to learn distinguishable sentiment information from the context words. Furthermore, we hierarchically stack LAN to construct a document-level model named hierarchically lexicon-enhanced attention network (H-LAN) for text sentiment classification.

The main contributions of our work are summarized as follows:

- We propose a lexicon-enhanced attention mechanism in the word embedding module that combines the common sentiment lexicon with the attention mechanism so as to incorporate sentiment linguistic knowledge into deep learning methods.
- We introduce a multi-head attention mechanism in the deep neural network module to learn the contextual information from different representation subspaces at different positions, so that our model can generate the high-quality representation of text.
- We stack an attention-based neural network to construct a document-level hierarchical model for sentiment classification. On the basis of hierarchical structure, we can effectively integrate sentiment linguistic knowledge into document-level representation for large-scale text sentiment classification.
- We conduct extensive experiments to evaluate the performance of text sentiment classification on four real-world datasets at both the sentence level and the document level. The experimental results demonstrate that the proposed models can achieve substantial improvements over the state-of-the-art baselines on 3 out of 4 datasets.

The remainder of this paper is organized as follows. Section 2 briefly introduces the related work on text sentiment classification task. Section 3 describes the details of the proposed methodology. Section 4 shows the results of the experiment with the proposed models. Section 5 draws the conclusions and discusses the future direction of development.

2. Related Work

In this section, we overview the mainstream representation models for text sentiment classification in terms of the knowledge and information they used. We briefly summarize the traditional statistical machine learning-based models in Section 2.1 and the current popular deep learning-based models in Section 2.2. We then describe the application of attention mechanism in text representation method in Section 2.3.

2.1. Statistical Machine Learning-Based Model

Traditional statistical machine learning-based representation models mainly focus on classification algorithms and feature engineering. Specifically, the conventional approaches for text sentiment analysis use the typical features such as bag-of-words [8], n-grams [7], and TF-IDF [9] as the input of machine learning algorithms such as Naïve Bayes classifier (NB) [18], Decision Tree (DT) [19], and Support Vector Machine (SVM) [20] for classification. In terms of sentiment classification, these text features are mostly designed based on statistical word frequency information of sentiment-related words derived from resources such as sentiment lexicon [21]. For instance, Kang et al. proposed to improve the performance of Naïve Bayes classifier for sentiment analysis of restaurant reviews by designing a high-quality sentiment lexicon, which can directly affect the text representation capability based on n-grams feature [22]. Hu et al. developed an improved Decision Tree model with the help of a sentiment lexicon, where co-occurrence sentiment-related words are reflected in the form of tree structure [23]. More typically, Support Vector Machine converted the raw text into a high-dimensional text representation vector, using the following word frequency features: n-grams, lexicon features, and part-of-speech tags, thus defeated all strong competitors in the SemEval-2014 task [24]. Accordingly, it is easy to see how the sentiment lexicon becomes a gold mine of sentiment linguistic knowledge to benefit the task of sentiment classification.

Although statistical machine learning-based representation models have achieved a comparable performance in the task of text sentiment classification, its shortcomings are obvious. Firstly, these methods only focus on word frequency features and completely ignore the context structure information of text, so that it is difficult to capture the semantics of words. Secondly, the success of these statistical machine learning approaches generally relies heavily on laborious feature engineering work and massive extralinguistic resources. Unlike these models, our model uses the sentiment lexicon as an auxiliary information to automatically learn the semantic relations between context words.

2.2. Deep Learning-Based Model

In recent years, it has been clearly shown that a shift of the state-of-the-art from statistical machine learning to deep learning based on text representation models [25,26]. The deep learning-based representation model is mainly to develop an end-to-end deep neural network to extract contextual features from the raw text. In terms of improving the quality of word embedding module, Mikolov et al. proposed a Word2Vec method that converted each word in the text from traditional one-hot representation to low-dimensional distributed representation, which can encode the different syntax and semantic features of each word into each dimension in a vector space [27]. Pennington et al. learned the word embedding with a comprehensive training of the global word–word co-occurrence of statistical data from the corpus, which shows an interesting linear substructure in word embedding space like Word2Vec [28]. Tang et al. designed a sentiment-based word embedding by encoding sentiment information of texts together with contexts of words, which can distinguish the opposite sentiment polarity of words in similar contexts [16]. On the basis of these improved word embedding modules, Kim adopted a CNN architecture for sentence classification, which can capture the local feature from different positions of a sentence [29]. Similarly, Zhang et al. designed a character-level CNN for text sentiment classification [30]. Furthermore, Yenter and Verma proposed a fused deep neural network for text sentiment analysis by the use of combined kernel from multiple branches of CNN with LSTM, which can achieve a good performance on the Internet Movie Database review sentiment dataset [31]. Among deep learning-based representation models, RNN has been the mainstream research method for text sentiment classification due to its ability to naturally model sequential correlation in the text. For example, Cho et al. employed RNN to learn the order of the words in the original text, which can process the raw text into a sequence of words [32]. Tang et al. proposed a deep neural network based on LSTM to model text representation for sentiment classification [33]. To further enhance the text representation capability of LSTM, Tai et al. introduced the tree structure into LSTM and treated root as text representation [34]. Analogously, Qian et al. integrated linguistic regularization into bidirectional

LSTM network to learn the high-quality representation of text [35]. Fu et al. took the sentiment lexicon as external knowledge to train the sentiment-based word embedding as the input of LSTM network [17].

A promising result has been achieved by incorporating external knowledge into deep neural networks, but the aforementioned models are time-consuming and depend heavily on parsers which are unable to guarantee an accurate syntactic structure. Our work is in line with these deep learning-based representation models, the major difference is that we incorporate sentiment linguistic knowledge as a flexibility integral part into deep neural network and learn the contextual information from different representation subspaces at different positions to generate a powerful semantic representation of text.

2.3. Attention Mechanism

The seminal NLP work using the attention mechanism is proposed by Bahdanau et al. to be used in the task of machine translation [36], where different attention weights are scored based on source words to implicitly learn the alignment for translation. The main idea of attention mechanism is to stimulate real-world human's attention reaction, that is, a man asked to understand the meaning of a sentence will selectively concentrate on parts of context words that contribute to the overall meaning after reading this sentence. Subsequently, the attention mechanism has been applied to all kinds of NLP tasks. Yang et al. proposed a hierarchical attention network to attend differently to more and less important content when processing the text for document-level classification [37]. Zhou et al. designed an attention-based LSTM network to construct the long-distance dependencies between words for bilingual text [38]. Lin et al. raised a target-specific attention network, which can extract the core part of a given text when referring to different targets, and realizes the integration of target-specific information into stance classification task [39].

Inspired by the characteristic of attention mechanism, we utilize the attention mechanism as a bridge to perfectly incorporate sentiment linguistic knowledge into deep neural networks. With this architecture, our model can explicitly reveal the importance of context evidences with regard to the sentiment polarity of text and enrich the sentiment-informative representation learning of text for sentiment analysis.

3. Methodology

In this section, we introduce the details of the LAN and H-LAN models for text sentiment classification task. The task definition and notations are first introduced. Then, we present the concrete design of our models in the following subsections.

3.1. Task Definition and Notation

Given a sentence $S = [s_1, s_2, \dots, s_n]$ consisting of n words, the aim of text sentiment classification is to predict the sentiment polarity of the given sentence. The architecture of the proposed LAN model as shown in Figure 1, which consists of three key components: the word embedding module based on lexicon-enhanced attention mechanism, the deep neural network module based on multi-head attention mechanism and the text classifier module.

We notate the meaning of notions used in this paper. In the word embedding module based on lexicon-enhanced attention mechanism, the sentiment lexicon is a set of words such as 'dreadful', 'pleasant', and 'violence', each of which is defined as a fixed negative or positive label that represents its sentiment polarity. Such information can serve as auxiliary features to improve the representation ability of word embedding. In this work, we implement the deep neural network module with some basic neural units, mainly including LSTM and multi-head attention mechanism. In particular, LSTM is based on an adaptive gating mechanism to control the data flow from and to their memory cell units. The multi-head attention mechanism can expand the text representation into three feature vectors (i.e., Q , K , V_S) through linear layers, and apply multiple layers of scaled dot-product attention to learn the contextual information from different representation subspaces at different positions. Through concat and linear layers, the text representations of different subspaces are combined to form the final

text representation $\tilde{S}$ as the input of text classifier. For the text classifier, softmax is a normalized logic function that can get the predicted probability distribution for each sentiment class.

Figure 1. The architecture of the proposed lexicon-enhanced attention network (LAN) model.

3.2. Word Embedding Module Based on Lexicon-enhanced Attention Mechanism

Above all, we do some preprocessing on the original input to improve the quality of the raw text, mainly including deleting the stopwords and segmentation. First, we remove the stopwords (e.g., ‘of’, ‘in’, and ‘from’) from the raw input sequence. Second, we split each sequence text into words by using spaces and punctuation (e.g., ‘,’ and ‘?’). Then, we input the processed data into the first layer of our LAN model.

The first layer of our LAN model is the word embedding layer. Let $L \in R^{|V| \times d}$ be an embedding lookup table initialized with the pre-trained GloVe embedding [28], where $|V|$ is the size of the word vocabulary and d is the dimension of word vectors. Given the input sentence $S = [s_1, s_2, \dots, s_n]$ where n is the number of words in the sentence, we can retrieve the word vectors from L to get the context matrix $W^C = [w_1, w_2, \dots, w_n] \in R^{n \times d}$ where w_i is the word vector of the i^{th} word with d dimension. In this way, we can encode each raw word in the sentence into a real-value vector in the high-dimensional space. As we all know, the word embedding module is the basis of the model in NLP task, which plays an important role in the ability of text representation. Therefore, in order to further improve the quality of the word embedding module, we construct the novel lexicon-enhanced attention mechanism to help select the crucial sentiment-resource-relevant context words for aggregating sentiment-informative text representation.

Specifically, we extract positive-sentiment words and negative-sentiment words respectively from the sentiment lexicon as sentiment resource words. And the sentiment lexicon is made up of SentiWordNet [40] and the extended version of Affective Norms of English Words [41]. Then, our attention mechanism leverages these two types of sentiment resource words extracted from the sentiment lexicon as an attention source attending to the context words to explicitly learn the sentiment-aware word embedding module. In the following, we take the positive-sentiment resource words as an example to elaborate the lexicon-enhanced attention mechanism in detail.

First, similar to the general matrix of word vectors, the positive-sentiment resource sequence can retrieve the word vectors from L and form the matrix $W^P = [w_1^P, w_2^P, \dots, w_m^P] \in R^{m \times d}$ where m is the number of positive-sentiment resource words in the sentiment lexicon.

Second, in order to exploit the word-level relationship between the positive-sentiment resource words and the context words, we define the dot product operation among the positive-sentiment resource words and the context words as a correlation matrix. Formally, the specific formulation is described as follows:

$$M^P = W^C \cdot (W^P)^T \quad (1)$$

where $\cdot$ stands for the dot product operation and $M^P \in R^{n \times m}$ is the correlation matrix to measure the relevance degree between the positive-sentiment resource words and the context words.

Third, based on the correlation matrix, we can calculate the sentiment-word-relevant context word representation matrix X_P^C with the dot product operation between the correlation matrix M^P and the positive-sentiment resource words W^P . Mathematically, the detailed formula is presented as follows:

$$X_P^C = (M^P \cdot W^P)^T \quad (2)$$

where $X_P^C \in R^{d \times n}$ is the context word representation matrix related to the positive-sentiment resource words. Meanwhile, we can also obtain the context-word-relevant positive-sentiment word representation matrix X^P generated by the dot product operation between the context words W^C and the correlation matrix M^P :

$$X^P = (W^C)^T \cdot M^P \quad (3)$$

where $X^P \in R^{d \times m}$ represents the positive-sentiment resource word representation matrix associated with the context words. The illustration of sentiment-context word correlation is shown in Figure 2. For the computational complexity, it can be seen from Figure 2 that the correlation matrix M^P is calculated by one-way multiplication action. Hence, the complexity of M^P is proportional to the dimension of the corresponding matrixes, which can be expressed as $O(ndm)$. Similarly, the complexity of X_P^C and X^P is based on the one-way multiplication action of M^P , i.e., $O(mnd^2)$.

Figure 2. Sentiment-context word correlation.

As with the positive-sentiment resource words, we can obtain the negative-sentiment resource word representation matrix $W^N \in R^{k \times d}$, the sentiment-word-relevant context word representation matrix $X_N^C \in R^{d \times n}$ and the context-word-relevant negative word representation matrix $X^N \in R^{d \times k}$ where k is the number of negative-sentiment resource words in the sentiment lexicon. The final feature-enhanced context word representation matrix X^C is calculated as follows:

$$X^C = X_P^C + X_N^C \quad (4)$$

Next, we employ the attention mechanism to interactively learn the correlation between the feature-enhanced context words and the context-word-relevant positive-sentiment words. In detail, the attention weight function β is computed as follows:

$$t_P = \frac{\sum_{i=1}^m X_i^P}{m} \quad (5)$$

$$\beta([X_{P_i}^C; t_P]) = u_P^T \tanh(W_P[X_{P_i}^C; t_P]) \quad (6)$$

where t_P represents the overall representation of positive-sentiment resource words by the mean-pooling operation towards X^P , β measures the importance degree of the i^{th} word $X_{P_i}^C$ in the context, W_P and u_P^T are learnable parameters. According to the attention weight function β , our attention mechanism can generate the attention vector α_i^P which indicates the importance of the i^{th} word in the context. The detailed formulation is defined as follows.

$$\alpha_i^P = \frac{\exp(\beta([X_{P_i}^C; t_P]))}{\sum_{i=1}^n \exp(\beta([X_{P_i}^C; t_P]))} \quad (7)$$

After obtaining the attention vector α_i^P , we can attend it to the context words by:

$$x_{P_i} = \alpha_i^P X_{P_i}^C \quad (8)$$

where x_{P_i} is the i^{th} word of the raw input sentence influenced by the positive context words. Similarly, with the context-word-relevant negative-sentiment words X^N as an attention source, we can obtain the x_{N_i} that is the i^{th} word of the raw input sentence influenced by the negative context words. Finally, we combine x_{P_i} with x_{N_i} as follows:

$$x_i = x_{P_i} + x_{N_i} \quad (9)$$

The final output sequence of the sentiment-aware word embedding module is $X = [x_1, x_2, \dots, x_n]$ where x_i denotes the i^{th} word of the final comprehensive sentence representation.

3.3. Deep Neural Network Module Based on Multi-head Attention Mechanism

Although the word embedding module based on lexicon-enhanced attention mechanism can solve the problem of word-level semantic confusion to some extent with the help of the interactive learning of attention mechanism, the generated text representation is insufficient for text sentiment analysis. Hence, we introduce a multi-head attention mechanism into the deep neural network to capture the important contextual information from different representation subspaces at different positions. And the network structure is mainly composed of the bidirectional LSTM network and the multi-head attention mechanism as shown in Figure 1.

We use a bidirectional version of LSTM network to process the output sequence $X = [x_1, x_2, \dots, x_n]$ generated by the sentiment-aware word embedding module, which can capture information from both the past context and the future context. The forward $\overrightarrow{LSTM}$ layer reads the sentence X from x_1 to x_n and the backward $\overleftarrow{LSTM}$ layer processes the sentence in reverse sequence order. The final hidden state h_t is the composition of the forward hidden state $\overrightarrow{h}_t$ and the backward hidden state $\overleftarrow{h}_t$ as follows:

$$\overleftarrow{h}_t = \overleftarrow{LSTM}(x_t, \overleftarrow{h}_{t-1}) \quad (10)$$

$$\overrightarrow{h}_t = \overrightarrow{LSTM}(x_t, \overrightarrow{h}_{t-1}) \quad (11)$$

$$h_t = [\overleftarrow{h}_t \oplus \overrightarrow{h}_t] \quad (12)$$

where $\oplus$ is the element-wise sum operation. The final output of the bidirectional LSTM network is $H = [h_1, h_2, \dots, h_n]$ where n denotes the length of the original input sequence. Taking the forward $\overrightarrow{LSTM}$ as an example, the detailed processing of each $\overrightarrow{LSTM}$ unit is described as follows:

For each word token x_t in the output sequence of the sentiment-aware word embedding module, the forward $\overrightarrow{LSTM}$ network can generate a corresponding hidden state vector $\overrightarrow{h}_t$ by an adaptive gating mechanism that takes control of the data flow from the past to the future. This adaptive gating mechanism consists of a forget gate f_t which controls how long certain information are stored in the

current unit, an input gate i_t which controls the input degree of new information into the current unit, and an output gate o_t which controls how much the information held in the current unit influences the output activation of the block. Mathematically, the hidden state vector $\vec{h}_t$ is calculated as follows:

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f) \quad (13)$$

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \quad (14)$$

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \quad (15)$$

$$\tilde{c}_t = \tanh(W_c[h_{t-1}, x_t] + b_c) \quad (16)$$

$$c_t = i_t \odot \tilde{c}_t + f_t \odot c_{t-1} \quad (17)$$

$$\vec{h}_t = o_t \odot \tanh(c_t) \quad (18)$$

where σ is the sigmoid function, $\odot$ denotes the element-wise multiplication, W_f , W_i , W_o , and W_c are the learnable weight matrixes for their corresponding gated units and the intermediate cell, respectively, while b_f , b_i , b_o , and b_c are their corresponding bias vectors. Following the same procedure of the forward $\overrightarrow{LSTM}$ network, the backward $\overleftarrow{LSTM}$ network can generate a corresponding hidden state vector $\overleftarrow{h}_t$. The detailed structure of the basic LSTM unit can be seen from Figure 3.

Figure 3. The architecture of the long short-term memory (LSTM) cell used in the LAN model.

Then, we employ the multi-head attention mechanism to encode hidden states of the context sequence from different representation subspaces at different positions. Taking the hidden states H as input, the multi-head attention mechanism as shown in Figure 1 converts it to three matrices through linear layers and feeds them into the scaled dot-product attention layers. Formally, the detailed formulation is as follows.

$$\text{Attention}(Q, K, V_S) = \text{softmax}\left(\frac{QK}{\sqrt{d}}\right)V_S \quad (19)$$

$$Q = HW_Q \quad (20)$$

$$K = HW_K \quad (21)$$

$$V_S = \frac{\sum_{i=1}^n h_i}{n} \quad (22)$$

where $W_Q, W_K \in R^{d \times d}$ are the projection matrixes of the linear layers and $\sqrt{d}$ denotes the scale factor that can avoid overly large values of the matrix multiplication especially when the dimensionality is high. With this design, the projection can generate different representation subspaces for the hidden states H so that the model has different interactions at different positions. It is worth noting that V_S is a

mean-pooling operation towards the hidden states H , which can record the overall historical sentiment information in the sentence. This is just like the query vector adopted by the memory network [42], our model can explicitly learn the text representation and effectively avoid sentence-level semantic confusion under the supervision of V_S . With the multi-head attention mechanism, the final sentence representation $\tilde{S}$ takes the following form:

$$\tilde{S} = \text{Concat}(\text{head}_1, \text{head}_2, \dots, \text{head}_h)W_O \quad (23)$$

$$\text{head}_i = \text{Attention}(Q, K, V_S) \quad (24)$$

where head_i represents the i^{th} head of the multi-head attention mechanism, h is the number of head in the multi-head attention mechanism, $W_O \in R^{d \times d}$ denotes the projection matrix of the linear layer, and Concat is the composition of multiple heads.

3.4. Hierarchical Lexicon-Enhanced Attention Network

When faced with large-scale text, i.e., document-level sentiment classification, we design a hierarchical classification model based on LAN. Supposing that the input document $D = [S_1, S_2, \dots, S_D]$ where D is the number of sentences and the i^{th} sentence S_i with length n is represented as $[s_1^i, s_2^i, \dots, s_n^i]$. As shown in Equation (23) in Section 3.3, we can apply LAN to encode the sentence representation from S to $\tilde{S}$. Similarly, we can use the same LAN to encode the representations $[S_1, S_2, \dots, S_D]$ of all sentences in a document into $[\tilde{S}_1, \tilde{S}_2, \dots, \tilde{S}_D]$. With the sentence-level LAN to represent the sentence as the input of the proposed model, we can employ the hierarchical structure to obtain the document-level text representation for sentiment classification. Let name this model hierarchical lexicon-enhanced attention network (H-LAN). The whole hierarchical architecture of H-LAN is shown in Figure 4.

Figure 4. The architecture of the proposed hierarchical lexicon-enhanced attention network (H-LAN) model.

The H-LAN model has the same architecture as the LAN model. And the only difference is that the input module of H-LAN is the sentence embedding processed by LAN instead of the lexicon-enhanced word embedding module in LAN. Specifically, the H-LAN model first uses the bidirectional LSTM network to encode the output of sentence embedding. The detailed formulas are described as follows:

$$\overleftarrow{h}_i^{sent} = \overleftarrow{LSTM}(\tilde{S}_i, \overleftarrow{h}_{i-1}) \quad (25)$$

$$\overrightarrow{h}_i^{sent} = \overrightarrow{LSTM}(\tilde{S}_i, \overrightarrow{h}_{i-1}) \quad (26)$$

$$h_i^{sent} = \left[\overleftarrow{h_i^{sent}} \oplus \overrightarrow{h_i^{sent}} \right] \quad (27)$$

where h_i^{sent} is the hidden state of the i^{th} sentence S_i in the document. The final output of the bidirectional LSTM network is $H^{sent} = [h_1^{sent}, h_2^{sent}, \dots, h_D^{sent}]$.

Following the same procedure of the multi-head attention mechanism as described in Section 3.3, we apply the multi-head attention mechanism to encode hidden states of the sentence sequence from different representation subspaces at different positions. The final document representation $\tilde{D}$ can be computed as:

$$\tilde{D} = \text{Concat}(\text{head}_1^{sent}, \text{head}_2^{sent}, \dots, \text{head}_h^{sent}) W_O^{sent} \quad (28)$$

$$\text{head}_i^{sent} = \text{Attention}(H^{sent} W_Q^{sent}, H^{sent} W_K^{sent}, V_D) \quad (29)$$

$$V_D = \frac{\sum_{i=1}^D h_i^{sent}}{D} \quad (30)$$

where V_D is a mean-pooling operation towards the hidden states H^{sent} , which can record the overall historical sentiment information in the document.

At last, we feed the final document representation $\tilde{D}$ into a softmax classifier to obtain the probability distribution of sentiment classes.

3.5. Text Classifier Module

Taking the final sentence representation $\tilde{S}$ as an example, we pass it into a softmax layer to predict the sentiment class distribution of a sentence by:

$$p = \text{softmax}(W_S \tilde{S} + b_s) \quad (31)$$

where p is the predicted probability distribution for each sentiment class, W_S is the learnable weight parameter and b_s is the corresponding bias parameter.

For model optimization, the training objective is to minimize the cross-entropy loss function of the true and predicted class distributions with L_2 regularization. The loss function is defined as follows:

$$L(\theta) = -\frac{1}{N} \sum_{i=1}^N y_i \log p_i + \lambda \|\theta\|^2 \quad (32)$$

where N is the number of training examples, y_i is the ground truth distribution of the i^{th} training example, $\lambda \|\theta\|^2$ is the L_2 regularization term, λ is the coefficient of L_2 regularization and θ is the parameter set of our model to be optimized during the training process. The final document representation $\tilde{D}$ follows the same procedure of sentence representation for sentiment classification. And the implementation details of our proposed models can be referred to Section 4.2.

4. Experiments

In this section, we first describe the four popular benchmark datasets and the experimental evaluation metrics in Section 4.1, and then, we present the implementation details of our experiments in Section 4.2. Afterwards, we introduce the state-of-the-art methods in Section 4.3. Finally, we analyze and discuss the experimental results in Section 4.4.

4.1. Datasets and Evaluation Metrics

In order to verify the effectiveness of our proposed models, we choose two sentence-level and document-level sentiment classification datasets to conduct our experiments. The detailed statistics of

all the datasets are given in Table 1. The first two datasets are sentence-level datasets and the last two are document-level datasets.

- Movie Review (MR) collected by Pang and Lee [43] is an authoritative movie review dataset for binary sentiment classification task, in which each review contains only one sentence. The corpus has 5331 negative samples and 5331 positive samples.
- Stanford Sentiment Treebank (SST) is an extended sentence-level version of the MR dataset proposed by Socher et al. [44], where each sentence is annotated as very positive, positive, neutral, negative, or very negative. To effectively avoid the poor generalization ability caused by the uneven distribution of samples, this dataset follows the same sample distribution of the MR dataset.
- Internet Movie Database (IMDB) proposed by Maas et al. [45] is a representative dataset for document-level sentiment classification task, which contains a large amount of movie reviews and recommendations with binary classes (i.e., positive and negative). It is made up of 50,000 movie reviews, and each movie review has several sentences.
- Yelp2013 obtained from the Yelp Dataset Challenge in 2013 [46], is another popular document-level sentiment classification dataset, which includes large-scale restaurant reviews labeled with five levels of ratings from 1 to 5 (i.e., very positive, positive, neutral, negative, and very negative).

Table 1. Summary statistics of the four mainstream datasets for text sentiment classification task. Type: level of text sentiment classification task, C: number of target classes, L: average sentence length, Train/Dev/Test: train/development/test set size, V: vocabulary size, CV: 10-fold cross validation, MR: Movie Review; SST: Stanford Sentiment Treebank; IMDB: Internet Movie Database; Yelp2013: Yelp Dataset Challenge in 2013.

Dataset	Type	C	L	Train	Dev	Test	V
MR	Sentence	2	21	10,662	-	CV	20,191
SST	Sentence	5	18	8544	1101	2210	17,836
IMDB	Document	2	126	25,000	-	25,000	106,249
Yelp2013	Document	5	179	62,522	7773	8671	73,691

For the experimental evaluation, we adopt Accuracy and F-measure as the metrics to measure the performance of text sentiment classification. Specifically, the Accuracy metric is the standard competition criteria used in most NLP tasks at present, which reflects the difference between the measured result and the target value. The specific calculation process is as follows:

$$\text{Accuracy} = \frac{\sum_{i=1}^k f(\text{predict}(i), \text{ground_truth}(i))}{k} \quad (33)$$

where k is the total number of test samples, $f(a, b)$ is a sign function (i.e., $f(a, b) = 1$ when a is equal to b ; otherwise, $f(a, b) = 0$), $\text{predict}(i)$ represents the predicted sentiment label for sample i and $\text{ground_truth}(i)$ corresponds to the true value of sample i .

In order to evaluate the experiment from a comprehensive perspective, the F-measure is introduced to measure the accuracy of the assessment analysis, which returns the weighted harmonic average value of Recall and Precision. The concrete calculation formula is as follows:

$$F\text{-measure} = \frac{2 \times \text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}} \quad (34)$$

where *Recall* is the number of samples correctly categorized divided by the total number of samples, and *Precision* is the number of samples correctly categorized divided by the total number of samples predicted to be a certain category.

4.2. Implementation Details

For the initialization phase of the model, we preprocess all the datasets by tokenization and removing stopwords with the Stanford's CoreNLP [47]. In particular, we do not remove stopwords on sentence-level datasets (i.e., MR and SST) after cleaning and tokenizing the original text, as the length of a sentence is very short. Compared with the general Word2Vec [27] approach, GloVe [28] takes into account the context global information, so the words initialized by the GloVe approach contain more comprehensive information. Hence, we use 200-dimensional GloVe word embedding as initialization for the word embedding and sentiment resource words. Out-of-vocabulary words are initialized with a uniform distribution $U(-0.1, 0.1)$. In order to show the generalization ability of our models, the GloVe embedding is not fine-tuned during the training process. When there are not sentiment resource words in the sentence, we take the context words as sentiment resource words to implement the self-attention mechanism. In addition, we set the maximum number of sentences in the document to be 30.

For the setup of deep neural network, the detailed parameters are defined as follows. In order to prevent the redundancy of dimensions, we set the number of units in each LSTM cell of the bidirectional LSTM network equal to the dimension of word embedding, i.e., 200. And the specific hyperparameter about the number of LSTM units is described in Section 4.4.2. For the multi-head attention mechanism, the number of heads is set to be 8. All the learnable weight matrices are initialized with a uniform distribution $U(-0.1, 0.1)$ and the corresponding bias vectors are initialized to zero.

For the training phase, we use the Adam optimizer [48] to train our models' parameters with an initial learning rate of 10^{-3} and the mini-batch size of 64. We tune other parameters and set the coefficient λ of L_2 regularization as 10^{-5} and the dropout rate as 0.5.

For the experimental framework, our proposed models are implemented using PyTorch and are trained on GPU Tesla K80.

4.3. Baseline Methods

To comprehensively measure the effectiveness of our proposed models, we compare the LAN and H-LAN models with several strong baseline methods used in the task of text sentiment classification, including conventional machine learning method, popular deep learning methods, deep learning methods based on specified parsing structure, deep neural networks relying on attention mechanism, and hierarchical neural networks.

- SVM [24]: A classic support vector machine classifier applied in text sentiment classification, which is trained with the following groups of features: n-grams, lexicon features, and part-of-speech tags.
- CNN [29]: Convolutional neural network consists of multiple one-dimensional convolution filters and a max pooling layer. This structure can capture the local features inside a multi-dimensional field to generate context-aware text representation.
- LSTM [32]: The standard recurrent neural network based on long short-term memory unit uses the last hidden state as the representation of the whole text, which can extract sequential correlation information in the raw text.
- CNN-LSTM [31]: A deep neural network combines kernel from multiple branches of CNN with LSTM, which can extract local features within the raw text and sequence information across the raw text.
- Tree-LSTM [34]: Tree-structured long short-term memory network utilizes pre-defined syntax tree structure as external knowledge to guide the training of LSTM network. It can better learn the representation of text from the syntax perspective.
- LR-Bi-LSTM [35]: Linguistically regularized-based bidirectional long short-term memory network is similar to Tree-LSTM, except that linguistic regularization is used as external knowledge to the intermediate outputs of bidirectional LSTM network with KL divergence.

- LE-LSTM [17]: Lexicon-enhanced LSTM network which uses sentiment lexicon as an extra information to train a sentiment-based word embedding and then combines the sentiment embedding with its word embedding to make word representation more accurate.
- LSTM-GRNN [49]: Hierarchical neural network which consists of a LSTM-based word encoder and a GRU-based sentence encoder for document-level sentiment classification task.
- Self-attention [50]: A self-attention mechanism and a special regularization term are integrated into deep neural networks to construct sentence embedding for sentiment classification task.
- HAN [37]: Hierarchical attention network for document-level sentiment classification which progressively builds a document vector aggregating important words into a sentence and then aggregating important sentences vector into document vectors.

4.4. Experimental Results

In this subsection, we first report the model comparison for sentiment classification task at both the sentence level and the document level. After that, we explore the effects of the dimension of sentiment resource words and the length of document on the proposed models. Furthermore, we visualize the sentence and document representations generated by the models to verify the validity of text representation for sentiment analysis.

4.4.1. Overall Performance

As shown in Table 2, the experimental results present all discussed models in this work for text sentiment classification on MR, SST, IMDB, and Yelp2013, respectively. We can draw the observations from Table 2 as follows.

Table 2. Evaluation results (%) of our models and state-of-the-art models on different datasets. The best result for each dataset is in bold. Results marked with # are retrieved from the references (i.e., [17,29,31,35,37,44,49]), while those unmarked with # are obtained either by our own implementation or with the same codes shared by the original authors. SVM: Support Vector Machine; CNN: convolutional neural network; Tree-LSTM: tree-structured long short-term memory; LR-Bi-LSTM: linguistically regularized-based bidirectional long short-term memory; LE-LSTM: lexicon-enhanced LSTM; LSTM-GRNN LSTM-based word encoder and a GRU-based sentence encoder; HAN: hierarchical attention network.

Models	MR		SST		IMDB		Yelp2013	
	Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1
SVM	76.4	78.8	40.7 #	42.4	79.8	81.7	57.6	59.4
CNN	81.5 #	82.7	48.0 #	49.3	87.0 #	88.3	59.7 #	61.4
LSTM	77.4 #	79.6	46.4 #	47.6	86.2 #	88.7	62.0	63.6
CNN-LSTM	-	-	-	-	89.5 #	90.2	-	-
Tree-LSTM	80.7 #	82.2	50.1 #	51.8	88.0 #	89.3	64.6	66.9
LR-Bi-LSTM	82.1 #	83.6	50.6 #	52.3	-	-	-	-
LSTM-GRNN	-	-	-	-	88.2	89.5	65.1 #	67.2
LE-LSTM	80.8 #	82.4	-	-	89.6 #	90.9	60.6 #	62.7
HAN	-	-	-	-	88.6	89.8	68.2 #	70.1
Self-attention	81.7	82.9	48.9	50.1	-	-	-	-
H-LAN	-	-	-	-	89.5	90.6	68.7	70.8
LAN	83.9	85.0	51.3	52.9	-	-	-	-

For sentence-level sentiment classification, the experimental results are listed in the left half of Table 2. Firstly, we observe that the traditional machine learning-based representation model (i.e., SVM) performs poorly in sentence-level sentiment classification. The main reason is that statistical machine learning classifier only focuses on word frequency features without considering any textual semantics. Secondly, compared with SVM on two datasets, LSTM improves the accuracy by roughly 3.4% and F1

by roughly 3.0%. Similarly, the accuracy and F1 of CNN on two datasets are increased by roughly 6.2% and 5.4%, respectively. The results show that the semantic structure learned by deep neural network is helpful to the performance of text sentiment analysis. Furthermore, when incorporating external knowledge into deep neural network, LR-Bi-LSTM improves upon the basic performance of LSTM by 4.5% on accuracy and 4.4% on F1 averagely. Tree-LSTM is slightly weaker than CNN on the MR dataset but shows a promising result on the fine-grained SST dataset. Our LAN model is the same as the spirit of LR-Bi-LSTM, but with a simple attention mechanism to capture salient sentiment-related words in sentences rather than relying on complex parsers like LR-Bi-LSTM. As can be seen from this table, the proposed LAN model outperforms LR-Bi-LSTM on two datasets by roughly 1.3% on accuracy and roughly 1.0% on F1. Our LAN model outperforms LE-LSTM on MR dataset by 3.1% on accuracy and 2.6% on F1, they all use the sentiment lexicon as external knowledge, which verifies that LAN combines the sentiment lexicon with attention mechanism to generate more efficient sentiment-based word embedding. Moreover, it is worth noting that deep neural network with attention mechanism (i.e., Self-attention) is comparable with CNN. Compared with Self-attention, our LAN model achieves a relatively better performance, which indicates that deep neural network with our proposed attention mechanisms can enrich the representation of sentences. According to these results, we can draw a conclusion that the proposed LAN model is effective for the task of text sentiment classification.

For document-level sentiment classification, we can observe from the right half of Table 2 that the performances are limited without a hierarchical structure (i.e., SVM, CNN, LSTM, and Tree-LSTM). This is because the text is organized and structured in document-level sentiment analysis but these methods process the whole document as a long sequence of words. SVM is the worst performer in the baseline models because it simply counts co-occurrence sentiment-related words regardless of the semantic structure of a document. As an exception CNN-LSTM performs well on IMDB dataset, which is related to its complex network structure generated by the combination of CNN and LSTM. Compared with CNN and LSTM, Tree-LSTM makes a relatively better result, but does not surpass the performance of HAN. This demonstrates that the hierarchical structure of deep neural network can adequately represent the raw text at the document level of sentiment classification task. Our proposed H-LAN model uses a similar hierarchical structure as that in HAN, but with an additional attention mechanism to highlight the important sentiment-related context words in the sentence and capture the contextual information from different representation subspaces at different positions. The improvement is more prominent on two datasets in which H-LAN outperforms Tree-LSTM by roughly 2.8% on accuracy and roughly 2.5% on F1. Furthermore, when compared HAN with the proposed H-LAN, we can find an improvement of 0.7% on accuracy and 0.8% on F1 averagely. This demonstrates that our model can enhance the representation of document by incorporating sentiment linguistic knowledge into hierarchical deep neural network. Although LE-LSTM has the best performance on IMDB dataset, our H-LAN model gets a comparative result on this dataset and outperforms LE-LSTM on Yelp2013 dataset by a large margin, which shows that our H-LAN model can capture more accurate fine-grained sentiment features from the document-level text. According to above results, we can conclude that our proposed model is a benefit to the task of document-level sentiment classification.

4.4.2. Effects of the Dimension of Sentiment Resource Words

Since the sentiment resource words are a vital component of our LAN model, we undertake a further investigation on the performance of LAN with different dimensions of sentiment resource words. In Figure 5, we show the classification accuracy of our discussed model for sentence-level sentiment classification on MR and SST datasets, respectively. Note that since the dimension of word vector is equal to the number of LSTM units, this experiment also directly reflects the fine-tuning process of the number of units in the LSTM network.

Clearly, for these two datasets, we can observe that our LAN model obtains high accuracies on sentiment resource words with moderate dimension, but performs relatively worse when the dimension of sentiment resource words is too low or too high. This is because the word vector

with low dimension usually contains less meaningful information, and the high-dimensional word vector implies more redundant information which will hinder the optimization process of model parameters. More specifically, on MR dataset, our LAN model shows a significant descending trend when the dimension of sentiment resource words exceeds 200. It demonstrates that the simpler binary sentence-level sentiment classification is extremely sensitive to the redundant data in the word vector, which affects the model to capture other sentiment-related knowledge so as to learn a poor-quality text representation for sentiment analysis. However, when the dimension of sentiment resource words exceeds 200 in SST dataset, the performance of LAN first decreases and then converges until the dimension arrives at 300. The main reason for this result is that LAN has fully utilized the potential of sentiment resource words when the dimension reaches at 200, and the parameters of LAN are in the stage of local fine-tuning when performing on the word vector with dimension from 200 to 300. Therefore, it is reasonable to set the dimension of sentiment resource words to 200 with the aim of ensuring the generalization ability of the proposed LAN model.

Figure 5. Classification accuracy of the proposed LAN model on two sentence-level sentiment classification datasets. (a) MR; (b) SST. When the dimension of sentiment resource words is equal to 0, it means that the lexicon-enhanced attention mechanism is not equipped with the word embedding module.

4.4.3. Effects of the Length of Document

To investigate the impact on performance of attention-based hierarchical neural networks over documents with different number of sentences, we design the experiment according to the number of sentences and then compare the performance of H-LAN and HAN on groups of documents with various sentence numbers. Figure 6 depicts the classification accuracy of both models on IMDB and Yelp2013 datasets, respectively.

From Figure 6a, we can observe that the accuracy of these two models presents a sharp upward trend when the number of sentences changes from (0, 5] to (15, 20]. As documents with the sentence number from (15, 20] to (25, 30], both models show a stable descending trend on IMDB dataset. The main reason for this result is that, fewer sentences usually contain less valuable information contributing to the orientation of sentiment analysis and documents with a large number of sentences contain more complex correlation between sentences so that it is difficult to learn a high-quality document representation from large-scale unstructured text. In particular, compared with HAN, our H-LAN model has a relatively small improvement when the number of sentences exceeds 20, which potentially demonstrates that H-LAN has a robust receptive field on the text sequences. As shown in Figure 6b, it is interesting to find the similar uptrend until the number of sentences arrives at (5, 10]. However, the trends in the accuracy rate of all discussed models show a significant fluctuation when performing on documents with sentence number from (5, 10] to (20, 25]. The reason is that too long documents contain too many noisy words, so that it can greatly mislead the prediction of fine-grained text sentiment classification. After that, H-LAN and HAN present a steady downward trend in terms

of accuracy before the number of sentences reaches (25, 30]. Specifically, comparing with H-LAN and HAN, it can be observed that H-LAN improves the accuracy of HAN by a large margin. This strongly proves that our H-LAN model has a powerful receptive ability to alleviate the impact of gradient vanishing [51] caused by the architecture of recurrent neural network. To sum up the above observation, we can conclude that the proposed H-LAN can effectively retain the semantic structure of the raw document to improve the performance of sentiment analysis.

Figure 6. Classification accuracy of the proposed H-LAN model and HAN on two document-level sentiment classification datasets. (a) IMDB; (b) Yelp2013.

4.4.4. Visualization of Sentence Representation

In order to show the proposed attention mechanisms can accurately extract sentiment-related words contributing to the orientation of sentiment polarity, we visualize the attention values of the word embedding module of our LAN model in Figure 7. We select a comment sentence with double contrasting sentiment polarities from the MR dataset as a case study, which expresses negative sentiment towards a film.

Figure 7. Visualization of sentence representation learned by the word embedding module of our LAN model. (a) initial attention visualization result; (b) Final attention visualization result. The attention score used for color-coding is calculated by Equation (7). The depth of color represents the importance degree of a word to the sentiment polarity of a sentence.

As can be seen in Figure 7a, the LAN model gives a relatively uniform distribution of attention weights when initializing the training phase of model. The reason for this result is that, the initial weight parameters of the word embedding module in LAN are set by random sampling from a uniform distribution, just as the general deep learning-based representation models do. It also does mean that each word in the sentence contributes roughly the same degree to sentiment polarity. Generally, deep learning-based representation models do not carry out special processing on the word

embedding module and input it directly into the deep neural network after initialization, which will mislead the judgment of deep learning methods that only use contextual information for sentiment classification. In contrast, with the proposed lexicon-enhanced attention mechanism, LAN can capture salient sentiment-related words to guide the deep neural network to focus on the accurate semantic structure of the sentence. Figure 7b shows the final attention visualization result. We can observe that, with the progress of training, our model gradually reduces the attention weight of the sentiment word ‘good’ and gives a higher attention weight to the sentiment word ‘fails’. Meanwhile, the word ‘but’ is also assigned a higher weight of attention, which does play a decisive role in changing the sentiment attitude of this sentence. This is closely related to the multi-head attention mechanism, where our model can interactively learn the correlation between the sentiment words and the contextual information to recognize sentiment transition in the sentence. Overall, the results of visualization are consistent with the performance of human attention in the real world, which partly explains the successful application of LAN in sentence-level sentiment classification task.

4.4.5. Visualization of Document Representation

To provide insight into the meaningfulness of our H-LAN model, we extract the vector representation from the penultimate layer of H-LAN and use t-SNE tool [52] to visualize them on a two-dimensional space. The visualization results of document representations learned by our H-LAN model on two sentiment classification datasets are shown in Figure 8. Each color in the figure indicates a different ground-truth sentiment class of a document and the point represents the document embedding belonging to a certain sentiment category. It can be observed that almost each document embedding with the same color falls into the internal area of the corresponding manifold, which means that these document embeddings are closely related to a certain sentiment polarity. This shows that the proposed H-LAN model can effectively learn the sentiment-aware information inherent in the document. Besides, the distinction between different sentiment classes is also obvious, which also demonstrates that the document embedding learned by H-LAN is more discriminative. From the above results, we can conclude that our H-LAN model has strong representative power of document for sentiment classification.

Figure 8. Visualization of document representation learned by our H-LAN model on two document-level sentiment classification datasets. (a) document representation learned by H-LAN on IMDB dataset; (b) document representation learned by H-LAN on Yelp2013 dataset.

5. Conclusions

In this paper, we propose a lexicon-enhanced attention network to solve the problem that deep learning-based models carry more semantic information rather than sentiment linguistic knowledge. By combining the sentiment lexicon with attention mechanism in the word embedding module, we can obtain the sentiment-aware word embeddings as the input of deep neural network, which bridges the gap between sentiment linguistic knowledge and deep learning methods. In order to capture the context

information more comprehensively, we introduce a multi-head attention mechanism in the deep neural network module which can learn the contextual information from different representation subspaces at different positions. And we further construct a hierarchical variant for document-level sentiment classification. Extensive experiments are conducted on four real-world sentiment classification datasets at both the sentence level and the document level. The experimental results show the effectiveness of our proposed models in text sentiment classification, as it can capture real sentiment information to improve the quality of text representation.

In the future, for the effective learning of context information, we plan to design a location-based attention mechanism in the word embedding module that encodes the explicit location information of words into deep neural network. And beyond the sentiment lexicon, we will try to exploit other sentiment linguistic knowledge suitable for deep learning methods to enhance the representation ability of text for sentiment classification.

Author Contributions: Conceptualization, W.L. and H.Y.; methodology, W.L.; software, W.L. and D.L.; validation, L.Z., P.L. and Z.Z.; formal analysis, D.L. and L.Z.; investigation, W.L.; resources, P.L.; data curation, D.L. and Z.Z.; writing—original draft preparation, W.L.; writing—review and editing, W.L.; visualization, D.L. and L.Z.; supervision, P.L.; project administration, Z.Z.; funding acquisition, P.L. and Z.Z.

Funding: This work was supported by the National Natural Science Foundation of China (Grant No. 61373148), the National Social Science Foundation of China (Grant No. 19BYY076), the Science Foundation of the Ministry of Education of China (Grant No. 14YJC860042), and the Shandong Provincial Social Science Planning Project (Grant No. 18CXWJ01/18BJYJ04/17CHLJ33/17CHLJ30/17CHLJ18).

Acknowledgments: The authors are grateful to the supports from the Shandong Normal University for the server used for experiments.

Conflicts of Interest: The authors declare no conflict of interest.

References

- Ren, F.; Wu, Y. Predicting User Topic Opinions in Twitter with Social and Topical Context. *IEEE Trans. Affect. Comput.* **2013**, *4*, 412–424. [\[CrossRef\]](#)
- Li, Z.; Guo, H.; Wang, W.M. A Blockchain and AutoML Approach for Open and Automated Customer Service. *IEEE Trans. Ind. Inform.* **2019**, *15*, 3642–3651. [\[CrossRef\]](#)
- Gao, Y.; Zhen, Y.; Li, H. Filtering of Brand-related Microblogs using Discriminative Social-Aware Multiview Embedding. *IEEE Trans. Multimed.* **2016**, *18*, 2115–2126. [\[CrossRef\]](#)
- Kim, Y.; Shim, K. TWILITE: A recommendation system for Twitter using a probabilistic model based on latent Dirichlet allocation. *Inf. Syst.* **2014**, *42*, 59–77. [\[CrossRef\]](#)
- Abdul, A.; Chen, J.; Liao, H.Y.; Chang, S.H. An Emotion-Aware Personalized Music Recommendation System Using a Convolutional Neural Networks Approach. *Appl. Sci.* **2018**, *8*, 1103. [\[CrossRef\]](#)
- Bing, L. Sentiment Analysis and Opinion Mining. In *Encyclopedia of Machine Learning and Data Mining*; Springer: Boston, MA, USA, 2012; Volume 30, p. 167.
- Damashek, M. Gauging Similarity with n-Grams: Language-Independent Categorization of Text. *Science* **1995**, *267*, 843–848. [\[CrossRef\]](#)
- Zhao, R.; Mao, K. Fuzzy Bag of Words Model for Document Representation. *IEEE Trans. Fuzzy Syst.* **2018**, *26*, 794–804. [\[CrossRef\]](#)
- Wu, H.C.; Luk, R.W.P.; Wong, K.F. Interpreting TF-IDF term weights as making relevance decisions. *ACM Trans. Inf. Syst.* **2008**, *26*, 1–37. [\[CrossRef\]](#)
- Lecun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436. [\[CrossRef\]](#)
- Tom, Y.; Devamanyu, H.; Soujanya, P. Recent Trends in Deep Learning Based Natural Language Processing Review Article. *IEEE Comput. Intell. Mag.* **2018**, *13*, 55–75.
- Li, H. Deep learning for natural language processing: Advantages and challenges. *Natl. Sci. Rev.* **2018**, *5*, 28–30. [\[CrossRef\]](#)
- Nal, K.; Edward, G.; Phil, B. A convolutional neural network for modelling sentences. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, Baltimore, MD, USA, 23–25 June 2014; pp. 655–665.
- Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [\[CrossRef\]](#)

15. Lin, Q.; Yong, C.; Nie, Z. Learning word representation considering proximity and ambiguity. In Proceedings of the Twenty-eighth AAAI Conference on Artificial Intelligence, Québec City, QB, Canada, 27–31 July 2014; pp. 1572–1578.
16. Tang, D.; Wei, F.; Qin, B. Sentiment Embeddings with Applications to Sentiment Analysis. *IEEE Trans. Knowl. Data Eng.* **2016**, *28*, 496–509. [[CrossRef](#)]
17. Fu, X.; Yang, J.; Li, J. Lexicon-enhanced LSTM with Attention for General Sentiment Analysis. *IEEE Access* **2018**, *6*, 71884–71891. [[CrossRef](#)]
18. Chen, J.; Huang, H.; Tian, S. Feature selection for text classification with Naïve Bayes. *Expert Syst. Appl.* **2009**, *36*, 5432–5435. [[CrossRef](#)]
19. Wawre, S.V.; Deshmukh, S.N. Sentiment classification using machine learning techniques. *Int. J. Sci. Res* **2016**, *5*, 819–821.
20. Lee, L.H.; Wan, C.H.; Rajkumar, R. An enhanced Support Vector Machine classification framework by using Euclidean distance function for text document categorization. *Appl. Intell.* **2012**, *37*, 80–99. [[CrossRef](#)]
21. Tang, D.; Wei, F.; Qin, B.; Zhou, M.; Liu, T. Building Large-Scale Twitter-Specific Sentiment Lexicon: A Representation Learning Approach. In Proceedings of the COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers, Dublin, Ireland, 23–29 August 2014; pp. 172–182.
22. Kang, H.; Yoo, S.J.; Han, D. Senti-lexicon and improved Naïve Bayes algorithms for sentiment analysis of restaurant reviews. *Expert Syst. Appl.* **2012**, *39*, 6000–6010. [[CrossRef](#)]
23. Chien, B.C.; He, S.Y. A Lexical Decision Tree Scheme for Supporting Schema Matching. *Int. J. Inf. Technol. Decis. Mak.* **2011**, *10*, 519–537. [[CrossRef](#)]
24. Kiritchenko, S.; Zhu, X.; Cherry, C.; Saif, M. Detecting aspects and sentiment in customer reviews. In Proceedings of the 8th International Workshop on Semantic Evaluation, Dublin, Ireland, 23–24 August 2014; pp. 437–442.
25. Tang, D.; Qin, B.; Liu, T. Deep Learning for Sentiment Analysis: Successful Approaches and Future Challenges. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* **2015**, *5*, 292–303. [[CrossRef](#)]
26. Tang, D.; Zhang, M. Deep Learning in Sentiment Analysis. In *Deep Learning in Natural Language Processing*; Deng, L., Liu, Y., Eds.; Springer: Singapore, 2018; pp. 219–253.
27. Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.S.; Dean, J. Distributed representations of words and phrases and their compositionality. In Proceedings of the Advances in Neural Information Processing Systems, Lake Tahoe, NV, USA, 5–10 December 2013; Volume 2, pp. 3111–3119.
28. Pennington, J.; Socher, R.; Manning, C. Glove: Global vectors for word representation. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, Doha, Qatar, 25–29 October 2014; pp. 1532–1543.
29. Kim, Y. Convolutional neural networks for sentence classification. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, Doha, Qatar, 25–29 October 2014; pp. 1746–1751.
30. Zhang, X.; Zhao, J.; Lecun, Y. Character-level Convolutional Networks for Text Classification. In Proceedings of the Advances in Neural Information Processing Systems, Montréal, QC, Canada, 7–12 December 2015; pp. 649–657.
31. Alec, Y.; Abhishek, V. Deep CNN-LSTM with combined kernels from multiple branches for IMDb review sentiment analysis. In Proceedings of the 8th IEEE Annual Ubiquitous Computing, Electronics and Mobile Communication Conference, New York, NY, USA, 19–21 October 2017; pp. 540–546.
32. Cho, K.; Merriënboer, B.V.; Bahdanau, C.F.; Bougares, H.; Bengio, Y. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, Doha, Qatar, 25–29 October 2014; pp. 1724–1734.
33. Tang, D.; Qin, B.; Feng, X. Target-Dependent Sentiment Classification with Long Short Term Memory. *Comput. Sci.* **2015**, *24*, 74–82.
34. Tai, K.S.; Socher, R.; Manning, C.D. Improved semantic representations from tree-structured long short-term memory networks. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, Beijing, China, 30–31 July 2015; pp. 1556–1566.
35. Qian, Q.; Huang, M.; Lei, J. Linguistically Regularized LSTMs for Sentiment Classification. In Proceedings of the Annual Meeting of the Association for Computational Linguistics, Vancouver, BC, Canada, 30 July–4 August 2017; pp. 1679–1689.

36. Bahdanau, D.; Cho, K.; Bengio, Y. Neural machine translation by jointly learning to align and translate. *arXiv* **2014**, arXiv:1409.0473.
37. Yang, Z.; Yang, D.; Dyer, C.; He, X.; Smola, A.; Hovy, E. 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, San Diego, CA, USA, 12–17 June 2016; pp. 1480–1489.
38. Zhou, X.J.; Wan, X.J.; Xiao, J.G. Attention-based LSTM network for cross-lingual sentiment classification. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Austin, TX, USA, 1–5 November 2016; pp. 247–256.
39. Gui, L.; Du, J.; Xu, R. Stance Classification with Target-specific Neural Attention. In Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, Melbourne, Australia, 19–25 August 2017; pp. 3988–3994.
40. Baccianella, S.; Esuli, A.; Sebastiani, F. SentiWordNet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In Proceedings of the International Conference on Language Resources and Evaluation, Valletta, Malta, 17–23 May 2010; pp. 83–90.
41. Warriner, A.B.; Kuperman, V.; Brysbaert, M. Norms of valence, arousal, and dominance for 13,915 English lemmas. *Behav. Res. Methods* **2013**, *45*, 1191–1207. [[CrossRef](#)]
42. Sukhbaatar, S.; Weston, J.; Fergus, R. End-to-end memory networks. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 7–12 December 2015; pp. 2440–2448.
43. Pang, B.; Lee, L. A Sentimental Education: Sentiment Analysis Using Subjectivity, Summarization Based on Minimum Cuts. In Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics, Barcelona, Spain, 21–26 July 2004; pp. 271–278.
44. Socher, R.; Perelygin, A.; Wu, J.Y.; Chuang, J.; Manning, C.D.; Potts, C. Recursive deep models for semantic compositionality over a sentiment treebank. In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, Seattle, WA, USA, 18–21 October 2013; Volume 1631, p. 1642.
45. Andrew, L.; Maas, R.E.; Daly, P.T.; Pham, D.H.; Andrew, Y.N.; Christopher, P. Learning word vectors for sentiment analysis. In Proceedings of the 49nd Annual Meeting of the Association for Computational Linguistics, Portland, OR, USA, 19–24 June 2011; pp. 142–150.
46. Tang, D.; Qin, B.; Liu, T. 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, Beijing, China, 26–31 July 2015; pp. 1014–1023.
47. Manning, C.D.; Surdeanu, M.; Bauer, J.; Finkel, J.; Bethard, S.J.; McClosky, D. The Stanford CoreNLP Natural Language Processing Toolkit. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations, Baltimore, MD, USA, 22–27 June 2014; pp. 55–60.
48. Kinga, D.P.; Ba, J. Adam: A method for stochastic optimization. In Proceedings of the International Conference of Learning Representation, San Diego, CA, USA, 7–9 May 2015; p. 115.
49. Tang, D.; Qin, B.; Liu, T. Document modeling with gated recurrent neural network for sentiment classification. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, Lisbon, Portugal, 7–21 September 2015; pp. 1422–1432.
50. Gui, L.; Zhou, Y.; Xu, R.; He, Y.; Lu, Q. Learning representations from heterogeneous network for sentiment classification of product reviews. *Knowl. Based Syst.* **2017**, *124*, 34–45. [[CrossRef](#)]
51. Graves, A. *Long Short-Term Memory*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 1735–1780.
52. van der Maaten, L.; Hinton, G. Visualizing data using t-SNE. *J. Mach. Learn. Res.* **2008**, *9*, 2579–2605.