A sentiment information collector–extractor architecture based neural network for sentiment analysis

A Sentiment Information Collector-Extractor Architecture Based Neural Network for Sentiment Analysis

Kai Shuang[a], Hao Guo[a,*], Zhixuan Zhang[a], Jonathan Loo[b]

[a]*State Key Laboratory of Networking & Switching Technology, Beijing University of Posts and Telecommunications, 100876, Beijing, P.R.China*
[b]*School of Computing and Engineering, University of West London, W5 5RF, UK*

Abstract

Sentiment analysis, also known as opinion mining is a key natural language processing (NLP) task that receives much attention these years. Deep learning based neural network models have achieved great success in it. However, the existing deep learning models cannot effectively make use of the sentiment information in the sentence for sentiment analysis. In our model, we apply a bi-directional Long Short Term Memory structure based sentiment information collector to collect the sentiment information in the sentence, which may collect information more completely compared with other types of neural network. Then we also apply an ensemble model of sentiment information extractor to combine the results of these sub-extractors and the new ensemble strategy makes our model more universal and outperforms any single sub-extractor. We conduct experiments on three datasets of different languages. The experimental results show that the proposed method outperforms the state-of-the-art methods on all datasets.

Keywords: sentiment analysis, sentiment information collector, sentiment information extractor, model ensemble

1. Introduction

Deep learning has made a great progress recently and plays an important role in academia and industry. In particularly, standard natural language processing (NLP) approaches for entity and relationship extraction are improved [1] and business-aware concept

*Corresponding author
Email addresses: shuangk@bupt.edu.cn (Kai Shuang), guo_yu_hao_1993@163.com (Hao Guo), 2228290335@qq.com (Zhixuan Zhang), jonathan.loo@uwl.ac.uk (Jonathan Loo)

detection by convolutional neural networks is proposed [2]. Based on deep neural network, new inspirations are brought to various NLP task. Recent progress in word representation provides good resources for lexical semantics [3]. Text classification is an essential component in many applications, such as sentiment analysis [4, 5] web searching and information filtering [6]. Therefore, it has attracted considerable attention in both academia and industry.

Sentiment analysis [7], also known as opinion mining [5], is a key NLP task that receives much attention these years. It refers to the process of computationally identifying and categorizing opinions expressed in a piece of text, in order to determine whether the writers attitude towards a particular topic or product is positive, negative, or even neutral. However, traditional feature representation methods for sentiment analysis often ignore the contextual word order information in texts or have the data sparsity problem which heavily affects the classification accuracy [8]. With the pre-trained word embeddings [9, 10, 11], neural networks demonstrate their great performance in sentiment analysis and many other NLP tasks.

In particularly, when classifying the sentiment polarity of a long sentence, the most essential work is to locate the key words which can indicate the sentiment polarity of the whole sentence. For examples, consider these three sentences (i) Happiness has stayed with me since I found out my own. (ii) I spent a whole day in the park, which far away from my house, in happiness. (iii) To be honest, I have not been pleasant since I was informed the terrible news. Both of sentence (i) and sentence (ii) contain the key word happiness which indicates positive emotion. However, this key word appears in two completely different positions. Besides, sentence (iii) contains two key words not and pleasant and they are separated by another word been. These two words together can indicate the sentiment polarity of the sentence. How to locate the key words remains a big challenge in sentiment analysis.

Researchers have designed many efficient models in order to capture the sentiment information. For example, Recurrent Neural Network (RNN) which includes Long Short Term Memory (LSTM), Gated Recurrent Unit(GRU) and so on is one of the most popular models. Standard RNN has the gradient vanishing or exploding problems. In order to overcome the issues, LSTM was developed and achieved superior performance [12]. The

model analyzes a text word by word in the order of they appear in the text and stores the semantics of all the previous text in a fixed-sized hidden layer [13] The advantage of RNN is the ability to better capture the contextual information. This could be beneficial to capture semantics of long texts. However, the RNN is a biased model, where later few words are more dominant than the earlier words [6]. Thus, it could reduce the effectiveness when RNN is used to capture the semantics of a whole sentence, because key components could appear anywhere in a sentence rather than at the end. For examples, in sentence (i) the key word happiness appears in the front of the sentence and the same key word appears in the back of the sentence (ii). The two key words not and pleasant appear in the middle of sentence (iii). When these three types of sentence are fed into RNN in the order of words appear in the sentence, the sentence (ii) will have the best performance comparing with the others.

Besides, Convolutional Neural Network (CNN) which is an unbiased model can fairly determine discriminative phrases in a text with a max-pooling layer. However, CNN network itself has a characteristic of local connection [14]. Previous studies on CNNs tends to apply CNN to analyze the local contextual information of a sentence [15, 16]. For example, some researchers take the results of word embedding as CNN input, each convolution window contains information of a few words in the sentence which means the outputs of the convolution layer are based on local information in the sentence. Although followed max-pooling layer can help extract information, but this result is mainly based on the local information output from the convolution layer. In this way, when using CNN to deal with long sentences, it is difficult to analyze the contextual information of the entire sentence. In order to cope with the existing problems and capture the key words that indicate the sentiment polarity, we propose a sentiment information collector-extractor architecture based neural network (SICENN) for text classification First, the bidirectional long short term memory (BLSTM) structure [17, 18] is applied as a Sentiment Information Collector (SIC) to generate sentence information matrix which contain all the contextual information of the sentence. Second, sentiment key words will be automatically extracted from sentence information matrix and the emotional polarity will be extracted by our Sentiment Information Extractor (SIE).

BLSTM has the ability to better capture the contextual information. BLSTM is an

unbiased model, because the output of BLSTM is a sentence vector at each time-step. Each sentence vector emphasizes the information around it. In other words, the output of the BLSTM at each time-step is a sentence vector which contains one particular aspect of information of the sentence that can also be regarded as a particular feature of the sentence. The SIE in our model stacks the vectors generated at each time step into a sentence information matrix, which contains all the features of the sentence, and feeds it into the SIE. The SIE aims at extracting the contextual information related to sentiment polarity from the sentence information matrix. Three sub-extractors are applied to extract the sentiment information respectively and model ensemble approach is used to combine and process the outputs of the three sub-extractors. Based on model ensemble theoryexperiment results show that, our ensemble SIE will outperform any sub-extractor.

To summarize, our contributions are as follows:

- Based on the characteristics of BLSTM structure, SIC is designed, which can collect the sentiment information in the sentence completely.
- Based on the model ensemble strategy, SIE is designed, which can extract the sentiment information precisely from the outputs of the SIC.
- Experiments are set up to validate the accuracy of our SICENN model, and the results show that our model outperforms previous state-of-the-art approaches and can better capture the sentiment information in the sentence.

2. Related Work

Deep learning based neural network models have achieved great success in many NLP tasks in the past few years, including learning distributed word, sentence and document representation [11], parsing [19], statistical machine translation [20], sentence classification [16, 21], etc. Learning distributed sentence representation through neural network models can reach satisfactory results in related tasks like sentiment classification, text categorization. Among the neural network models, CNN and RNN are two most popular models and the variants of these models are applied in sentiment analysis recently. For CNN, a multichannel CNN model [16] is proposed to increase the accuracy for sentence classification, but each convolution window contains information of a few words in the sentence

which means the outputs of the convolution layer are only based on local information in the sentence. For RNN, gated neural networks [22] is proposed to capture the influence of the surrounding words when performing sentiment classification of entities. LSTM is developed [12] and achieved more superior performance then both tradition RNN structure and GRU [23]. But LSTM is still a biased model, where later few words are more dominant than the earlier words [6]. In order to overcome the weakness of LSTM, BLSTM is applied to sentiment analysis [24] by researchers and outperforms the traditional LSTM.

CNN and RNN models can be applied to sentiment analysis task individually and they can also be combined properly to improve the performance on classification. Although there are many previous models [6, 25, 26] combining CNN & RNN, they may not make best use of the ability for CNN & RNN to collection and extract sentiment information base on the characteristics of CNN and RNN. For example, the Recurrent Convolutional Neural Network (RCNN) model in [6] didn't make best use of RNN and CNN. The bi-directional recurrent structure used in RCNN model is similar to BLSTM structure but concatenates word embedding vector with sentence vector, which may make the accuracy for sentiment classification decline. There is only a linear transformation together with the tanh activation function after bi-directional recurrent structure which cannot extract the sentence information effectively. The weakness of RCNN model will be explained more thoroughly in Section 4.4.1.

The idea of using neural networks in an ensemble has been proposed previously in [27, 28, 29]. An ensemble of residual nets is applied to image recognition [30]. The ensemble model can combine the results of different individual sub-models, which makes the whole model learn the characteristic of the datasets better and outperform all the sub-models. In these paper, the SICENN is proposed, by make full use of CNN and BLSTM using in an ensemble. Based on our proper ensemble strategy, accuracy on sentiment classification is improved further.

3. Model

In this section, we will introduce our model in details. Figure 1 shows the architecture of the whole model. As is illustrated in Figure 1, the model can be divided into two part: (i) SIC and (ii) SIE.

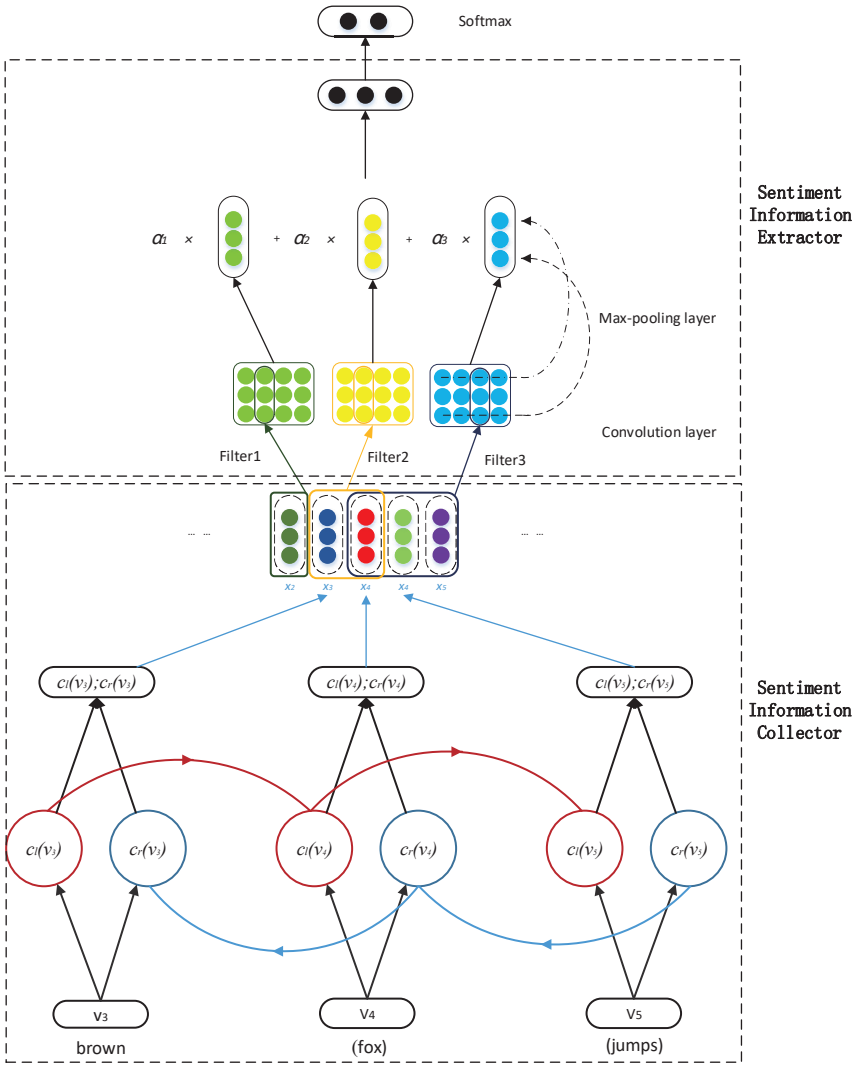

Figure 1: The architecture of SICENN

The input of the model is a sentence consisting of a sequence of word expressed as word vectors v_1, v_2 v_n. All sentiment information will be collected through information collector based on the characteristics of BLSTM and the output is a sentence information matrix X consisting of sentence vectors x_1, x_2 x_n. Then the matrix X is fed into information extrac-
130 tor and latent semantic information will be extracted based on model ensemble strategy.

The details of these two parts are explained in the following subsections.

3.1. Sentiment Information Collector (SIC)

We first describe the architecture of the SIC in our model. SIC is designed to collect the sentiment information of whole the sentence and generate an unbiased sentence information matrix. The structure we used to design SIC is BLSTM, which is an unbiased bidirectional LSTM structure and have the characteristics to collect the sentiment information completely.

We define v_i for the i-th word, x_i for the output of LSTM at the i-th time step and L for the maximum number of words in a sentence. Then we define $c_l(v_i)$ as the left context of word v_i and $c_r(v_i)$ as the right context of word v_i. Both $c_l(v_i)$ and $c_r(v_i)$ are dense vectors with $|c|$ real value elements.

The left-side context $c_l(v_i)$ of word v_i is calculated using Equation(1), where $e(v_i)$ is the word embedding of word v_i, which is a dense vector with $|e|$ real value elements. $c_l(v_{i-1})$ is the left-side context of the previous word v_i. The left-side context for the first word in any document uses the same shared parameters $c_l(v_1)$. W_l is a matrix that transforms the hidden layer (context) into the next hidden layer. W_{sl} is a matrix that is used to combine the semantic of the current word with the next words left context. f is a nonlinear activation function. The right-side context $c_r(v_i)$ is calculated in a similar manner, as shown in Equation (2). The right-side contexts of the last word in a document share the parameters$c_r(v_L)$.

$$c_l(v_i) = f(W_l c_l(v_{i-1}) + W_{sl} e_l(v_i)) \tag{1}$$

$$c_r(v_i) = f(W_r c_r(v_{i+1}) + W_{sr} e_r(v_i)) \tag{2}$$

The model can also reserve a larger range of the word ordering when learning representations of texts. As shown in Equations (1) and (2), the context vector captures the semantics of all left-side and right-side contexts. So v_i is more dominant than other words both in $c_l(v_i)$ and $c_r(v_i)$. For example, for the sentence The quick brown fox jumps over the lazy dog, $c_r(fox)$ encodes the semantics of the left-side context The quick brown fox and $c_r(fox)$ encodes the semantics of the right-side context fox jumps over the lazy dog. Then, we define x_i as the representation of the sentence which emphasized the meaning of

word v_i at time step i in Equation (3), which is the concatenation of the left-side context vector $c_l(v_i)$ and the right-side context vector $c_r(v_i)$.

$$x_i = [c_l(v_i); c_r(v_i)] \tag{3}$$

As a result, x_i vector contains the whole information of the sentence but emphasized the information around the i-th word. As L stand for the maximum number of words in a sentence, the model will generate L different x_i, and each x_i is a biased vector that emphasized the information around the i-th word. Finally, the model stack the output sentence vector x_i at each time step and generate the unbiased sentence information matrix X, which collects all the features of the sentence.

$$X = [x_1;\ x_2;\ x_3; \ldots;\ x_L] \tag{4}$$

Therefore, matrix X contains all the sentiment information and other noises, and the sentiment information need to be extract from matrix X.

3.2. Sentiment Information Extractor (SIE)

The information extractor, which is an ensemble model, is designed to extract sentiment information precisely from sentence information matrix X. The SIE consists of three sub-extractors. Each sub-extractor extract the sentiment information independently and the outputs of the sub-extractors are combined based on the model ensemble strategy.

Let N be the length of x_i, each sub-extractor applies an information extraction layer using convolution operation as is shown in Equation (5), and j stands for the width of the filters, where $j \in \{1, 2, 3\}$ and the length of filter is equal to N.

$$m_i^j = f w_i^j\ x_i + b^j \tag{5}$$

where b is a bias term and f is a non-linear function such as the sigmoid, hyperbolic tangent, etc. In our case, we choose ReLU [31] as the nonlinear function. m_i^j is a latent semantic vector, in which each semantic factor will be analyzed to determine the most useful factor for representing the text.

When all of the latent semantic vectors m_i^j are calculated separately, each sub-extractor will apply a max-pooling operation:

$$m^j = \max_{i=1}^{L} m_i^j \tag{6}$$

The max function is an element-wise function. The element m_i^j is the maximum in the L elements of m_i^j, $i \in \{1, 2, ..., L\}$ as is shown in Figure 1. The pooling layer converts texts with various lengths into a fixed-length vector. With the max-pooling layer, we can capture the information throughout the entire text and find out the most important latent semantic factors in the document.

The last part of our model is an output layer. We combine outputs of the three sub-extractors based on model ensemble approaches:

$$y = \alpha_1 m^1 + \alpha_2 m^2 + \alpha_3 m^3 \tag{7}$$

where α_1, α_2 and α_3 are trainable parameters of weight that automatically determine which size of information extraction window is more important based on the results of training. When using single fixed size of information extraction window for convolution layer, the same window size may have different performance in different datasets, and window size need to be changed in order to suit different datasets. However, in our ensemble model, three different window sizes, which have the best performance comparing to other window sizes among different datasets, are chosen for extracting the sentiment information. Then three trainable weights α_1, α_2 and α_3 are applied to combine the results of these sub-extractors and they change automatically based on the characteristic of the datasets which makes our model more universal and outperforms using convolution layer with single fixed window size.

Finally, the Softmax function is applied to y. It can convert the output numbers into probabilities.

$$p = \frac{\exp(y_i)}{\sum_{k=1}^{n} \exp(y_i)} \tag{8}$$

4. Experiment

For datasets, we take both English dataset and Chinese dataset to validate our model, as different language may have different sentence structure, which can validate our model in different aspect.

For word-embedding method, we initialize word vectors with those obtained from an unsupervised neural language model [11].

We perform a series of experiments to validate our model for sentiment analysis. First we perform some experiments to make a clear description that how hyper-parameter settings influence the final results and how we chose the critical hyper-parameters. Second, we compare our model with currently state-of-the-art approaches and prove the accuracy promotion for sentiment analysis. Finally, we reach a conclusion that the new model we designed can collect sentiment information more completely and extract sentiment information more precisely and the classification accuracy outperforms previous state-of-the-art approaches

4.1. Datasets

For English, we have downloaded some reviews from Amazon about daily necessities which are the data source of Zhang X et al in [32], which spans 18 years with $34,686,770$ reviews from $6,643,669$ users on $2,441,053$ product. Two classification tasks are constructed from this dataset one predicting full number of stars the user has given, which is called $Amazon5$ in the following paper, and the other predicting a polarity label by considering star 1 negative, star 3 neutral and star 5 positive, which is called $Amazon3$ in the following paper. The $Amazon5$ dataset and the $Amazon3$ dataset contains $45,000$ training samples and $5,000$ testing samples in each class, and the samples are randomly selected from the origin data source.

For Chinese, we take microblogs as the source of corpus, as the short (140 characters limit), noisy and various nature of microblogs make it contain a wealth of emotional information which is very suitable for sentiment analysis. We have crawled microblogs from Sina microblog website (http://weibo.com/) which has grown to be a major social media platform with hundreds of millions of users in China. The total number of microblog records is about $5,000,000$. We cut off some records whose emotional tendencies are not obvious and there are $3,000,000$ samples left. $45,000$ positive samples and $45,000$ negative samples are randomly selected as training samples while $5,000$ positive samples and $5,000$ negative samples are randomly selected as testing samples, which is called $SinaMicroblog$ in the following paper.

We regard these three datasets as a benchmark to evaluate different models and explore the influence of parameters in the following experiments.

4.2. Pre-training and Word Embedding

There is no blank in a Chinese sentence which is different from English, so preprocessing work must be done at first to separate each sentence into several words which is called word segment and in our work we use an open source tool called JieBa[33] to conduct it. After the word segment, the whole sentence is transformed into a sequence of Chinese words.

The word-vector generator aims at generating distributed representation of each word. Initializing word vectors with those obtained from an unsupervised neural language model is a popular method to improve performance in the absence of a large supervised training set [34, 15, 35]. We use the publicly available *word2vec* tools that were trained on reviews from Amazon and *SinaMicroblog* for English and Chinese respectively. The vectors have dimensionality of 300 and were trained using the continuous skip-gram architecture [11]. Words not present in the set of pre-trained words are initialized randomly.

4.3. Experiment Settings

The models are trained by min-batch back propagation with optimizer RMSprop [36] which is usually a good choice for LSTM. The batch size chosen in the experiment is 128 and gradients are averaged over each batch. Parameters of the model are randomly initialized over a uniform distribution with [-0.5, 0.5]. We set the number of kernels of convolution layers all as 200 with different window sizes and also set the number of hidden units in BLSTM as 200. For regularization we use dropout [37] with probability 0.5 on the last Softmax layer within all models. We train our model on training set with enough epochs to obtain the best performance of accuracy on testing samples.

4.4. Results and Discussions

n our SICENN model, the structure of SIC is a fixed structure based on the BSLTM model. However, the structure of SIE is more flexible. Three critical factors that influence the effectiveness of SIE are explored in our following experiments.

- The sizes of information extraction windows in sub-extractors.
- The depth of sub-extractors.
- The model ensemble strategy used to combine sub-extractors.

4.4.1. Size of information-extracting windows

In order to extract sentiment information from the sentence information matrix more precisely, the sizes of information-extracting windows need to be carefully chosen. We perform a group of experiments to show the classification accuracy using only one information-extracting windows size on each dataset.

Table 1: Accuracy of different sizes of information-extracting windows

Size of information-extracting windows	$Amazon5$	$Amazon3$	$SinaMicroblog$
RCNN	57.30%	81.74%	83.63%
1	57.32%	81.82%	83.72%
2	57.54%	81.78%	83.78%
3	57.48%	81.75%	83.75%
4	57.38%	81.68%	83.70%
5	57.23%	81.59%	83.65%

Table 1 provides detailed accuracy information for each method in different dataset, where Amazon 5 represents reviews from amazon contain five categories, $Amazon3$ represents views from amazon contains 3 categories and $SinaMicroblog$ contain 2 categories. RCNN refers to the model that Siwei proposed in [6].

When we set 1 as the size of information-extracting windows, the entire structure of the model is similar to RCNN except the inner structure of BLSTM as we explained in Model. Table 1 shows that the accuracy when the window size set as 1 is higher than the RCNN model on different datasets, which indicates that the BLSTM structure in our model is more scientific and efficient. Because the outputs of our SIC are the concatenation of the left-side context vectors and the right-side context vectors as is shown in Equation (3). While in RCNN model, they define the representation of word x_i as the concatenation of the left-side sentence vector $c_l(v_i)$, the word embedding $e(v_i)$ and the right-side sentence vector $c_r(v_i)$ [6]. However, word embedding $e(v_i)$ is a pre-trained vector containing the semantic information of words, while sentence vectors $c_l(v_i)$ and $c_r(v_i)$ are the outputs of BLSTM containing the contextual information. So concatenate the word embedding with the two sentence vectors will not promote the accuracy for text classification, or even worse, it may bring noises into the model and reduce the accuracy.

By comparing the accuracy of different sizes of information-extracting windows as $1, 2, 3, 4$ and 5, the accuracy of window sizes as $1, 2, 3$ are better than that as $4, 5$ in every dataset. For $Amazon5$ dataset window size 2 for conclusion reach the accuracy of 57.54% and better than the other window sizes. For $Amazon3$ and $SinaMicroblog$ window size of 1 and 2 have the best performance respectively. The experiments results show that the same window size have different performance in different datasets, which indicates the necessity to use ensemble strategy and combine the advantages of different window sizes. Besides, when the window sizes increase lager than 2, the accuracy declines with the window size become larger. As a result, we apply the ensemble strategy in the SIE and sizes of information-extracting window in its sub-extractor are set as 1, 2, and 3 separately.

4.4.2. Depth of sub-extractors

The depth of the sub-extractors is determined by the number of information-extracting layers, which can influence the accuracy for classification. We have performed a series of experiments to explore how the depth of the sub-extractors influences the accuracy in the SIE.

Table 2: Accuracy of different Number of information-extracting layers

Number of information-extracting layers	$Amazon5$	$Amazon3$	$SinaMicroblog$
one	57.32%	81.82%	83.72%
two	56.34%	81.47%	82.77%
three	55.80%	81.75%	82.02%

Table 2 shows that all the accuracy for classification on above three datasets have the similar tendency. The model with one information-extracting layer have the best performance in all datasets, that is to, say SIE cannot extract more useful information from the outputs of SIC by increasing the depth of sub-extractors. It is clear that the sub-extractor with more information-extracting layers contains more parameters and has a lager solution space than that with fewer layers, but more layers will also bring much difficulty to optimizer with backward propagation strategy. So it can be known that there is a trade-off between the depth of model and the difficulty of optimization. The experiments results show that one layer just stands at a balance point. As increasing the depth of sub-extractors

cannot improve the accuracy for classification. The model ensemble strategy is essential for improve the performance of the information extractor and improve the accuracy for classification.

4.4.3. Model ensemble strategy

Model ensemble strategy can directly impact the effectiveness of the SIE and influence the results of sentiment classification. We combine outputs of the three sub-extractors based on model ensemble approaches by applying three trainable parameters α_1, α_2 and α_3 as is shown in Equation (7).Because the parameters in neural network are updated by iteration and search for the local optimal, so the initialization of these trainable parameters can influence the accuracy of sentiment classification. We performs a series of experiments to explore the proper strategy to initialize the trainable parameters and construct an effective ensemble SIE.

Table 3: Accuracy of different model ensemble strategy

Weights Initialization	$Amazon5$	$Amazon3$	$SinaMicroblog$
randomly	57.57%	81.89%	84.01%
1,1,1	57.62%	81.94%	84.14%
1,0,0		82.46	
0,1,0	58.12%		84.36%

By comparing Table 1 and Table 3, we can discovery that the SIE with model ensemble strategy outperforms the all the sub-extractor. Besides, the SICENN model can reach a better accuracy if we initial the weights properly based on the results of Table 1 on different datasets.

Table 3 shows the accuracy of different initial value of weights for each dataset. For example, randomly indicates initializing α_1, α_2 and α_3 randomly, (1, 1, 1) indicates initializing α_1, α_2 and α_3 with the same weight and (1, 0, 0) indicates initialing α_1, α_2 and α_3 by 1, 0, 0 respectively. We firstly initial α_1, α_2 and α_3 randomly. The results show that when we initial them with the same weight can improve the classification accuracy among all the datasets.

Based on the results of Table 1, our ensemble model initial the weight variables which

is multiplied with the sub-extractor of the best performance as 1 and initial other weights as 0. For example, we initial weights variables in $Amazon5$ as 0 ,1, 0because the extractor whose size of information extraction windows is 2 has the best performance among all the single window size, as is shown in Table 1. Thus the weights initialization strategy of (1, 0, 0) will not be applied on $Amazon5$. Table 1 shows that the accuracy of best performance on $Amazon5$ is 57.54%. When we initial the weights of our ensemble model as Table 3 shows, we set α_2 as 1 and set α_1 and α_3 as 0, the accuracy of our SICENN model on $Amazon5$ reach 58.12%. Because the training process of a neural network is to search the local optimal solution by iteration and the local optimal solution may not be the best solution, when we put more weights on the best sub-extractor and the training process can reach a better solution.

4.5. Comparison of Methods

We compare our method with widely-used artificial neural network for sentiment analysis including Siweis [6] model, which model has been compared with other state-of-the-art model.

Table 4: Comparison of Methods

Model	$Amazon5$	$Amazon3$	$SinaMicroblog$
CNN[16]	54.90%	80.14%	82.34%
LSTM [21]	54.72%	80.46%	82.56%
CNN & LSTM[25]	55.03%	80.57%	82.99%
BLSTM[18]	56.94%	80.86%	82.16%
RCNN [6]	57.30%	81.74%	83.63%
SICENN	58.12%	82.46%	84.36%

Table 4 provides detailed accuracy information for different methods in different datasets, where CNN, LSTM, CNN & LSTM and RCNN refer to the existing model proposed in the corresponding reference. By comparing the model results of CNN, LSTM and CNN & LSTM, we observe that the accuracy of CNN & LSTM model performs better than CNN model and LSTM in all the datasets. For example, in $Amazon5$ datasets, the accuracy of

CNN and LSTM are 54.90% and 54.72% respectively. The accuracy of CNN & LSTM can reach 55.03%.

The BLSTM model has a better performance in Amazon dataset. We can observe that the accuracy in *Amazon*5 dataset using BLSTM model can reach 56.94%, much higher than that of CNN & LSTM (55.03%). The accuracy in *Amazon*3 dataset using BLSTM model is a little higher than that of CNN & LSTM. But the results in *SinaMicroblog* dataset are quite different from that in Amazon datasets. The accuracy of BLSTM model is 82.16% lower than that of CNN & LSTM model (82.99%), which indicates that the effectiveness by using only BLSTM model without any changes cant outperform the CNN & LSTM model.

The RCNN model improves the accuracy apparently comparing with the CNN & LSTM model and BLSTM model. The accuracy in *Amazon*5, *Amazon*3, *SinaMicroblog* using RCNN model can reach 57.30%, 81.74%, and 83.63% respectively. However, our model outperforms any state-of-art methods in each dataset as is shown in Table 4. The accuracy in *Amazon*5 datasets using our SICENN model can reach 58.12%, which has an improvement of 0.82% comparing with that of RCNN. The improvements in *Amazon*3 and *SinaMicroblog* are 0.72% and 0.73% respectively comparing our SICENN model with RCNN model.

5. Conclusion and future work

We propose Sentiment Information Collector-Extractor architecture based neural network for sentiment classification. The experiment results validate the effectiveness of BLSTM structure for collecting contextual information and demonstrate the SIE we designed can extract more information and promote text classification accuracy by combining different window sizes through model ensemble theory comparing to using any single window size. The experiment results on various datasets also demonstrate our model outperforms previous state-of-the-art approaches.

In the future, we will explore how to combine the semantic information of the sentence and the characteristics of the person who speak the sentence. We may build more sophisticated ensemble models and may involve more structures, such as attention model, to extract the sentiment information in the sentence more precisely.

References

[1] Y. Wang, H. Ma, N. Lowe, M. Feldman, C. Schmitt, Business event curation: merging human and automated approaches, in: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, AAAI Press, 2016, pp. 4272–4273.

[2] B.-C. Chen, Y.-Y. Chen, F. Chen, D. Joshi, Business-aware visual concept discovery from social media for multimodal business venue recognition., in: AAAI, 2016, pp. 101–107.

[3] C. Liang, P. Paritosh, V. Rajendran, K. D. Forbus, Learning paraphrase identification with structural alignment, in: Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, AAAI Press, 2016, pp. 2859–2865.

[4] B. Pang, L. Lee, Opinion mining and sentiment analysis, Foundations and trends in information retrieval 2 (1-2) (2008) 1–135.

[5] B. Liu, Sentiment analysis and opinion mining, Synthesis lectures on human language technologies 5 (1) (2012) 1–167.

[6] S. Lai, L. Xu, K. Liu, J. Zhao, Recurrent convolutional neural networks for text classification., in: AAAI, Vol. 333, 2015, pp. 2267–2273.

[7] T. Nasukawa, J. Yi, Sentiment analysis:capturing favorability using natural language processing, in: International Conference on Knowledge Capture, 2003, pp. 70–77.

[8] M. Post, S. Bergsma, Explicit and implicit syntactic features for text classification, in: Meeting of the Association for Computational Linguistics, 2013, pp. 866–872.

[9] Y. Bengio, A. Courville, P. Vincent, Representation learning: a review and new perspectives, IEEE Transactions on Pattern Analysis & Machine Intelligence 35 (8) (2012) 1798.

[10] T. A. Mikolov, Statistical language models based on neural networks.

[11] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, in: Advances in neural information processing systems, 2013, pp. 3111–3119.

[12] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural computation 9 (8) (1997) 1735–1780.

[13] J. L. Elman, Finding structure in time, Cognitive science 14 (2) (1990) 179–211.

[14] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proceedings of the IEEE 86 (11) (1998) 2278–2324.

[15] R. Collobert, J. Weston, M. Karlen, K. Kavukcuoglu, P. Kuksa, Natural language processing (almost) from scratch, Journal of Machine Learning Research 12 (1) (2011) 2493–2537.

[16] Y. Kim, Convolutional neural networks for sentence classification, Eprint Arxiv.

[17] A. Graves, J. Schmidhuber, rgen, 2005 Special Issue: Framewise phoneme classification with bidirectional LSTM and other neural network architectures, Elsevier Science Ltd., 2005.

[18] A. Graves, S. Ndez, J. Schmidhuber, rgen, Bidirectional lstm networks for improved phoneme classification and recognition, in: Artificial Neural Networks: Formal MODELS and Their Applications - ICANN 2005, International Conference, Warsaw, Poland, September 11-15, 2005, Proceedings, 2005, pp. 799–804.

[19] R. Socher, A. Perelygin, J. Y. Wu, J. Chuang, C. D. Manning, A. Y. Ng, C. Potts, Recursive deep models for semantic compositionality over a sentiment treebank.

[20] I. Sutskever, O. Vinyals, Q. V. Le, Sequence to sequence learning with neural networks, in: Advances in neural information processing systems, 2014, pp. 3104–3112.

[21] D. Tang, B. Qin, T. Liu, Document modeling with gated recurrent neural network for sentiment classification, in: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, 2015, pp. 1422–1432.

[22] M. Zhang, Y. Zhang, D.-T. Vo, Gated neural networks for targeted sentiment analysis., in: AAAI, 2016, pp. 3087–3093.

[23] Y. Wang, M. Huang, X. Zhu, L. Zhao, Attention-based lstm for aspect-level sentiment classification, in: Conference on Empirical Methods in Natural Language Processing, 2016, pp. 606–615.

[24] M. Yang, W. Tu, J. Wang, F. Xu, X. Chen, Attention based lstm for target dependent sentiment classification., in: AAAI, 2017, pp. 5013–5014.

[25] C. Zhou, C. Sun, Z. Liu, F. C. M. Lau, A c-lstm neural network for text classification, Computer Science 1 (4) (2015) 39–44.

[26] D. Liang, Y. Zhang, Ac-blstm: Asymmetric convolutional bidirectional lstm networks for text classification.

[27] L. K. Hansen, P. Salamon, Neural Network Ensembles, IEEE Computer Society, 1990.

[28] L. K. Hansen, P. Salamon, Neural network ensembles, IEEE Transactions on Pattern Analysis & Machine Intelligence 12 (10) (2002) 993–1001.

[29] M. P. Perrone, L. N. Cooper, When networks disagree: Ensemble methods for hybrid neural networks, 1993.

[30] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Computer Vision and Pattern Recognition, 2016, pp. 770–778.

[31] V. Nair, G. E. Hinton, Rectified linear units improve restricted boltzmann machines, in: International Conference on International Conference on Machine Learning, 2010, pp. 807–814.

[32] X. Zhang, J. Zhao, Y. LeCun, Character-level convolutional networks for text classification, in: Advances in Neural Information Processing Systems, 2015, pp. 649–657.

[33] J. Sun, Jiebachinese word segmentation tool.

[34] R. Socher, J. Pennington, E. H. Huang, A. Y. Ng, C. D. Manning, Semi-supervised recursive autoencoders for predicting sentiment distributions, in: Conference on Empirical Methods in Natural Language Processing, EMNLP 2011, 27-31 July 2011, John Mcintyre Conference Centre, Edinburgh, Uk, A Meeting of Sigdat, A Special Interest Group of the ACL, 2011, pp. 151–161.

[35] M. Iyyer, P. Enns, J. Boyd-Graber, P. Resnik, Political ideology detection using recursive neural networks, in: Meeting of the Association for Computational Linguistics, 2014, pp. 1113–1122.

[36] T. Tieleman, G. Hinton, Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude, COURSERA: Neural Networks for Machine Learning 4 (2).

[37] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, R. R. Salakhutdinov, Improving neural networks by preventing co-adaptation of feature detectors, Computer Science 3 (4) (2012) 212–223.