2019 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation

LSTM Based Semi-supervised Attention Framework for Sentiment Analysis

Hanxue Ji[1,2], Wenge Rong[1,2], Jingshuang Liu[1,2], Yuanxin Ouyang[1,2], Zhang Xiong[1,2]
[1]State Key Laboratory of Software Development Environment, Beihang University, Beijing 100191, China
[2]School of Computer Science and Engineering, Beihang University, Beijing 100191, China
{jihanxue, w.rong, jingshuangl, oyyx, xiongz}@buaa.edu.cn

***Abstract*—With the rapid development of Internet technology and social media, people are accustomed to making comments on the Internet. Sentiment analysis, as an efficient technique, has been used by researchers in the tasks of analysing the sentiment polarity under these comments. To better achieve this target, the fundamental challenge is how to extract the feature and build a proper mechanism to learn them. A lot of word embedding based deep learning models for sentiment analysis are proposed in the literature. And the semi-supervised learning methods make it possible to use both labelled and unlabelled data for this kind of task. Furthermore, the attention mechanism proposed in recent years has achieved great accomplishments for natural language processing (NLP) tasks since it helps to capture the important information of the documents. In this paper, inspired by these works, we proposed a long short term memory (LSTM) based semi-supervised attention framework for sentiment analysis tasks, which is composed of an unsupervised attention based LSTM encoder-decoder and an attention based supervised LSTM model attached by a Softmax layer. The unsupervised part worked for attaining the high dimensional representation of the documents, and the supervised part extracted feature and enhanced the important parts for classification. Experimental study on commonly used datasets has demonstrated its ability for sentiment analysis tasks.**

***Index Terms*—Sentiment Analysis, Semi-supervised Learning, Attention Mechanism, Long Short Term Memory, Encoder-decoder**

I. Introduction

Aanalyzing the vast amount of valuable information and extracting the sentiment information beneath them is a significant task. In order to do this, sentiment analysis has become a kind of valid technique to investigate the sentiment polarity of the documents posted on-line [1] and gain much attention in both academic and industrial communities.

Recently, with the development of word embedding based deep learning techniques, many models have been introduced to the sentiment analysis tasks [2]–[4], among which Recurrent Neural Network (RNN) has shown its great power since it can learn the underlying relationships between the words. However, RNN based solutions still have limitations. One of them is normally called vanishing gradient problem [5]. To overcome this shortcoming, LSTM has been proposed in the literature and proven its potential in maintaining the advantages of RNN while overcoming the problems of vanishing gradient [6].

However, with the input document getting longer, it is hard for the model like RNN and LSTM to capture the most important part for the global sentiment of the document [7]. Attention mechanism is a valid method to catch the useful information in the long sentence [8]. Therefore, in our model, we use the encoded document representation and attention mechanism to enhance the weights of crucial part of input document. Then we extract the important features and classify the sentiment polarities with the model.

Besides, it is believed that the small amount of corpus in the experimental dataset is limited [9]. Importing external knowledge from different domains may help to improve the model's performance [10]. Under this assumption, we propose to design an unsupervised model to import the external messages. At present, the encoder-decoder is a powerful structure for unsupervised learning in natural language process (NLP) tasks [11]. And attention mechanism is a valid promotion to the behaviour of unsupervised learning model [8]. Therefore, we propose to train an attention mechanism based encoder-decoder using unlabelled corpus. Then we use it to encode the input documents into vectors for sentiment classification.

The main contribution of our work is that we propose a semi-supervised framework for sentiment analysis. The framework is mainly composed of two separate parts: 1)An unsupervised LSTM encoder-decoder structure based on attention mechanism working for importing external knowledge and encoding the input document into vectors. 2)A supervised LSTM network based on attention mechanism with Softmax layer for further feature extraction and classification.

The rest of the paper is organized as follows. Sect. II will present the background knowledge including feature presentation, effective model for sentiment analysis and semi-supervised learning technique. In Sect. III, we introduce the pipeline of the proposed semi-supervised framework. Then, the experiment study and discussion will be illustrated in Sect. IV. Finally Sect. V concludes the paper and points out some possible future research directions.

II. Background

A. Sentiment Analysis

Traditionally, the most simple approach for feature extraction from documents is using one-hot representation (also known as Bag-of-words) [12]. Each word is translated to vector of a single 1 with numerous 0, which is very convenient to implement. However, although it is convenient in implementation, Bag-of-words features cannot effectively

978-1-7281-4034-6/19/$31.00 ©2019 IEEE

DOI 10.1109/SmartWorld-UIC-ATC-SCALCOM-IOP-SCI.2019.00218

deliver the syntactic and semantic relations of words [13]. Two similar words may have totally different representations [13]. To overcome this limitation, low dimensional word embedding has been proposed in recent years by learning joint representation from a large quantity of documents [10]. GloVe is a weighted least squares regression model which can address the drawbacks of noisy co-occurrence in former models like skip-gram [14]:

$$J = \sum_{i,j=1}^{V} f(X_{ij})(w_i^T \tilde{w}_j + b_i + \tilde{b}_j - \log X_{ij})^2 \quad (1)$$

where V is the vocabulary size, X_{ij} is the matrix of word-word co-occurrence counts, $f(X_{ij})$ is the weighting function, w_i^T and $\tilde{w}_j$ are word vectors and separate context word vectors respectively. And b_i, $\tilde{b}_j$ are bias vector for w_i^T and $\tilde{w}_j$. In this research, we initialised our model in both the unsupervised step and the supervised step with the GloVe word embedding.

With the development of word embedding, many deep learning models based on word embedding features have been introduced to sentiment analysis. Among them the RNN, a time series model, has been thoroughly studied [3]. Due to its recurrent trait, it can perform the same task for every input word, thereby achieving success in sentiment analysis tasks [4]. In our former work, we once proposed a dual recurrent neural network which can better cover the long history memory [15]. The RNN structure has shown ability for sentiment analysis tasks, but in some cases, there are much useful information between two words which are several steps away. In this situation, it is difficult for RNN to learn the underlying relations between them because of the vanishing gradient problem [5]. As a result, the LSTM with gates and cell mechanism were developed to solve it [16].

Although LSTM can learn the underlying relationships between words it still has some problems in capturing the most important parts of the input. In recent years, with the proposing of attention mechanism, it is possible for the sequence model to catch the significant information of the input. The attention mechanism was first presented in the task of machine translation, and had been employed in other tasks like video and image analysis [8], [17], [18]. The attention mechanism helps the sequence model to focus on the most relevant and vital part in the input, which can improve the performance of the model [8]. In this paper, we choose LSTM as the basic model to learn the word embedding features extracted from the datasets. And then we set the attention mechanism to the LSTM to enhance the behaviour of our framework.

B. Semi-supervised Learning

Supervised learning has proven its success in many tasks for sentiment analysis [19]. However, in order to enhance the generalization ability of the learning model, this kind of method needs a large number of labelled corpus [20]. In order to take advantage of the large amount of unlabelled data, unsupervised learning is presented as an effective way. Combined both the unsupervised and supervised learning, semi-supervised learning is a valid method [21]. In NLP tasks, one of the semi-supervised learning technique is to use the unlabelled corpus for unsupervised learning and use the labelled corpus for supervised learning.

Recently, the encoder-decoder structure has been presented for the tasks of NLP. It is effective in the tasks of machine translation, question answering, speech recognition and text parsing [18], [22]–[24]. The structure is composed of two separate steps. In the encoding step, a sequence model such as RNN, LSTM and gated recurrent unit (GRU) is applied as an encoder to encode the input data. And in the decoding step, the encoded vector is input as the start state in the decoder, then the output sequence can be produced step by step [11].

In the encoder-decoder, catching the important input parts is always difficult. Hence the attention mechanism we introduced before can be applied to the encoder-decoder. With the attention mechanism, in the decoding process, instead of only using the single encoded vector, the decoder takes the whole encoder structure into consideration and catches the most significant information [8]. In this paper, we employ LSTM as both the encoder and decoder in the unsupervised learning step.

In our experiments, firstly, we employ an attention based LSTM encoder-decoder to import the external information. We use the model to encode the input documents. Secondly, we implement an attention based LSTM model. In this step, We input the encoded document vectors got from first step into the attention based LSTM. The document vectors works as part of the attention mechanism here. Experimental study on commonly used dataset against baseline methods has demonstrated its promising performance.

III. Methodology

The proposed sentiment analysis framework is shown in Fig. 1(The source code of the proposed approach is available at the follwing link[1]). The first part is the unsupervised model to take advantage of the unlabelled dataset and encode the input document. The second part is a supervised attention based LSTM network used to capture the important information in the input and learn the hidden features of the instance to make prediction.

A. Unsupervised learning

1) Model training: In the aim of training an encoder-decoder to make use of the unlabelled dataset and encode the input documents. We design an attention based encoder-decoder structure for unsupervised learning process. In this part, we use the LSTM network for both the encoder and the decoder.

The detail of the structure is shown in Fig. 3. The goal of the LSTM encoder is to encode the whole documents into a single vector, and the LSTM decoder in our framework works for reconstructing the input word sequence. In this model, each one-hot word in the input sequence is projected to a word embedding layer in a fixed dimension, we initialise

[1] https://github.com/xuexue1996/semi-supervised

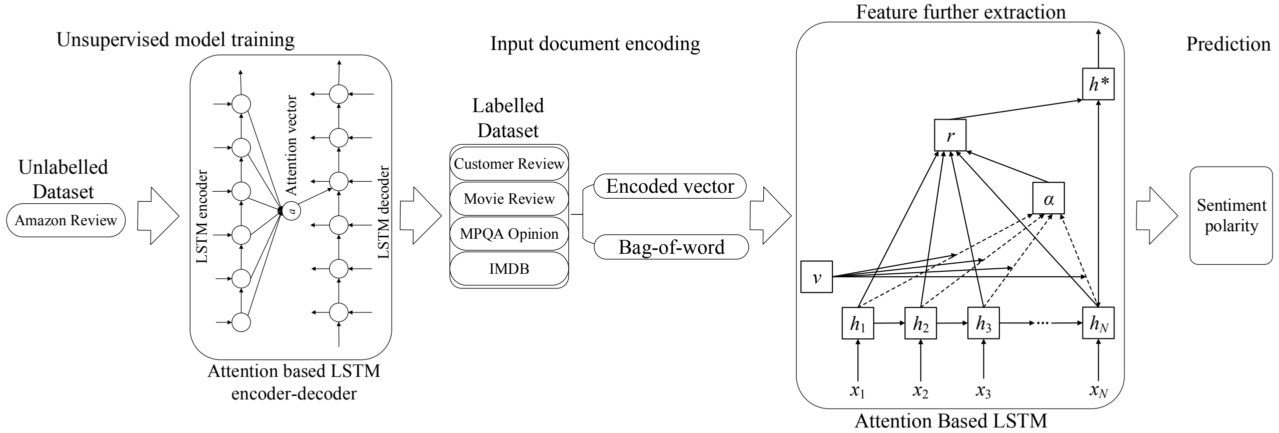


Fig. 1: Model Pipeline

the word embedding layer with the pre-trained word vectors and fine-tuned it in the training process.

For easier understanding, we depict a sequence to sequence model to a single LSTM encoder-decoder without attention mechanism in Fig.2. For the encoding part, the input document is end with an $\langle EOS\rangle$ symbol which means the end. The words in the documents are input into the encoder one by one. And for the decoding part, the hidden state that produced by the encoding part is utilized as the initial state of the LSTM decoder. At each time step, the output is applied as the input for next time step until the model outputs an $\langle EOS\rangle$ symbol. And in this step, we add the attention mechanism to the unsupervised learning model. Fig. 3 is the attention based LSTM encoder-decoder to capture the significant information. In practice, at each time step, there is one attention vector. And In Fig. 3, we only depict the attention vector in time step t_4 for example. As is shown in the figure, we add a context vector a_t at time step t to the output layer of the LSTM decoder. The vector a_t is computed as follows:

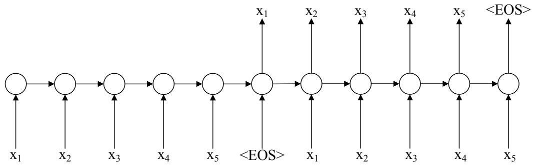


Fig. 2: Sequence to sequence model

$$e_{tj} = sim(s_{t-1}, h_j) \tag{2}$$

$$\alpha_{tj} = \frac{\exp(e_{tj})}{\sum_{i=1}^{T_x} \exp(e_{ti})} \tag{3}$$

$$a_t = \sum_{j=1}^{T_x} \alpha_{tj} h_j \tag{4}$$

where e_{tj} is the cosine similarity between the state of the memory cell in time step t_1 in the decoder s_{t-1} and the state of the memory cell in time step j of the encoder h_j. α_{tj} is the attention weight of h_j.

The values of the gates and cells in the LSTM decoder with attention vectors added are computed by:

$$i = \sigma(U^i x_t + W^i s_{t-1} + a_t) \tag{5}$$

$$f = \sigma(U^f x_t + W^f s_{t-1} + a_t) \tag{6}$$

$$o = \sigma(U^o x_t + W^o s_{t-1} + a_t) \tag{7}$$

$$\hat{c}_t = \tanh(U^c x_t + W^c s_{t-1} + a_t) \tag{8}$$

$$c_t = c_{t-1} \circ f + \hat{c}_t \circ i \tag{9}$$

$$s_t = \tanh(c_t) \circ o \tag{10}$$

where x_t is the input to the memory cell. W^i, W^f, W^o, W^c, U^i, U^f, U^o, U^c are weight matrices. i, f, o stand for values of the input gate, forget gate, and output gate. $\hat{c}_t$ is the candidate value for states of the memory cell. c_t is the new state of memory cell and s_t is the output of hidden state at time step t. And a_t is added to i, f, o, c_t respectively.

We train the unsupervised model with a relatively large unlabelled corpus Amazon Reviews[2]. The encoded vector produced from the encoding step carries the important information of the input. Thus we use it to encode the input document into a low dimensional vector.

2) Input document encoding: After finishing training the attention based LSTM encoder-decoder, we encode the document vectors. We input the training set and test set into the LSTM encoder model respectively. In this way, we can get the encoded hidden representation h_t of the documents:

$$i = \sigma(U^i x_t + W^i s_{t-1}) \tag{11}$$

$$f = \sigma(U^f x_t + W^f s_{t-1}) \tag{12}$$

$$o = \sigma(U^o x_t + W^o s_{t-1}) \tag{13}$$

$$\hat{v}_t = \tanh(U^c x_t + W^c s_{t-1}) \tag{14}$$

$$v_t = v_{t-1} \circ f + \hat{v}_t \circ i \tag{15}$$

$$s_t = \tanh(v_t) \circ o \tag{16}$$

where x_t is the input to the memory cell. W^i, W^f, W^o, W^c, U^i, U^f, U^o, U^c are weight matrices. i, f, o are the gates. $\hat{v}_t$ and v_t are the hidden state of LSTM. And s_t is

[2] http://snap.stanford.edu/data/sentiment/web-Amazon.html

the output of hidden state at time step t. In this step, the matrix v represents the document vectors and we input it to the supervised model in next step.

B. Supervised learning

1) Feature further extraction: In the second part, a supervised LSTM network is employed to further extract the features of the input instances in a deeper level. In this step, the encoded document vectors and the Bag-of-words features of the instances are input into the model simultaneously. We refer to Wang et al.'s work to build an Attention-based LSTM structure in this step [7]. Different from using the pre-defined target as the vector in the network, we use the document vectors extracted from the unsupervised learning step to compute the attention weights. The attention mechanism in the structure can help to obtain the important information in the input documents. Thus the significant parts for computing the global sentiment can be captured.

First, we employ the standard LSTM network which is the same as the encoder in Sect. III-A2:

$$i = \sigma(U^i x_t + W^i s_{t-1}) \tag{17}$$

$$f = \sigma(U^f x_t + W^f s_{t-1}) \tag{18}$$

$$o = \sigma(U^o x_t + W^o s_{t-1}) \tag{19}$$

$$\hat{h}_t = \tanh(U^c x_t + W^c s_{t-1}) \tag{20}$$

$$h_t = h_{t-1} \circ f + \hat{h}_t \circ i \tag{21}$$

$$s_t = \tanh(h_t) \circ o \tag{22}$$

Different from the equations in Sect. III-A2, we use $\hat{h}_t$ and h_t to represent the hidden state of LSTM here.

Then we use the v got from Eq. 15 to represent the document vectors encoded in the last step. Now we compute the attention weight α as:

$$e_t = sim(v, h_t) \tag{23}$$

$$\alpha_t = \frac{\exp(e_t)}{\sum_{i=1}^{T_x} \exp(e_i)} \tag{24}$$

where h_t is the LSTM hidden state got from Eq. 21, T_x is the current input length at time step t, e_t is the cosine similarity of document vectors and hidden state and α_t is the attention weight of time step t.

Then we use H to represent the hidden vectors $[h_1, ..., h_N]$ of the LSTM network. Where N is the length of the input documents. And we used A as a matrix of $[\alpha_1, ..., \alpha_N]$ which has N attention vectors in total. The further deep level representation can be computed as:

$$r = HA^T \tag{25}$$

With r and h, we compute the final whole document representation h^* as follow:

$$h^* = \tanh(U^r r + U^h h_N) \tag{26}$$

where U^r and U^h are weight matrices.

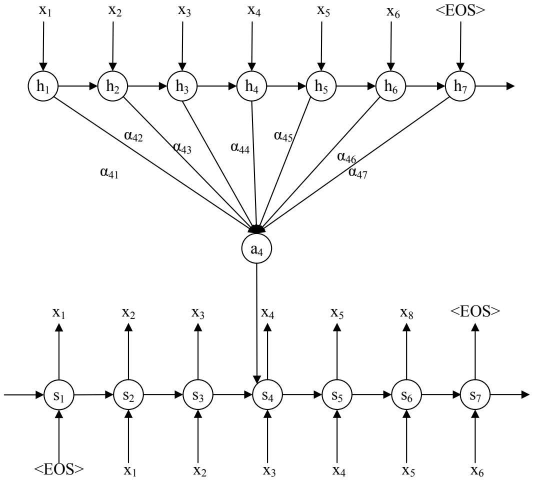


Fig. 3: Attention mechanism based LSTM encoder-decoder

2) Prediction: Finally, a Softmax classifier is attached to the LSTM network to predict whether the output of the LSTM network represents positive or negative. The input of the Softmax layer is the deep level feature extracted in last step, and the output of the Softmax layer is computed as:

$$\hat{y}_i = Softmax(U^p h^* + b_p) \tag{27}$$

where U^p and b_p are the weight matrix and bias vector in the Softmax layer.

We split the dateset into training set and test set, the specific splitting strategy will be introduced in Sect. IV. And as our task is a binary classification task, the Cross Entropy is used as loss function:

$$\text{Cos}t = [y_i \log \hat{y}_i + (1 - y_i) \log(1 - \hat{y}_i)] \tag{28}$$

where $\hat{y}_i$ is obtained from Eq. 27 and y_i is the label of the instance i. In this research, we use the pre-trained GloVe word embedding to initialise our model [14]. And we employ Adam, an algorithm based on adaptive estimates of lower-order moments, to optimise the back propagation through time algorithm [25]. The update of the parameter θ_t in Adam is computed as follow:

$$g_t = \nabla_\theta f(\theta) \tag{29}$$

$$m_t = \mu * m_{t-1} + (1 - \mu) * g_t \tag{30}$$

$$n_t = \nu * n_{t-1} + (1 - \nu * g_t^2) \tag{31}$$

$$\hat{m}_t = \frac{m_t}{1 - \mu^t} \tag{32}$$

$$\hat{n}_t = \frac{n_t}{1 - \nu^t} \tag{33}$$

$$\Delta\theta = -\frac{\hat{m}_t}{\sqrt{\hat{n}_t} + \epsilon} * \eta \tag{34}$$

where $f(\theta)$ is the objective function, g_t is the gradient of θ at time step t, m_t is called the biased first moment estimate and n_t is called the biased second raw moment estimate. And $\hat{m}_t$, $\hat{n}_t$ are bias-corrected first moment estimate and second raw moment estimate respectively. The hyper-parameters η

is step size, $\mu, \nu \in [0, 1)$ are exponential decay rates for the moment estimates. Finally, the ϵ is a constant set for ensuring the denominator bigger than 0.

Our algorithm is implemented on the Theano platform which contains many useful functions for deep learning. And in order to accelerate our training process, we refer Bengio et al.'s approach to train the i, f, o, a_t together [26].

IV. Experiment study

A. Datasets

In the unsupervised learning step, we utilise the unlabelled Amazon Reviews dataset. To guarantee the validity of the experiments, we filter the small part of the corpus that appear in the Customer Review dataset. The length of the instances in this dataset are between 5 and 100 words. And there are 607,551 instances in total.

And in our work, we use four public datasets to test the performance of our proposed model and compare our framework with the baseline methods. In the aim of training in a short time, three small datasets are introduced at first: The non-balanced dataset Customer Review, MPQA Opinion Corpus and the balanced dataset Movie Review. After achieving pretty good results in these three datasets, we further employ a larger balanced dataset, IMDB. The details of the four datasets are listed in Table I.

TABLE I: Dataset Configuration

Dataset	Instances number	Positive/Negative
Customer Review	3,772	0.64/0.36
MPQA Opinion	10,624	0.31/0.69
Movie Review	10,662	0.5/0.5
IMDB	50,000	0.5/0.5

As to the specific splitting strategy of the dataset, for the first three datasets, we adopt the common method in Nakagawa et al.'s work [27]. we randomly split the datasets into ten sets and adopt the 10-fold cross validation strategy to compute the average accuracy. As for IMDB, the dataset is originally split into 50%/50% for training and testing. And we split 10% of the training set as the validation set.

B. Evaluation Metrics and Baselines

In the aim of evaluating the proposed model's ability, we utilise the common method to measure the model's performance:

$$\text{Accuracy} = \frac{\sum_{i=1}^{\text{n}} 1\{y_i = \hat{y}_i\}}{n} \tag{35}$$

where y_i stands for the true value that the instance is labelled, $\hat{y}_i$ is the result predicted by our model, and n is the number of testing instances. Then we assess the generalization ability of our model by predicting the sentiment polarity of the instances in the testing set.

To test the performance of the proposed model, several baseline methods in the literature are adopted. As Nakagawa et al. has shown to us, for the Customer Review, MPQA Opinion Corpus and Movie Review, the following four approaches achieve better accuracy [27]:

1) Bag-of-words: A logistic regression model trained by Bag-of-words features is used to classify the polarity of the instances.
2) Vote by lexicon: In this method, a pre-defined sentiment lexicon is employed to score each word in the statement. Then the polarity of the statement is decided by the overall sentiment score.
3) Rule-based reversal: The polarity of a subjective sentence is deterministically decided based on rules, by considering the sentiment polarities of dependency subtrees.
4) Tree-Based CRF: This is a dependency tree based method using conditional random fields with hidden variables.

And for the IMDB dataset, as Maas et al.'s study has demonstrated, the following four techniques could get preferable results in this dataset [28].

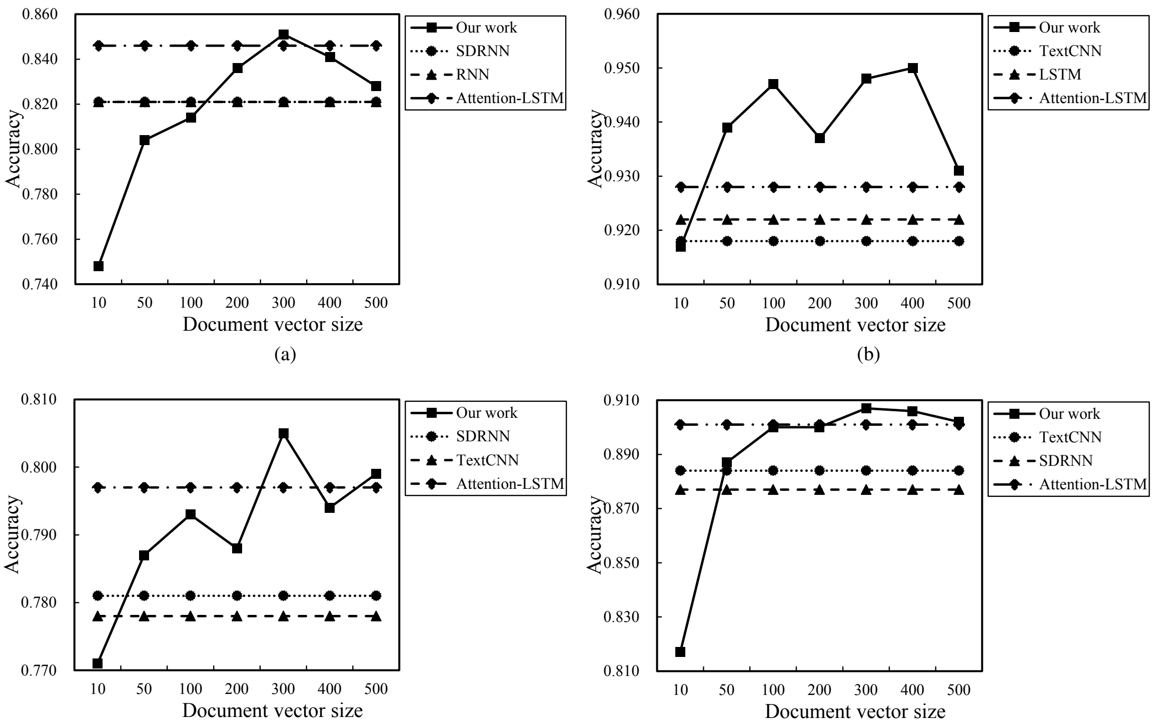
1) LSA: In this method, the document from sparse high-dimensional space is mapped to a low-dimensional vector space called Latent Semantic Space.
2) LDA: Latent Dirichlet Allocation is an unsupervised machine learning method to generate the topics of the statements.
3) MAAS Semantic: This is a probabilistic model optimized by the semantic similarity for sentiment analysis.
4) MAAS Full: It is similar to the MAAS Semantic method. The only difference is that MAAS Full is optimized by both the word sentiment and semantic similarity.

Also, we compare the performance of our proposed model with other popular methods for the sentiment analysis tasks.

1) TextCNN: A 2-Dimension convolutional neural network with word embedding layer.
2) RNN: A standard recurrent neural network with the same word embedding used in our proposed framework.
3) SDRNN: In this method, a dual recurrent neural network which kept the former output of the RNN.
4) LSTM: A standard Long short term memory with the same word embedding used in our work.
5) Attention-LSTM: We employed an attention aware semi-supervised LSTM for sentiment analysis in our former work. The parameters in the unsupervised learning step can be used to initialise the supervised model, which improve the performance [29].

C. Results and discussion

In our experiments, to evaluate the best performance of our framework, we try different document vectors size in the framework. Fig.4a, Fig.4b, Fig.4c, Fig.4d depict the performance variation with the document vectors size. We also depict the best three baseline methods for comparison. As we can see in the figures, the accuracy increases with the document size getting larger. And after getting the best result, the accuracy would decrease. On the one hand, if the



(a)

(b)

(c)

(d)

Fig. 4: Accuracy variation over document vectors size in four datasets. (a), (b), (c), (d) illustrate performance variation over different document vectors size on Customer Review, MPQA, Movie Review and IMDB respectively.

document vectors size is too small, it cannot carry enough important information needed. On the other hand, if the vector is too large, it would lead to over-fitting problem, as a result, the performance of the model decreased.

We list the test results of our proposed framework against the the baseline models in Table II and Table III. As to the different chosen baseline methods, the results for Customer Review, MPQA and Movie Review are shown in Table II and the results for IMDB were shown in Table III. In the four datasets, we can see our proposed framework's performance is better than the other baseline models.

Different from those traditional models, our proposed model is based on low-dimensional word embedding and deep neural networks. For one thing, the input features include more useful syntactic and semantic information of the words instead of a single number. For another, our framework has the ability to learn the long distance relations between words since it had deeper structure.

With regard to the TextCNN method, it is not based on the time series. As a result, it is hard to learn the underlying relationships between the words in the long input. And about the normal RNN and SDRNN model, they both have the problem of vanishing gradient and may lose the information between words in a long distance. The normal LSTM can weaken the vanishing gradient problem, however it cannot make use of the information in the unlabelled dataset.

Compared with these deep learning methods, our proposed model has more superiorities. In the unsupervised learning step, it utilises the unlabelled dataset to train an unsupervised learning model which could reserve the useful information in the large corpus at the least expense.

And as for the Attention-LSTM method, in the supervised learning step, since there is no mechanism to help the model to catch the vital parts of the input documents, our proposed method also outperforms it.

From the results listed in the tables, we can see that our proposed framework performs better than all of the baseline models. The reason is mainly three fold: 1) The LSTM network behaves well in learning long-range dependencies of words as it can catch the sequence information of the instance and has conquered the problem of gradient vanishing; 2) The pre-trained unsupervised encoder-decoder based on attention mechanism helps to exploit the external knowledge, which helps to enhance the model's ability to classify the sentiment polarity; 3) The attention mechanism in the supervised learning step improves the behaviour of the LSTM since it helps to capture the important information in the input documents. In conclusion, the proposed semi-supervised learning model can promote the performance of the deep LSTM network for sentiment polarity classification tasks.

In Fig. 5, we also visualise the word embedding that our framework learned. We use the widely used dimension reduction method t-SNE [30] to project the word embedding into a two-dimensional space. In the figure, each word embedding

TABLE II: Accuracy for Customer Review, MPQA Opinion and Movie Review

Method	Customer Review	MPQA Opinion	Movie Review
Bag-of-words	0.814	0.841	0.764
Voting by lexicon	0.742	0.817	0.631
Rule-Based reversal	0.743	0.818	0.629
Tree-CRF	0.814	0.861	0.773
TextCNN	0.819	0.918	0.778
RNN	0.821	0.867	0.781
SDRNN	0.821	0.867	0.781
LSTM	0.764	0.922	0.774
Attention-LSTM	0.846	0.928	0.797
Framework in this paper	**0.851**	**0.950**	**0.805**

TABLE III: Accuracy for IMDB

Method	IMDB
LSA	0.839
LDA	0.674
MAAS Semantic	0.873
MAAS full	0.874
TextCNN	0.884
RNN	0.829
SDRNN	0.877
LSTM	0.835
Attention-LSTM	0.901
Framework in this paper	**0.907**

is projected into a single point. And we can see that the word embeddings are roughly separated into two parts. The reason is that our task is classifying the documents into two classes. The upper part mainly includes the positive words such as “best”, “fascinating” and “wonderful”. The lower left part contains the negative words like “awful”, “terrible” and “boring”. As depicted in Fig. 5, words with same sentiment polarities are clustered, which means that our framework has the ability to take use of the word embedding to classify the sentiment polarity of the documents.

V. Conclusion and Future Work

In this work, for the tasks of sentiment analysis, we first analysed the common used methods for representing and learning the features in the documents. Moreover, we introduced widely used models for sentiment analysis. In detail, we introduced the advantage of deep neural networks such as RNN and LSTM, then we explained the advantage of making use of unlabelled data. We also interpreted the importance of focusing on the import part of the input document.

Afterwards, we proposed a LSTM based semi-supervised attention framework. In this work, we designed an unsupervised encoder-decoder model which exploits the information in large unlabelled corpus. Then we encoded the input documents into vectors using the unsupervised model. Finally we employed an attention based LSTM network for further feature extraction with a Softmax layer for prediction. At this step, in order to compute the attention vector, the encoded document and the Bag-of-words features of the input documents were input into the LSTM network simultaneously.

In addition, since the document vectors played a crucial role in both steps of our framework, we adjusted the dimension of the encoded document vectors to get the best result. In the results, we can see that this parameter can influence the model’s performance greatly. From the comparison between our proposed semi-supervised framework with other baseline methods, it is found the proposed framework beats all of other baseline methods. The experimental study reveals the superiority of unsupervised learning and the promising performance of the attention mechanism. Last but not the least, the results show the generalization ability of the LSTM based semi-supervised attention architecture for the sentiment analysis tasks.

Concerning the future work, on the one hand, we plan to use other methods to get the document vectors, for example, we could apply the gated recurrent unit network (GRU) for the unsupervised learning step. The structure of it is simpler and could effectively avoid the over-fitting problem [31]. On the other hand, we project to design more powerful models in the supervised learning step to capture the important parts. For instance, we can input the sentiment score of the certain word together with the Bag-of-words feature into the model.

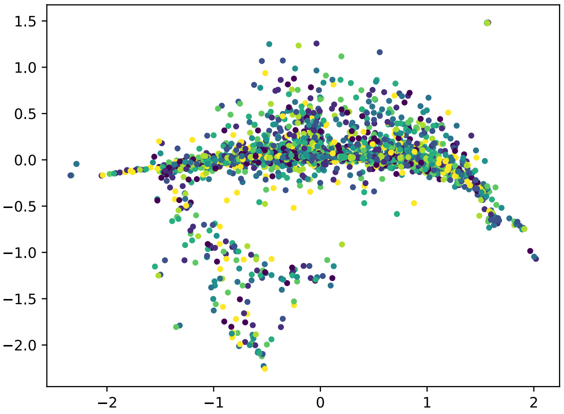


Fig. 5: Embeddings learned by our proposed framework

Acknowledgement

This work was partially supported by the State Key Laboratory of Software Development Environment of China (No. SKLSDE-2019ZX-16).

References

[1] B. Pang and L. Lee, "Opinion mining and sentiment analysis," *Foundations and Trends in Information Retrieval*, vol. 2, no. 1-2, pp. 1–135, 2008.

[2] L. Duan, Y. Xu, S. Cui, J. Chen, and M. Bao, "Feature extraction of motor imagery eeg based on extreme learning machine auto-encoder," in *Proceedings of 2015 International Conference on Extreme Learning Machines*, 2015, pp. 361–370.

[3] G. Keren and B. Schuller, "Convolutional rnn: an enhanced model for extracting features from sequential data," *arXiv preprint arXiv:1602.05875*, 2016.

[4] F. Hu, L. Li, Z. Zhang, J. Wang, and X. Xu, "Emphasizing essential words for sentiment classification based on recurrent neural networks," *Journal of Computer Science and Technology*, vol. 32, no. 4, pp. 785–795, 2017.

[5] R. Jozefowicz, W. Zaremba, and I. Sutskever, "An empirical exploration of recurrent network architectures," in *Proceedings of the 32nd International Conference on Machine Learning*, 2015, pp. 2342–2350.

[6] A. Graves, "Supervised sequence labelling with recurrent neural networks," *Studies in Computational Intelligence*, vol. 385, 2008.

[7] Y. Wang, M. Huang, X. Zhu, and L. Zhao, "Attention-based LSTM for aspect-level sentiment classification," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 2016, pp. 606–615.

[8] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[9] C. Doersch, A. Gupta, and A. A. Efros, "Unsupervised visual representation learning by context prediction," in *Proceedings of 2015 IEEE International Conference on Computer Vision*, 2015, pp. 1422–1430.

[10] A. Nikfarjam, A. Sarker, K. OConnor, R. Ginn, and G. Gonzalez, "Pharmacovigilance from social media: mining adverse drug reaction mentions using sequence labeling with word embedding cluster features," *Journal of the American Medical Informatics Association*, vol. 22, no. 3, pp. 671–681, 2015.

[11] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," *Advances in Neural Information Processing Systems*, vol. 4, pp. 3104–3112, 2014.

[12] Y. Zhuang, H. Wang, J. Xiao, F. Wu, Y. Yang, W. Lu, and Z. Zhang, "Bag-of-discriminative-words (bodw) representation via topic modeling," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 5, pp. 977–990, 2017.

[13] Y. Hou, C. Tan, X. Wang, Y. Zhang, J. Xu, and Q. Chen, "Hitszicrc: Exploiting classification approach for answer selection in community question answering," in *Proceedings of the 9th International Workshop on Semantic Evaluation*, 2015, pp. 196–202.

[14] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proceedings of 2014 Conference on Empirical Methods in Natural Language Processing*, 2014, pp. 1532–1543.

[15] W. Rong, B. Peng, Y. Ouyang, C. Li, and Z. Xiong, "Structural information aware deep semi-supervised recurrent neural network for sentiment analysis," *Frontiers of Computer Science*, vol. 9, no. 2, pp. 171–184, 2015.

[16] A. E. Mousa and B. W. Schuller, "Contextual bidirectional long short-term memory recurrent neural network language models: A generative approach to sentiment analysis," in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, 2017, pp. 1023–1032.

[17] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhutdinov, R. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention," in *Proceedings of the 32nd International Conference on Machine Learning*, 2015, pp. 2048–2057.

[18] S. Majerus, F. Peters, M. Bouffier, N. Cowan, and C. Phillips, "The dorsal attention network reflects both encoding load and top-down control during working memory," *Journal of Cognitive Neuroscience*, vol. 30, no. 2, 2018.

[19] N. Mukhtar and M. A. Khan, "Urdu sentiment analysis using supervised machine learning approach," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 32, no. 2, pp. 1–15, 2018.

[20] A. G. Pablos, M. Cuadros, and G. Rigau, "W2VLDA: almost unsupervised system for aspect based sentiment analysis," *Expert Systems with Applications*, vol. 91, pp. 127–137, 2018.

[21] M.-R. Amini and N. Usunier, "Semi-supervised learning," *Synthesis Lectures on Artificial Intelligence & Machine Learning*, vol. 172, no. 2, pp. 1826–1831, 2006.

[22] K. Cho, B. V. Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.

[23] J. Chorowski, D. Bahdanau, K. Cho, and Y. Bengio, "End-to-end continuous speech recognition using attention-based recurrent nn: First results," *Eprint Arxiv*, 2014.

[24] Y. Shi, K. Yao, H. Chen, Y. Pan, M. Hwang, and B. Peng, "Contextual spoken language understanding using recurrent neural networks," in *Proceedings of 2015 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2015, pp. 5271–5275.

[25] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[26] F. Bastien, P. Lamblin, R. Pascanu, J. Bergstra, I. Goodfellow, A. Bergeron, N. Bouchard, D. Warde-Farley, and Y. Bengio, "Theano: new features and speed improvements," *arXiv preprint arXiv:1211.5590*, 2012.

[27] T. Nakagawa, K. Inui, and S. Kurohashi, "Dependency tree-based sentiment classification using CRFs with hidden variables," in *Proceedings of 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, 2010, pp. 786–794.

[28] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, "Learning word vectors for sentiment analysis," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, 2011, pp. 142–150.

[29] J. Liu, W. Rong, C. Tian, M. Gao, and Z. Xiong, "Attention aware semi-supervised framework for sentiment analysis," in *Proceedings of 26th International Conference on Artifical Artificial Neural Networks*, 2017, pp. 208–215.

[30] L. V. D. Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research*, vol. 9, no. 2605, pp. 2579–2605, 2017.

[31] J. Chung, C. Gulcehre, K. H. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.