Aspect Level Sentiment Classification with Deep Memory Network

Duyu Tang, Bing Qin[*], Ting Liu
Harbin Institute of Technology, Harbin, China
{dytang, qinb, tliu}@ir.hit.edu.cn

Abstract

We introduce a deep memory network for aspect level sentiment classification. Unlike feature-based SVM and sequential neural models such as LSTM, this approach explicitly captures the importance of each context word when inferring the sentiment polarity of an aspect. Such importance degree and text representation are calculated with multiple computational layers, each of which is a neural attention model over an external memory. Experiments on laptop and restaurant datasets demonstrate that our approach performs comparable to state-of-art feature based SVM system, and substantially better than LSTM and attention-based LSTM architectures. On both datasets we show that multiple computational layers could improve the performance. Moreover, our approach is also fast. The deep memory network with 9 layers is 15 times faster than LSTM with a CPU implementation.

1 Introduction

Aspect level sentiment classification is a fundamental task in the field of sentiment analysis (Pang and Lee, 2008; Liu, 2012; Pontiki et al., 2014). Given a sentence and an aspect occurring in the sentence, this task aims at inferring the sentiment polarity (e.g. positive, negative, neutral) of the aspect. For example, in sentence "*great food but the service was dreadful!*", the sentiment polarity of aspect "*food*" is positive while the polarity of aspect "*service*" is negative. Researchers typically use machine learning algorithms and build sentiment classifier in a supervised manner. Representative approaches in literature include feature based Support Vector Machine (Kiritchenko et al., 2014; Wagner et al., 2014) and neural network models (Dong et al., 2014; Lakkaraju et al., 2014; Vo and Zhang, 2015; Nguyen and Shirai, 2015; Tang et al., 2015a). Neural models are of growing interest for their capacity to learn text representation from data without careful engineering of features, and to capture semantic relations between aspect and context words in a more scalable way than feature based SVM.

Despite these advantages, conventional neural models like long short-term memory (LSTM) (Tang et al., 2015a) capture context information in an implicit way, and are incapable of explicitly exhibiting important context clues of an aspect. We believe that only some subset of context words are needed to infer the sentiment towards an aspect. For example, in sentence "*great food but the service was dreadful!*", "*dreadful*" is an important clue for the aspect "*service*" but "*great*" is not needed. Standard LSTM works in a sequential way and manipulates each context word with the same operation, so that it cannot explicitly reveal the importance of each context word. A desirable solution should be capable of explicitly capturing the importance of context words and using that information to build up features for the sentence after given an aspect word. Furthermore, a human asked to do this task will selectively focus on parts of the contexts, and acquire information where it is needed to build up an internal representation towards an aspect in his/her mind.

[*] Corresponding author.

In pursuit of this goal, we develop deep memory network for aspect level sentiment classification, which is inspired by the recent success of computational models with attention mechanism and explicit memory (Graves et al., 2014; Bahdanau et al., 2015; Sukhbaatar et al., 2015). Our approach is data-driven, computationally efficient and does not rely on syntactic parser or sentiment lexicon. The approach consists of multiple computational layers with shared parameters. Each layer is a content- and location- based attention model, which first learns the importance/weight of each context word and then utilizes this information to calculate continuous text representation. The text representation in the last layer is regarded as the feature for sentiment classification. As every component is differentiable, the entire model could be efficiently trained end-to-end with gradient descent, where the loss function is the cross-entropy error of sentiment classification.

We apply the proposed approach to laptop and restaurant datasets from SemEval 2014 (Pontiki et al., 2014). Experimental results show that our approach performs comparable to a top system using feature-based SVM (Kiritchenko et al., 2014). On both datasets, our approach outperforms both LSTM and attention-based LSTM models (Tang et al., 2015a) in terms of classification accuracy and running speed. Lastly, we show that using multiple computational layers over external memory could achieve improved performance.

2 Background: Memory Network

Our approach is inspired by the recent success of memory network in question answering (Weston et al., 2014; Sukhbaatar et al., 2015). We describe the background on memory network in this part.

Memory network is a general machine learning framework introduced by Weston et al. (2014). 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. Formally, a memory network consists of a 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.

Let us take question answering as an example to explain the work flow of memory network. Given a list of sentences and a question, the task aims to find evidences from these sentences and generate an answer, e.g. a word. During inference, I component reads one sentence s_i at a time and encodes it into a vector representation. Then G component updates a piece of memory m_i based on current sentence representation. After all sentences are processed, we get a memory matrix m which stores the semantics of these sentences, each row representing a sentence. Given a question q, memory network encodes it into vector representation e_q, and then O component uses e_q to select question related evidences from memory m and generates an output vector o. Finally, R component takes o as the input and outputs the final response. It is worth noting that O component could consist of one or more computational layers (hops). The intuition of utilizing multiple hops is that more abstractive evidences could be found based on previously extracted evidences. Sukhbaatar et al. (2015) demonstrate that multiple hops could uncover more abstractive evidences than single hop, and could yield improved results on question answering and language modeling.

3 Deep Memory Network for Aspect Level Sentiment Classification

In this section, we describe the deep memory network approach for aspect level sentiment classification. We first give the task definition. Afterwards, we describe an overview of the approach before presenting the content- and location- based attention models in each computational layer. Lastly, we describe the use of this approach for aspect level sentiment classification.

3.1 Task Definition and Notation

Given a sentence $s = \{w_1, w_2, ..., w_i, ...w_n\}$ consisting of n words and an aspect word w_i [1] occurring in sentence s, aspect level sentiment classification aims at determining the sentiment polarity of

[1]In practice, an aspect might be a multi word expression such as "*battery life*". For simplicity we still consider aspect as a single word in this definition.

sentence s towards the aspect w_i. For example, the sentiment polarity of sentence "*great food but the service was dreadful!*" towards aspect "*food*" is positive, while the polarity towards aspect "*service*" is negative. When dealing with a text corpus, we map each word into a low dimensional, continuous and real-valued vector, also known as word embedding (Mikolov et al., 2013; Pennington et al., 2014). All the word vectors are stacked in a word embedding matrix $L \in \mathbb{R}^{d \times |V|}$, where d is the dimension of word vector and $|V|$ is vocabulary size. The word embedding of w_i is notated as $e_i \in \mathbb{R}^{d \times 1}$, which is a column in the embedding matrix L.

3.2 An Overview of the Approach

We present an overview of the deep memory network for aspect level sentiment classification.

Given a sentence $s = \{w_1, w_2, ..., w_i, ...w_n\}$ and the aspect word w_i, we map each word into its embedding vector. These word vectors are separated into two parts, aspect representation and context representation. If aspect is a single word like "*food*" or "*service*", aspect representation is the embedding of aspect word. For the case where aspect is multi word expression like "*battery life*", aspect representation is an average of its constituting word vectors (Sun et al., 2015). To simplify the interpretation, we consider aspect as a single word w_i. Context word vectors $\{e_1, e_2 \text{ ... } e_{i-1}, e_{i+1} \text{ ... } e_n\}$ are stacked and regarded as the external memory $m \in \mathbb{R}^{d \times (n-1)}$, where n is the sentence length.

An illustration of our approach is given in Figure 1, which is inspired by the use of memory network in question answering (Sukhbaatar et al., 2015). Our approach consists of multiple computational layers (hops), each of which contains an attention layer and a linear layer. In the first computational layer (hop 1), we regard aspect vector as the input to adaptively select important evidences from memory m through attention layer. The output of attention layer and the linear transformation of aspect vector[2] are summed and the result is considered as the input of next layer (hop 2). In a similar way, we stack multiple hops and run these steps multiple times, so that more abstractive evidences could be selected from the external memory m. The output vector in last hop is considered as the representation of sentence with regard to the aspect, and is further used as the feature for aspect level sentiment classification.

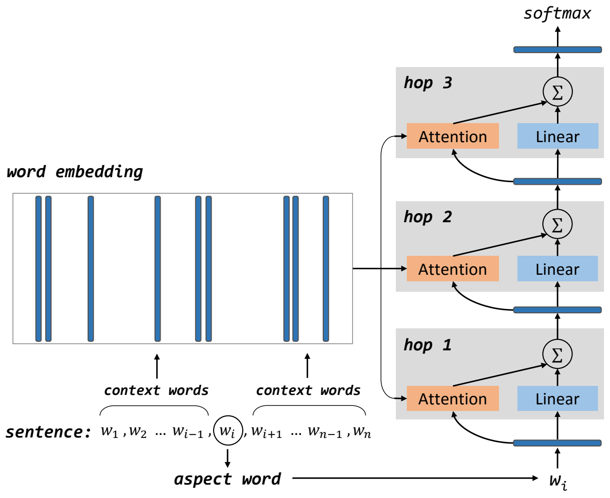

Figure 1: An illustration of our deep memory network with three computational layers (hops) for aspect level sentiment classification.

It is helpful to note that the parameters of attention and linear layers are shared in different hops. Therefore, the model with one layer and the model with nine layers have the same number of parameters.

3.3 Content Attention

We describe our attention model in this part. The basic idea of attention mechanism is that it assigns a weight/importance to each lower position when computing an upper level representation (Bahdanau et al., 2015). In this work, we use attention model to compute the representation of a sentence with regard to an aspect. The intuition is that context words do not contribute equally to the semantic meaning of a sentence. Furthermore, the importance of a word should be different if we focus on different aspect. Let us again take the example of "*great food but the service was dreadful!*". The context word "*great*" is more important than "*dreadful*" for aspect "*food*". On the contrary, "*dreadful*" is more important than "*great*" for aspect "*service*".

Taking an external memory $m \in \mathbb{R}^{d \times k}$ and an aspect vector $v_{aspect} \in \mathbb{R}^{d \times 1}$ as input, the attention model outputs a continuous vector $vec \in \mathbb{R}^{d \times 1}$. The output vector is computed as a weighted sum of each

[2]In preliminary experiments, we tried directly using aspect vector without a linear transformation, and found that adding a linear layer works slightly better.

piece of memory in m, namely

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

where k is the memory size, $\alpha_i \in [0, 1]$ is the weight of m_i and $\sum_i \alpha_i = 1$. We implement a neural network based attention model. For each piece of memory m_i, we use a feed forward neural network to compute its semantic relatedness with the aspect. The scoring function is calculated as follows, where $W_{att} \in \mathbb{R}^{1\times 2d}$ and $b_{att} \in \mathbb{R}^{1\times 1}$.

$$g_i = tanh(W_{att}[m_i; v_{aspect}] + b_{att}) \tag{2}$$

After obtaining $\{g_1, g_2, ... \ g_k\}$, we feed them to a $softmax$ function to calculate the final importance scores $\{\alpha_1, \alpha_2, ... \ \alpha_k\}$.

$$\alpha_i = \frac{exp(g_i)}{\sum_{j=1}^{k} exp(g_j)} \tag{3}$$

We believe that such an attention model has two advantages. One advantage is that this model could adaptively assign an importance score to each piece of memory m_i according to its semantic relatedness with the aspect. Another advantage is that this attention model is differentiable, so that it could be easily trained together with other components in an end-to-end fashion.

3.4 Location Attention

We have described our neural attention framework and a content-based model in previous subsection. However, the model mentioned above ignores the location information between context word and aspect. Such location information is helpful for an attention model because intuitively a context word closer to the aspect should be more important than a farther one. In this work, we define the location of a context word as its absolute distance with the aspect in the original sentence sequence[3]. On this basis, we study four strategies to encode the location information in the attention model. The details are described below.

[3]The location of a context word could also be measured by its distance to the aspect along a syntactic path. We leave this as a future work as we prefer to developing a purely data-driven approach without using external parsing results.

• Model 1. Following Sukhbaatar et al. (2015), we calculate the memory vector m_i with

$$m_i = e_i \odot v_i \tag{4}$$

where $\odot$ means element-wise multiplication and $v_i \in \mathbb{R}^{d\times 1}$ is a location vector for word w_i. Every element in v_i is calculated as follows,

$$v_i^k = (1 - l_i/n) - (k/d)(1 - 2 \times l_i/n) \tag{5}$$

where n is sentence length, k is the hop number and l_i is the location of w_i.

• Model 2. This is a simplified version of Model 1, using the same location vector v_i for w_i in different hops. Location vector v_i is calculated as follows.

$$v_i = 1 - l_i/n \tag{6}$$

• Model 3. We regard location vector v_i as a parameter and compute a piece of memory with vector addition, namely

$$m_i = e_i + v_i \tag{7}$$

All the position vectors are stacked in a position embedding matrix, which is jointly learned with gradient descent.

• Model 4. Location vectors are also regarded as parameters. Different from Model 3, location representations are regarded as neural gates to control how many percent of word semantics is written into the memory. We feed location vector v_i to a sigmoid function σ, and calculate m_i with element-wise multiplication:

$$m_i = e_i \odot \sigma(v_i) \tag{8}$$

3.5 The Need for Multiple Hops

It is widely accepted that computational models that are composed of multiple processing layers have the ability to learn representations of data with multiple levels of abstraction (LeCun et al., 2015). In this work, the attention layer in one layer is essentially a weighted average compositional function, which is not powerful enough to handle the sophisticated computationality like negation, intensification and contrary in language. Multiple computational layers allow the deep memory network to learn representations of text with multiple levels of abstraction. Each layer/hop retrieves important context words, and transforms the representation at previous level into a representation at a higher, slightly more abstract level. With the composition of enough such transformations, very complex functions of sentence representation towards an aspect can be learned.

3.6 Aspect Level Sentiment Classification

We regard the output vector in last hop as the feature, and feed it to a $softmax$ layer for aspect level sentiment classification. The model is trained in a supervised manner by minimizing the cross entropy error of sentiment classification, whose loss function is given below, where T means all training instances, C is the collection of sentiment categories, (s, a) means a sentence-aspect pair.

$$loss = -\sum_{(s,a)\in T}\sum_{c\in C} P_c^g(s,a)\cdot log(P_c(s,a)) \quad (9)$$

$P_c(s, a)$ is the probability of predicting (s, a) as category c produced by our system. $P_c^g(s, a)$ is 1 or 0, indicating whether the correct answer is c. We use back propagation to calculate the gradients of all the parameters, and update them with stochastic gradient descent. We clamp the word embeddings with 300-dimensional Glove vectors (Pennington et al., 2014), which is trained from web data and the vocabulary size is 1.9M[4]. We randomize other parameters with uniform distribution $U(-0.01, 0.01)$, and set the learning rate as 0.01.

4 Experiment

We describe experimental settings and report empirical results in this section.

4.1 Experimental Setting

We conduct experiments on two datasets from SemEval 2014 (Pontiki et al., 2014), one from laptop domain and another from restaurant domain. Statistics of the datasets are given in Table 1. It is worth noting that the original dataset contains the fourth category - conflict, which means that a sentence expresses both positive and negative opinion towards an aspect. We remove conflict category as the number of instances is very tiny, incorporating which will make the dataset extremely unbalanced. Evaluation metric is classification accuracy.

[4]Available at: http://nlp.stanford.edu/projects/glove/.

Dataset	Pos.	Neg.	Neu.
Laptop-Train	994	870	464
Laptop-Test	341	128	169
Restaurant-Train	2164	807	637
Restaurant-Test	728	196	196

Table 1: Statistics of the datasets.

4.2 Comparison to Other Methods

We compare with the following baseline methods on both datasets.

(1) **Majority** is a basic baseline method, which assigns the majority sentiment label in training set to each instance in the test set.

(2) **Feature-based SVM** performs state-of-the-art on aspect level sentiment classification. We compare with a top system using ngram features, parse features and lexicon features (Kiritchenko et al., 2014).

(3) We compare with three LSTM models (Tang et al., 2015a)). In **LSTM**, a LSTM based recurrent model is applied from the start to the end of a sentence, and the last hidden vector is used as the sentence representation. **TDLSTM** extends LSTM by taking into account of the aspect, and uses two LSTM networks, a forward one and a backward one, towards the aspect. **TDLSTM+ATT** extends TDLSTM by incorporating an attention mechanism (Bahdanau et al., 2015) over the hidden vectors. We use the same Glove word vectors for fair comparison.

(4) We also implement **ContextAVG**, a simplistic version of our approach. Context word vectors are averaged and the result is added to the aspect vector. The output is fed to a $softmax$ function.

Experimental results are given in Table 2. Our approach using only content attention is abbreviated to MemNet (k), where k is the number of hops. We can find that feature-based SVM is an extremely strong performer and substantially outperforms other baseline methods, which demonstrates the importance of a powerful feature representation for aspect level sentiment classification. Among three recurrent models, TDLSTM performs better than LSTM, which indicates that taking into account of the aspect information is helpful. This is reasonable as the sentiment polarity of a sentence towards different as-

	Laptop	Restaurant
Majority	53.45	65.00
Feature+SVM	**72.10**	**80.89**
LSTM	66.45	74.28
TDLSTM	68.13	75.63
TDLSTM+ATT	66.24	74.31
ContextAVG	61.22	71.33
MemNet (1)	67.66	76.10
MemNet (2)	71.14	78.61
MemNet (3)	71.74	79.06
MemNet (4)	72.21	79.87
MemNet (5)	71.89	80.14
MemNet (6)	72.21	80.05
MemNet (7)	**72.37**	80.32
MemNet (8)	72.05	80.14
MemNet (9)	72.21	**80.95**

Table 2: Classification accuracy of different methods on laptop and restaurant datasets. Best scores in each group are in bold.

pects (e.g. "*food*" and "*service*") might be different. It is somewhat disappointing that incorporating attention model over TDLSTM does not bring any improvement. We consider that each hidden vector of TDLSTM encodes the semantics of word sequence until the current position. Therefore, the model of TDLSTM+ATT actually selects such mixed semantics of word sequence, which is weird and not an intuitive way to selectively focus on parts of contexts. Different from TDLSTM+ATT, the proposed memory network approach removes the recurrent calculator over word sequence and directly apply attention mechanism on context word representations.

We can also find that the performance of ContextAVG is very poor, which means that assigning the same weight/importance to all the context words is not an effective way. Among all our models from single hop to nine hops, we can observe that using more computational layers could generally lead to better performance, especially when the number of hops is less than six. The best performances are achieved when the model contains seven and nine hops, respectively. On both datasets, the proposed approach could obtain comparable accuracy compared to the state-of-art feature-based SVM system.

4.3 Runtime Analysis

We study the runtime of recurrent neural models and the proposed deep memory network approach with different hops. We implement all these approaches based on the same neural network infrastructure, use the same 300-dimensional Glove word vectors, and run them on the same CPU server.

Method	Time cost
LSTM	417
TDLSTM	490
TDLSTM + ATT	520
MemNet (1)	3
MemNet (2)	7
MemNet (3)	9
MemNet (4)	15
MemNet (5)	20
MemNet (6)	24
MemNet (7)	26
MemNet (8)	27
MemNet (9)	29

Table 3: Runtime (seconds) of each training epoch on the restaurant dataset.

The training time of each iteration on the restaurant dataset is given in Table 3. We can find that LSTM based recurrent models are indeed computationally expensive, which is caused by the complex operations in each LSTM unit along the word sequence. Instead, the memory network approach is simpler and evidently faster because it does not need recurrent calculators of sequence length. Our approach with nine hops is almost 15 times faster than the basic LSTM model.

4.4 Effects of Location Attention

As described in Section 3.4, we explore four strategies to integrate location information into the attention model. We incorporate each of them separately into the basic content-based attention model. It is helpful to restate that the difference between four location-based attention models lies in the usage of location vectors for context words. In Model 1 and Model 2, the values of location vectors are fixed and calculated in a heuristic way. In Model 3 and Model 4, location vectors are also regarded as the parameters and jointly learned along with other parameters in the deep memory network.

(a) Aspect: *service*, Answer: -1, Prediction: -1

	hop 1	hop 2	hop 3	hop 4	hop 5
great	0.20	0.15	0.14	0.13	0.23
food	0.11	0.07	0.08	0.12	0.06
but	0.20	0.10	0.10	0.12	0.13
the	0.03	0.07	0.08	0.12	0.06
was	0.08	0.07	0.08	0.12	0.06
dreadful	0.20	0.45	0.45	0.28	0.40
!	0.19	0.08	0.08	0.12	0.07

(b) Aspect: *food*, Answer: +1, Prediction: -1

	hop 1	hop 2	hop 3	hop 4	hop 5
great	0.22	0.12	0.14	0.12	0.20
but	0.21	0.11	0.10	0.11	0.12
the	0.03	0.11	0.08	0.11	0.06
service	0.11	0.11	0.08	0.11	0.06
was	0.04	0.11	0.08	0.11	0.06
dreadful	0.22	0.32	0.45	0.32	0.43
!	0.16	0.11	0.08	0.11	0.07

Table 4: Examples of attention weights in different hops for aspect level sentiment classification. The model only uses content attention. The hop columns show the weights of context words in each hop, indicated by values and gray color. This example shows the results of sentence "*great food but the service was dreadful!*" with "*food*" and "*service*" as the aspects.

(a) Aspect: *service*, Answer: -1, Prediction: -1

	hop 1	hop 2	hop 3	hop 4	hop 5
great	0.08	0.10	0.10	0.09	0.09
food	0.08	0.07	0.07	0.07	0.07
but	0.10	0.15	0.16	0.13	0.11
the	0.07	0.07	0.07	0.07	0.07
was	0.07	0.07	0.07	0.07	0.07
dreadful	0.52	0.48	0.48	0.50	0.52
!	0.07	0.07	0.07	0.07	0.07

(b) Aspect: *food*, Answer: +1, Prediction: +1

	hop 1	hop 2	hop 3	hop 4	hop 5
great	0.31	0.26	0.32	0.28	0.32
but	0.14	0.18	0.15	0.18	0.15
the	0.08	0.05	0.08	0.05	0.07
service	0.09	0.09	0.09	0.08	0.09
was	0.09	0.08	0.09	0.08	0.08
dreadful	0.18	0.21	0.18	0.22	0.19
!	0.11	0.12	0.10	0.11	0.10

Table 5: Examples of attention weights in different hops for aspect level sentiment classification. The model also takes into account of the location information (Model 2). This example is as same as the one we use in Table 4.

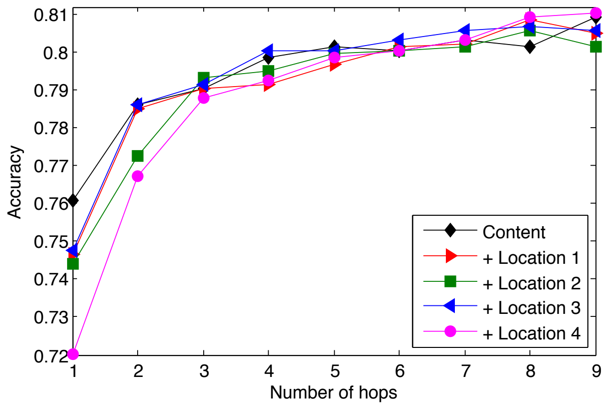

Figure 2: Classification accuracy of different attention models on the restaurant dataset.

Figure 2 shows the classification accuracy of each attention model on the restaurant dataset. We can find that using multiple computational layers could consistently improve the classification accuracy in all these models. All these models perform comparably when the number of hops is larger than five. Among these four location-based models, we prefer Model 2 as it is intuitive and has less computation cost without loss of accuracy. We also find that Model 4 is very sensitive to the choice of neural gate. Its classification accuracy decreases by almost 5 percentage when the $sigmoid$ operation over location vector is removed.

4.5 Visualize Attention Models

We visualize the attention weight of each context word to get a better understanding of the deep memory network approach. The results of context-based model and location-based model (Model 2) are given in Table 4 and Table 5, respectively.

From Table 4(a), we can find that in the first hop the context words "*great*", "*but*" and "*dreadful*" contribute equally to the aspect "*service*". While after the second hop, the weight of "*dreadful*" increases and finally the model correctly predict the polarity towards "*service*" as negative. This case shows the effects of multiple hops. However, in Table 4(b), the content-based model also gives a larger weight to "*dreadful*" when the target we focus on is "*food*". As a result, the model incorrectly predicts the polarity towards "*food*" as negative. This phenomenon might be caused by the neglect of location information. From Table 5(b), we can find that the weight of "*great*" is increased when the location of context word is considered. Accordingly, Model 2 predicts the correct sentiment label towards "*food*". We believe that location-enhanced model captures both content and location information. For instance, in Table 5(a) the closest context words of the aspect "*service*" are "*the*" and "*was*", while "*dreadful*" has the largest weight.

4.6 Error Analysis

We carry out an error analysis of our location enhanced model (Model 2) on the restaurant dataset, and find that most of the errors could be summarized as follows. The first factor is non-compositional sentiment expression. This model regards single context word as the basic computational unit and cannot handle this situation. An example is "*dessert was also to die for!*", where the aspect is underlined. The sentiment expression is "*die for*", whose meaning could not be composed from its constituents "*die*" and "*for*". The second factor is complex aspect expression consisting of many words, such as "*ask for the round corner table next to the large window.*" This model represents an aspect expression by averaging its constituting word vectors, which could not well handle this situation. The third factor is sentimental relation between context words such as negation, comparison and condition. An example is "*but dinner here is never disappointing, even if the prices are a bit over the top*". We believe that this is caused by the weakness of weighted average compositional function in each hop. There are also cases when comparative opinions are expressed such as "*i 've had better japanese food at a mall food court*".

5 Related Work

This work is connected to three research areas in natural language processing. We briefly describe related studies in each area.

5.1 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 (Pontiki et al., 2014). 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. Experts could design effective feature templates and make use of external resources like parser and sentiment lexicons (Kiritchenko et al., 2014; Wagner et al., 2014). In recent years, neural network approaches (Dong et al., 2014; Lakkaraju et al., 2014; Nguyen and Shirai, 2015; Tang et al., 2015a) are of growing attention for their capacity to learn powerful text representation from data. However, these neural models (e.g. LSTM) are computationally expensive, and could not explicitly reveal the importance of context evidences with regard to an aspect. Instead, we develop simple and fast approach that explicitly encodes the context importance towards a given aspect. It is worth noting that the task we focus on differs from fine-grained opinion extraction, which assigns each word a tag (e.g. B,I,O) to indicate whether it is an aspect/sentiment word (Choi and Cardie, 2010; Irsoy and Cardie, 2014; Liu et al., 2015). The aspect word in this work is given as a part of the input.

5.2 Compositionality in Vector Space

In NLP community, compositionality means that the meaning of a composed expression (e.g. a phrase/sentence/document) comes from the meanings of its constituents (Frege, 1892). Mitchell and Lapata (2010) exploits a variety of addition and multiplication functions to calculate phrase vector. Yessenalina and Cardie (2011) use matrix multiplication as compositional function to compute vectors for longer phrases. To compute sentence representation, researchers develop denoising autoencoder (Glorot et al., 2011), convolutional neural network (Kalchbrenner et al., 2014; Kim, 2014; Yin and Schütze, 2015), sequence based recurrent neural models (Sutskever et al., 2014; Kiros et al., 2015; Li et al., 2015b) and tree-structured neural networks (Socher et al., 2013; Tai et al., 2015; Zhu et al., 2015). Several recent studies calculate continuous representation for documents with neural networks (Le and Mikolov, 2014; Bhatia et al., 2015; Li et al., 2015a; Tang et al., 2015b; Yang et al., 2016).

5.3 Attention and Memory Networks

Recently, there is a resurgence in computational models with attention mechanism and explicit memory to learn representations of texts (Graves et al., 2014; Weston et al., 2014; Sukhbaatar et al., 2015;

Bahdanau et al., 2015). In this line of research, memory is encoded as a continuous representation and operations on memory (e.g. reading and writing) are typically implemented with neural networks. Attention mechanism could be viewed as a compositional function, where lower level representations are regarded as the memory, and the function is to choose "where to look" by assigning a weight/importance to each lower position when computing an upper level representation. Such attention based approaches have achieved promising performances on a variety of NLP tasks (Luong et al., 2015; Kumar et al., 2015; Rush et al., 2015).

6 Conclusion

We develop deep memory networks that capture importances of context words for aspect level sentiment classification. Compared with recurrent neural models like LSTM, this approach is simpler and faster. Empirical results on two datasets verify that the proposed approach performs comparable to state-of-the-art feature based SVM system, and substantively better than LSTM architectures. We implement different attention strategies and show that leveraging both content and location information could learn better context weight and text representation. We also demonstrate that using multiple computational layers in memory network could obtain improved performance. Our potential future plans are incorporating sentence structure like parsing results into the deep memory network.

Acknowledgments

We would especially want to thank Xiaodan Zhu for running their system on our setup. We greatly thank Yaming Sun for tremendously helpful discussions. We also thank the anonymous reviewers for their valuable comments. This work was supported by the National High Technology Development 863 Program of China (No. 2015AA015407), National Natural Science Foundation of China (No. 61632011 and No.61273321).

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. *International Conference on Learning Representations (ICLR)*.

Parminder Bhatia, Yangfeng Ji, and Jacob Eisenstein. 2015. Better document-level sentiment analysis from rst discourse parsing. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2212–2218.

Yejin Choi and Claire Cardie. 2010. Hierarchical sequential learning for extracting opinions and their attributes. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 269–274. Association for Computational Linguistics.

Li Dong, Furu Wei, Chuanqi Tan, Duyu Tang, Ming Zhou, and Ke Xu. 2014. Adaptive recursive neural network for target-dependent twitter sentiment classification. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 49–54.

Gottlob Frege. 1892. On sense and reference. *Ludlow (1997)*, pages 563–584.

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. 2011. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 513–520.

Alex Graves, Greg Wayne, and Ivo Danihelka. 2014. Neural turing machines. *arXiv preprint arXiv:1410.5401*.

Ozan Irsoy and Claire Cardie. 2014. Opinion mining with deep recurrent neural networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 720–728.

Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 655–665.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1746–1751.

Svetlana Kiritchenko, Xiaodan Zhu, Colin Cherry, and Saif Mohammad. 2014. Nrc-canada-2014: Detecting aspects and sentiment in customer reviews. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 437–442.

Ryan Kiros, Yukun Zhu, Ruslan R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-thought vectors. In *Advances in Neural Information Processing Systems*, pages 3276–3284.

Ankit Kumar, Ozan Irsoy, Jonathan Su, James Bradbury, Robert English, Brian Pierce, Peter Ondruska, Ishaan

Gulrajani, and Richard Socher. 2015. Ask me anything: Dynamic memory networks for natural language processing. *arXiv preprint arXiv:1506.07285*.

Himabindu Lakkaraju, Richard Socher, and Chris Manning. 2014. Aspect specific sentiment analysis using hierarchical deep learning. In *NIPS Workshop on Deep Learning and Representation Learning*.

Quoc V. Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *Proceedings of The 31nd International Conference on Machine Learning*, pages 1188–1196.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *Nature*, 521(7553):436–444.

Jiwei Li, Thang Luong, and Dan Jurafsky. 2015a. A hierarchical neural autoencoder for paragraphs and documents. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics*, pages 1106–1115.

Jiwei Li, Thang Luong, Dan Jurafsky, and Eduard Hovy. 2015b. When are tree structures necessary for deep learning of representations? In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2304–2314.

Pengfei Liu, Shafiq Joty, and Helen Meng. 2015. Fine-grained opinion mining with recurrent neural networks and word embeddings. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1433–1443.

Bing Liu. 2012. Sentiment analysis and opinion mining. *Synthesis Lectures on Human Language Technologies*, 5(1):1–167.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119.

Jeff Mitchell and Mirella Lapata. 2010. Composition in distributional models of semantics. *Cognitive Science*, 34(8):1388–1429.

Thien Hai Nguyen and Kiyoaki Shirai. 2015. Phrasernn: Phrase recursive neural network for aspect-based sentiment analysis. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2509–2514.

Bo Pang and Lillian Lee. 2008. Opinion mining and sentiment analysis. *Foundations and trends in information retrieval*, 2(1-2):1–135.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1532–1543.

Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. 2014. Semeval-2014 task 4: Aspect based sentiment analysis. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 27–35.

Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 379–389.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642.

Sainbayar Sukhbaatar, Arthur Szlam, Jason Weston, and Rob Fergus. 2015. End-to-end memory networks. In *Advances in Neural Information Processing Systems*, pages 2431–2439.

Yaming Sun, Lei Lin, Duyu Tang, Nan Yang, Zhenzhou Ji, and Xiaolong Wang. 2015. Modeling mention, context and entity with neural networks for entity disambiguation. *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence (IJCAI 2015)*, pages 1333–1339.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Kai Sheng Tai, Richard Socher, and Christopher D. Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics*, pages 1556–1566.

Duyu Tang, Bing Qin, Xiaocheng Feng, and Ting Liu. 2015a. Target-Dependent Sentiment Classification with Long Short Term Memory. *ArXiv preprint arXiv:1512.01100*.

Duyu Tang, Bing Qin, and Ting Liu. 2015b. Document modeling with gated recurrent neural network for sentiment classification. *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1422–1432.

Duy-Tin Vo and Yue Zhang. 2015. Target-dependent twitter sentiment classification with rich automatic features. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence (IJCAI 2015)*, pages 1347–1353.

Joachim Wagner, Piyush Arora, Santiago Cortes, Utsab Barman, Dasha Bogdanova, Jennifer Foster, and Lamia Tounsi. 2014. Dcu: Aspect-based polarity classification for semeval task 4. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 223–229.

Jason Weston, Sumit Chopra, and Antoine Bordes. 2014. Memory networks. *arXiv preprint arXiv:1410.3916*.

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

Ainur Yessenalina and Claire Cardie. 2011. Compositional matrix-space models for sentiment analysis. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 172–182.

Wenpeng Yin and Hinrich Schütze. 2015. Multichannel variable-size convolution for sentence classification. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning*, pages 204–214.

Xiaodan Zhu, Parinaz Sobhani, and Hongyu Guo. 2015. Long short-term memory over tree structures. In *Proceedings of The 32nd International Conference on Machine Learning*, pages 1604–1612.