Improved Memory Network for Aspect Sentiment Analysis

Hu HAN\(^*\), Xu-xu BAI and Jin LIU

School of Electronic and Information Engineering, Lanzhou Jiaotong University, Lanzhou 730070, China

\(^*\)Corresponding author

Keywords: Sentiment analysis, Deep memory network, Convolutional neural networks, Bidirectional Long short term memory

Abstract. In order to explore impact of different memory modules under the framework of deep memory network for aspect level sentiment classification, three kinds of different memory modules are designed. One of them uses CNN to build a memory module, which is capable of capturing local information from original sentence. The other one of them uses BiLSTM to build another memory module, which is capable of capturing sequence information. And the last one of them uses CNN and BiLSTM, which combines both local and sequence information together, to build memory module at the same time. Experiments on laptop and restaurant datasets demonstrate that our three methods achieve better results than MemNet and feature based SVM approach.

Introduction

Aspect level sentiment classification is a fine-grained task in the field of sentiment analysis[1]. The sentiment polarity of a sentence is dependent on not only the content but also the aspect. For example, in sentence “USB3 Peripherals are noticeably less expensive than the ThunderBolt ones”, there are two aspects, “USB3 Peripherals” and “ThunderBolt”, the sentiment polarity of “ThunderBolt” is negative and “USB3 Peripherals” is positive. Sentence-level sentiment classification algorithm are not capable to capture such fine-grained sentiment on opinion targets.

Memory network is a general machine learning framework introduced by Weston[2]. Its central idea is inference with a long-term memory component, which could be read, written to, and jointly learned with the goal of using it for prediction. In 2016, Tang proposed to use memory network for aspect level sentiment classification[3], which capture importance of context words. It is verified that the proposed approach performs better than LSTM architectures. However, local information and sequence information from original sentence are not taken into account, and it only uses the sequence of word vectors as memory module.

In order to further explore impact of different memory modules for aspect level sentiment classification, we propose to build up three kinds of memory modules. One of them uses CNN to build a memory module, which is capable of capturing local information from original sentence. The other one of them uses BiLSTM to build another memory module, which is capable of capturing sequence information. And the last one uses CNN and BiLSTM, which combines both local and sequence information together, to build memory module. Experimental results show that our method outperforms most of the baseline methods and the state-of-the art approaches. The main contribution of this work are as follows:

a) CNN and BiLSTM are used to design three kinds of different memory modules for aspect level sentiment classification.

b) It is found that CNN and BiLSTM is capable of capturing local and sequence information from original sentence, and our approach gets better performance comparing with the model only using the input words vectors as memory module.
Related Work

Aspect Level Sentiment Classification

Aspect level sentiment classification is a fine grained classification task in sentiment analysis, which aims at identifying the sentiment polarity of a sentence expressed towards an aspect [4,5]. Most existing works use machine learning algorithms, and build sentiment classifier from sentences with manually annotated polarity labels. One of the most successful approaches in literature is feature based SVM[1]. Experts could design effective feature templates and make use of external resources like parser and sentiment lexicons. Nguyen and Shirai proposed a method which is an extension of RNN ( Recursive Neural Network) that takes both dependency and constituent trees of a sentence into account[6]. In recent years, neural network approaches are of growing attention for their capacity to learn powerful text representation from data. Attention mechanism and LSTM ( Long Short-Term Memory) are combined to build a model as ATAE-LSTM, which can concentrate on different parts of a sentence when different aspects are taken as input [7]. However, these neural network models are computationally expensive, and could not explicitly reveal the importance of context evidences with regard to an aspect.

Attention

Inspired by human visual attention, the attention mechanism is proposed by Bahdanau, Cho, and Bengio in machine translation, which is introduced into the Encoder-Decoder framework to select the reference words in source language for words in target language[8]. It is also used in image caption generation and natural language question answering[9,10]. As can be see that not all words or sentences contribute equally for these work. In order to capture the crucial components over different semantic levels for document classification, many research introduce attention mechanism to document classification and aspect level sentiment classification. [11] proposed attention based hierarchical neural network for document classification achieved promising performance. [12] proposed user product attention based(UPA-NSC) method improve performance of user-product sentiment classification.

Memory Networks

Recently, Weston introduced memory networks as an inference components combined with a long-term memory component, this memory can be read and written. Memory network architecture consists of an array of objects called memory m and four components I, G, O and R. where m is an array of objects such as an array of vectors. Among these four components, I converts input to internal feature representation, G updates old memories with new input, O generates an output representation given a new input and the current memory state, R outputs a response based on the output representation. Motivated by the success of memory network’s application in many NLP field, Tang introduce a deep memory network (MemNet) for aspect level sentiment classification[3], MemNet used the sequence of word vectors as memory module and adopted the multiple attentions as computational layers to read and write memory modules. Chen[13] used the Bidirectional LSTM (BiLSTM) to build the memory module and adopted the multiple attentions with a GRU network to read memory information.

Methodology

In this section, we describe our model in detail. First, the task definition and notation are given. Afterwards, we give an overview of our model. Lastly, we give four strategies to build memory cell from input sentence and give a brief introduce to how to evaluation aspect-level sentiment classification.
Task Definition and Notation

We represent the review as a sentence $S = \{w_1, w_2, \cdots, w_i, \cdots, w_n\}$ and an aspect word $w_i$. Aspect-level sentiment classification aims to inferring the numeric rating (1-5 or 1-10 stars) or sentiment polarity (positive, neutral, negative) of these reviews towards the aspect $w_i$ according to their text information. For example, the sentiment polarity of sentence “Best of all is the warm vibe, the owner is super friendly and service is fast” towards aspect “vibe” and “service” is both positive. When dealing with a text corpus, we use word2vec embed each word into a low dimension, continuous and real-valued semantic space $L \in \mathbb{R}^{d \times |V|}$, where $d$ is the dimension of word vectors and $|V|$ is the vocabulary size.

An Overview of the Approach

The basic structure of our method is the same as MemNet proposed by Tang[3], one difference between our model and MemNet is that we redesign four kinds of memory models by using CNN and BiLSTM. In order to better introduce our work, the basic structure of MemNet is described in Figure 1. Which consist of four modules: embedding layer, memory cell, multiple computational layers and Softmax layer. In every computational layer, we regard aspect vector as input to adaptive select important slice from memory module output through attention layer. The computational layer’s output and the linear transformation of aspect vector are summed and the result is considered as the input of next computational layer.

Taking an embedding layer output as external memory $M$ and an aspect vector $V$ as computational layer’s input, the attention model outputs a continuous vector $vec$. The output vector is computed as a weighted sum of each piece of memory in $M$, namely

$$vec = \sum_{i=1}^{k} \alpha_i m_i$$

where $\alpha_i$ measures the importance for each piece of memory $m_i$ for current sentence and $k$ is the memory size. For each piece of memory $m_i$, we compute its semantic relatedness with aspect, the score function is defined as:

$$h_i = \tanh(W_h [m_i ; v] + b)$$

where $W_h$ are weight matrices. Then we feed $h_i$ to softmax function to calculate the final importance score $\alpha_i$. 


dd
\[ \alpha_i = \frac{\exp(h_i)}{\sum_{j=1}^{k} \exp(h_j)} \]  

(3)

**Memory Cell**

We describe our memory module in this part. The basic idea of modifying memory cell is that using only the sequence of word vectors as memory can’t capture enough information from original input. However, CNN and LSTM can capture local and sequence information from input sentence, respectively. In this work, we study three strategies to encode the input sentence in the memory cell.

**Model 1: Cnn Memory**

In comment: “This dress is too expensive, but the quality not good”. Obviously “good” is a positive word, but when we added “not” before “good”, thus the sentiment polarity of “quality” is negative. So capturing local information from input sentence had huge impact to sentiment analysis accuracy. Due to CNN is capable to capture context information from sequence, we take word vectors as input to CNN layer to compose word context representation and learn local information from words as memory module. An illustration of Cnn Memory module is given in Figure 2.

Given an input word \( w_i \) as current word, we employ filters with window sizes \( h = 3 \):

\[
w_{\text{context}(i)} = f(W_{i-1}^j \cdot w_{i-1}^j + W_i^j \cdot w_i + W_{i+1}^j \cdot w_{i+1}^j + b^j)\]

(4)

\[ M = [w_{\text{context}(1)}, w_{\text{context}(2)}, \ldots, w_{\text{context}(n)}] \]

(5)

**Model 2: BiLstm Memory**

MemNet[3] simply used the sequence of word vectors as memory, which cannot synthesize phrase-like features in the original sentence. It is straightforward to achieve the goal with the models of RNN family. In this paper, we use Deep Bidirectional LSTM (DBLSTM) to build the memory which records all information to be read in the subsequent modules. At each time step \( t \), the forward LSTM not only outputs the hidden state \( \overrightarrow{h_i^t} \) at its layer \( t(h_i^0 = v_i) \) but also maintains a memory \( \overrightarrow{c_i^t} \) inside its hidden cell. The update process at time \( t \) is as follows:

\[
i = \sigma(W_i \overrightarrow{h_i^{t-1}} + U_i \overrightarrow{h_{i-1}}) \]

(6)

\[
f = \sigma(W_f \overrightarrow{h_i^{t-1}} + U_f \overrightarrow{h_{i-1}}) \]

(7)

\[
o = \sigma(W_o \overrightarrow{h_i^{t-1}} + U_o \overrightarrow{h_{i-1}}) \]

(8)

\[
g = \tanh(W_g \overrightarrow{h_i^{t-1}} + U_g \overrightarrow{h_{i-1}}) \]

(9)

\[
\overrightarrow{c_i^t} = f \cdot \overrightarrow{c_{i-1}} + i \cdot g \]

(10)

\[
\overrightarrow{h_i^t} = o \cdot \tanh(\overrightarrow{c_i^t}) \]

(11)
where $\sigma$ and tanh are sigmoid and hyperbolic tangent functions, $\vec{W}_i, \vec{W}_f, \vec{W}_o, \vec{W}_g \in \mathbb{R}^{d_i \times d_i}$, $\vec{U}_i, \vec{U}_f, \vec{U}_o, \vec{U}_g \in \mathbb{R}^{d_i \times d_i}$, and $d_i$ is the number of hidden cells at the layer $l$ of the forward LSTM. The gates $i, f, o \in \mathbb{R}^{d_i}$ simulate binary switches that control whether to update the information from the current input, whether to forget the information in the memory cells, and whether to reveal the information in memory cells to output, respectively. $h = [h^i_1, h^o_1, h^f_1, \ldots, h^o_L]$ is forward LSTM. The backward LSTM does the same thing, except that its input sequence is reversed $h^b_i$. In our framework, we use BiLSTM to build the memory $M = [h, h^b]$ and it generally performs well in aspect level sentiment classification, the illustration is given in Figure 3.

![Figure 3. BiLSTM Memory.](image)

![Figure 4. CNNBiLSTM Memory.](image)

**Model 3: CNNBiLSTM Memory**

We use CNN and BiLSTM to build the memory which records text local information and sequence information, the illustration is given in Figure 4. We model the semantic representation of a memory slice in two stages. Firstly, we embed each word into a low dimension semantic space, using CNN to learn word context vectors. The reason is that CNN is capable of capturing local semantics of n-grams of various granularities, which are proven powerful for sentiment classification. At CNN layer, we take word vectors as input to CNN layer to compose word context representation. Given an input word $w_i$ as current word, we employ filters with window sizes $h = 3$:

$$w_{\text{context}(i)} = f(W_{i-1}^j \cdot w_{i-1} + W_i^j \cdot w_i + W_{i+1}^j \cdot w_{i+1} + b^j_i)$$

(12)

In the second stage, a LSTM layer takes in word context vectors as input to learn memory representation.

**Model Training**

We apply our model to aspect level sentiment classification under supervised learning framework. Sentence representation $\hat{d}$ is extracted from the words in the Sentence, it is a high level representation of the document. The sentiment classifier is built from documents with gold standard sentiment labels.

We use softmax to build the classifier because its outputs can be interpreted as conditional probabilities. softmax is calculated as given in Equation 13, where $C$ is the category number (e.g. positive or negative):

$$p_c = \frac{\exp(d_c)}{\sum_{k=1}^C \exp(d_k)}$$

(13)
Where $p_c$ is the predicted probability of sentiment class C. In our model, cross-entropy error between gold sentiment distribution and our model's sentiment distribution is defined as loss function for optimization when training:

$$L = - \sum_{d \in D} \sum_{c \in V} p^g_d(c) \cdot \log(p_c(d))$$

(14)

**Experimental Results**

In this section we evaluate our approach. Firstly we introduce the experimental setting and then report empirical results.

**Experimental Setting**

We evaluate the effectiveness of our approach on SemEval 2014[14], containing reviews of Laptop and Restaurant datasets. The statistics of the datasets are summarized in Table 1. We split each corpus into training and testing sets in the proportion of 8:2.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Negative</th>
<th>Neutral</th>
<th>Positive</th>
</tr>
</thead>
<tbody>
<tr>
<td>Laptop Reviews</td>
<td>858</td>
<td>454</td>
<td>980</td>
</tr>
<tr>
<td>Test</td>
<td>128</td>
<td>171</td>
<td>340</td>
</tr>
<tr>
<td>Restaurant Reviews</td>
<td>800</td>
<td>632</td>
<td>2159</td>
</tr>
<tr>
<td>Test</td>
<td>195</td>
<td>196</td>
<td>730</td>
</tr>
</tbody>
</table>

We use pre-trained the 300-dimensional word embeddings by GloVe[15] and use metrics Accuracy which measures the overall sentiment classification performance.

$$\text{Accuracy} = \frac{T}{N}$$

(15)

where $T$ is the numbers of predicted sentiment rating that are identical with gold sentiment ratings, $N$ is the numbers of documents.

**Experiment Results**

We compare our memory network with several baseline methods for Laptop and Restaurant datasets.

1. Feature-based SVM[1] It extracts some text features such as surface features, lexicon features and parsing features, and train a SVM for sentiment classification.

2. TD-LSTM-Att[8]: It uses two LSTM networks towards the aspect, and incorporate one attention on the outputs of forward and backward LSTMs.

3. MemNet[3]: It applies attention multiple times on the word embeddings, and the last attention’s output is fed into softmax for prediction. Obviously, MemNet is a simplistic form of our approach, it’s means memory cell is word embedding.

For each method, we use pre-trained the 300-dimensional GloVe[15] and tune the hyper parameters on the validation sets and use ADAM to update parameters when training. Experimental results are given in Table 2. It can be found that feature-based SVM outperforms other methods except CnnBiLstmMem on Restaurant datasets, which demonstrates the importance of a powerful feature representation for aspect level sentiment classification. Among four memory network models, CnnMem, BiLstmMem and CnnBiLstmMem all perform better than MemNet, which indicates that using CNN and BiLSTM is helpful to capture local and sequence information, respectively. Particularly BiLstmMem is a bit better than CnnMem, which explains that BiLstmMem not only capture sequence information but also preserves local information partially. And we can also find that the performance of CnnBiLstmMem is the best, which means that combining both local and sequence information is very effective.
Table 2. Experimental results.

<table>
<thead>
<tr>
<th>Model</th>
<th>Laptop</th>
<th>Restaurant</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feature-based SVM</td>
<td>70.49</td>
<td>80.16</td>
</tr>
<tr>
<td>TD-LSTM-Att</td>
<td>66.24</td>
<td>74.31</td>
</tr>
<tr>
<td>MemNet</td>
<td>70.33</td>
<td>78.16</td>
</tr>
<tr>
<td>CnnMem</td>
<td>70.89</td>
<td>78.73</td>
</tr>
<tr>
<td>BiLstmMem</td>
<td>71.35</td>
<td>79.62</td>
</tr>
<tr>
<td>CnnBiLstmMem</td>
<td>71.85</td>
<td>80.46</td>
</tr>
</tbody>
</table>

Effects of Attention Layers

One major setting that affects the performance of our model is the number of attention layers. We evaluate our models with 1 to 8 attention layers. The classification accuracy of model 1, model 2 and model 3 on the restaurant dataset are shown in Figure 5.

As can be seen that model 2 and model 3 achieve the best performance with 5 attention layers, and model 1 achieves the best performance with 7 attention layers, it is proved that using multiple attention layers might be sufficient to capture the sentiment features in complicated cases than using single attention layer. However, the performance is not monotonically increasing with the number of attention layers. For example, using 6 or 7 attention layers is worse than using 5 attention layers in model 2 and model 3, that is because the model will become more difficult to train and generalize with the complexity increasing.

Conclusion

In order to explore impact of different memory modules for aspect level sentiment classification, three kinds of different memory modules are designed. Compared with the model MemNet only using the input words vectors as memory module, our approach gets better performance. Especially, the last approach which uses CNN and BiLSTM to build up memory module at the same time achieves the best results. In the future, how to integrate location information about aspect and how to use memory modules into other neural networks are the focus of our research.

Acknowledgement

This work was supported by the National Social Science Foundation of China (No. 17BXW071)

References


