Content Attention Model for Aspect Based Sentiment Analysis

Qiao Liu
University of Electronic Science and Technology of China
Chengdu, China
qliu@uestc.edu.cn

Haibin Zhang
University of Electronic Science and Technology of China
Chengdu, China
herb.zhang@std.uestc.edu.cn

Yifu Zeng
University of Electronic Science and Technology of China
Chengdu, China
ifz@std.uestc.edu.cn

Ziqi Huang
University of Electronic Science and Technology of China
Chengdu, China
2016220401037@std.uestc.edu.cn

Zufeng Wu
University of Electronic Science and Technology of China
Chengdu, China
wuzufeng@uestc.edu.cn

ABSTRACT

Aspect based sentiment classification is a crucial task for sentiment analysis. Recent advances in neural attention models demonstrate that they can be helpful in aspect based sentiment classification tasks, which can help identify the focus words in human. However, according to our empirical study, prevalent content attention mechanisms proposed for aspect based sentiment classification mostly focus on identifying the sentiment words or shifters, without considering the relevance of such words with respect to the given aspects in the sentence. Therefore, they are usually insufficient for dealing with multi-aspect sentences and the syntactically complex sentence structures. To solve this problem, we propose a novel content attention based aspect based sentiment classification model, with two attention enhancing mechanisms: *sentence-level content attention mechanism* is capable of capturing the important information about given aspects from a global perspective, whiles the *context attention mechanism* is responsible for simultaneously taking the order of the words and their correlations into account, by embedding them into a series of *customized memories*. Experimental results demonstrate that our model outperforms the state-of-the-art, in which the proposed mechanisms play a key role.

CCS CONCEPTS

• **Information systems** → **Sentiment analysis**; • **Computing methodologies** → *Information extraction*; *Neural networks*;

KEYWORDS

Sentiment Analysis, Aspect Based, Attention Mechanism

ACM Reference Format:
Qiao Liu, Haibin Zhang, Yifu Zeng, Ziqi Huang, and Zufeng Wu. 2018. Content Attention Model for Aspect Based Sentiment Analysis. In *WWW 2018: The 2018 Web Conference, April 23–27, 2018, Lyon, France.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3178876.3186001

WWW 2018, April 23–27, 2018, Lyon, France

ACM ISBN 978-1-4503-5639-8/18/04.
https://doi.org/10.1145/3178876.3186001

1 INTRODUCTION

Aspect based sentiment analysis is an important subtask of sentiment analysis (SA), which is also a central concern of the semantic web and the computational linguistics community in recent years [25]. The goal of aspect based SA is to identify the aspects of given entities (aspect extraction), and determine the sentiment expressed for each aspect (a.k.a. aspect based sentiment classification,). In this paper, we focus on the problem of aspect based sentiment classification (ABSC), the aim is to determine whether user opinions conveyed in comments/tweets on specific aspects are positive, negative, or neutral [16]. As an example, consider the following sample sentence taken from the SemEval 2014 restaurant dataset:

> *Looking around, I saw a room full of New Yorkers enjoying a real meal in a real restaurant, not a clubhouse of the fabulous trying to be seen.*

The aspects for this sentence are "room", "meal", and "clubhouse", the expected outputs of the aspect based sentiment classifier are intend to be neutral, positive and negative respectively.

Recent advances in neural network based ABSC models have deeply reshaped the research because of their capability of learning to predict in an end-to-end manner [7, 36, 39]. In these studies, context words are usually regarded with equal importance across the mentioned aspects. However, in many cases only a subset of the context words would be relevant to the sentiment polarity of a given aspect [37]. As one could see from the above example, the word "enjoying" is important for determining the sentiment polarity w.r.t the aspect "meal", but the word "fabulous" seems to be irrelevant to it. If the classification model can not differentiate aspect words, it would be problematic for practical use.

To solve this problem, some neural attention models were introduced to this area [4, 19, 37, 41]. However, according to our empirical study, there are some common problems shared by these existing neural attention models. **Firstly**, most of the attention modeling strategies in this area only consider a *partial* of the context information in a sentence without considering the relevance or contribution of each context word to the given aspect, which we consider could be a serious problem since most of the sentiment words and shifters will be focused but not all of them are relative to the given aspect. Take the following example,

> *The mini's body hasn't changed since late 2010- and for a good reason.*

In this statement, the words "n't", "good" and "late" would be captured by a typical attention model as *focused words*, which indicate an overall negative sentiment on aspect word "body". However, one could tell that among these words, only the word "good" is *really* associated with the sentimental polarity of the "body", therefore the correct sentiment conveyed should be positive. In this study, we argue that such a *short-sighted* behavior will cause a significant loss in the predictive accuracy of the classifier.

Secondly, most of the existing attention models only consider the "words-level" attention calculation without taking into account the overall meaning conveyed by the sentence. However, for complex sentences such as ironical or sarcastic statements which are commonly seen in practical user comments, the classifier may need more precise information (about the entire sentence) to predict the correct results. For example, in the following sentence:

> *Maybe the mac os improvement were not the product they want to offer.*

a typical attention model would allocate a high attention weight to the word "improvement", which is an obvious *polar word* with positive sentiment tendency closely related to the given aspect "mac os" in that statement. However, this is apparently an ironic statement, which expresses a negative sentiment on "mac os".

Thirdly, a sentence might contain multiple aspects of a given topic. Therefore, each word may have a different importance in a sentence depending on the given aspect. Previous works have taken into consideration this problem with prevalent solutions including the hidden state based model [41] and the memory based model [4, 37]. All these solutions have their cons. For the hidden state based model, the sentiment feature contained in a word representation is an *inexplicit* word sequence mixed feature. The memory based model is usually based on position attention mechanism which assumes that a context word closer to the aspect should be more important, this assumption is not true in some cases.

In this paper, we consider all these problems systematically. In order to solve the first and second problems, we propose a *sentence-level content attention mechanism*. When calculating the attention weights, our model does not only consider the information conveyed by each word and aspect in the sentence, but also considers the whole meaning of the full sentence. Based on this mechanism, our sentence-level content attention module (SAM) can capture the important information about a given aspect from a global perspective and embeds the full sentence into the output embedding vector. The output vector of the SAM can be treated as an aspect-specific sentence representation, we argue that this will improve the ability of the ABSC model to handle complex sentences.

In order to tackle the third problem, we propose a *context attention mechanism*, which does not only considers the order of the word sequence, but also takes into account the correlations between the words and the aspect. Based on this, our context attention based memory module (CAM) provides a *customized memory* for each aspect, which will be updated in a sequential manner.

The major contributions of this paper are: (1) We develop a neural **C**ontent **a**ttention **b**ased **a**spect based **s**entiment **c**lassification model called **Cabasc**. The framework is illustrated in Fig. 1. (2) We propose two novel attention modeling mechanisms to tackle the semantic-mismatch problem discussed above, and we also carry out comparison studies with respect to the proposed model to verify the validity of these mechanisms. (3) We evaluate our model on three datasets, two of which are the laptop and restaurant datasets from SemEval 2014 [25], the other one is the twitter dataset introduced by Dong et al. [7]. Experimental results show that our model can help improve aspect based sentiment classification accuracy.

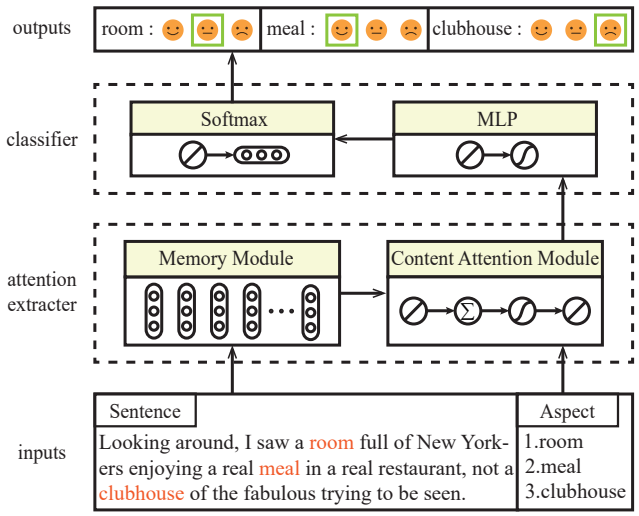


Figure 1: The framework of content attention based aspect based sentiment classification model.

2 RELATED WORK

Aspect based sentiment analysis is a fundamental task in sentiment analysis research field [25, 40], which includes several core subtasks: aspect extraction [5, 20, 26], opinion identification [10, 17] and aspect based sentiment classification [11, 13, 37].

Some previous studies have try to solve these subtasks jointly [22, 27], dedicating most of the research work in solving an individual subtask. This is attributed to the fact that the remain research tasks are still challenging. In this study, we focus on improving the attention modeling mechanism for solving the aspect based sentiment classification problem. Some related works are briefly introduced in the following section.

2.1 Neural Network Models for ABSC Task

Aspect based sentiment classification aims at determining the sentiment polarity of a given aspect [16, 25]. Conventional methods usually come from the computational linguistic community, which are mostly machine learning models based on hand-crafted lexicons and syntactic features [11, 13]. The performance of such models is highly dependent on the quality of the artifact features which is labor intensive. Therefore, recent research has turned its focus on developing end-to-end neural network models.

Recursive neural networks (RecNNs) were firstly introduced into this field by Dong et al. [7], they propose an adaptive recursive neural network which can adaptively propagate the sentiments of context words to the target. It has been demonstrated that the RecNNs are effective in obtaining sentence representations from

the recursive structure of the text [23], but they may suffer from syntax parsing errors which are common in practice [39, 43]. In contrast, the recurrent neural networks (RNNs) have been proven to be effective in many (language) sequence learning tasks, hence, most of state-of-the-art solutions are based on RNNs [29, 36, 43].

Tang et al. [36] propose a target-dependent long short-term memory network model (TD-LSTM), which learns representations directly from the left and right context w.r.t. the given aspect by making use of two LSTM networks respectively. Zhang et al. [43] used gated neural network structures to model the syntax and semantics in tweets and interaction between the surrounding contexts and the target. These RNNs based models have achieved promising results, but they are computationally expensive. Some other researchers have try to improve the computational efficiency with other deep learning architectures. Vo and Zhang [39] used neural pooling functions to automatically extract features from word embeddings. Tang et al. [37] developed a deep memory network based on a multi-hop attention mechanism, which is effective and computationally inexpensive. It is worth mentioning that this model is also a neural content attention model that is closely related to our model, which is discussed in the section below.

2.2 Neural Attention Models for ABSC Task

As mentioned in the previous section, most of the neural network models proposed for solving the ABSC problem do not take into account the correlations between the context words and the given aspect. Therefore they easily suffer from the semantic mismatching problem. To solve this problem a variety of attention models have recently been introduced to this area. Attention mechanism has been successfully applied to many natural language processing tasks [12], such as neural machine translation [2, 18], question answering [14, 34], textual entailment recognizing [28], sentence summarization [30] and machine comprehension [9, 31]. A significant advantage of the neural attention model is that it can *automatically* identify the relevant information w.r.t. a specific target in a source sentence, which can be directly used for improving the quality of the feature extraction results of the neural leaning models [18].

Some of the representative neural attention models proposed for ABSC task are discussed below. Wang et al. [41] proposed an LSTM based single-hop attention model, which takes the concatenations of the aspect and the word embeddings as input and uses the LSTM hidden states for attention computation. Ma et al. [19] propose an interactive attention mechanism, which interactively learns attentions from the context and the aspect. We call these models *hidden state based* attention models. The neural attention models that are most closely related to this work are probably [37] and [4], we call them *memory based position-aware* attention models.

Tang et al. [37] compute the attentions based on the relative distance of the words w.r.t the given aspect in a sentence. But the authors implicitly assume that a word located closer to the aspect should be given more credits, which we consider is arguable in practice. For instance, in "*The two waitress's looked like they had been sucking on lemons.*", the phrase "sucking on lemons" is more important than other words such as the closer mention "like" for determining the sentiment polarity of the aspect "waitress". In our model, the proposed context attention mechanism can help alleviate this contradiction by explicitly considering correlation between each context word and the given aspect. Chen et al. [4] propose a recurrent attention mechanism based on *customized memory* for each aspect to capture sentiment features separated by a long distance, which is similar in part to our memory mechanism. However, due to the nature of the LSTM adopted in their model, it can only consider partial sentence information when calculating attention weights. In our model, the proposed sentence-level content attention mechanism can provide general and effective remedy for the *short-sight* problem of these *memory* models.

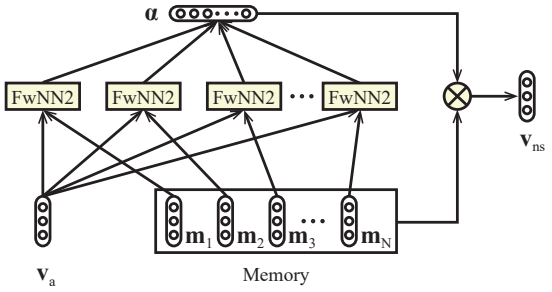


Figure 2: The content attention module of BaseA model.

3 MODEL

The proposed model is introduced in this section. Before dive into the details of our proposed Cabasc model, we first introduce three baseline models used to verify the validity of the above mentioned mechanisms. The training process is also covered in this section.

3.1 Baseline Model A (BaseA)

The baseline model A is a basic model, and the subsequent models are derived from the model. Let $S = \{s_1, s_2, \ldots, s_i, \ldots, s_{i+L}, \ldots, s_N\}$ denote an input sentence which consists of N words and $S_a = \{s_i, \ldots, s_{i+L}\}$ denote the given aspect that appears in the input sentence, which consists of L words. The goal of our models is to predict the polarity of sentence S towards the aspect S_a.

Inputs. We use $\mathbb{L} \in \mathbb{R}^{d \times |V|}$ to be a word embedding matrix for a vocabulary V, where d is the dimension of a word vector. The matrix can be generated by an unsupervised method [21, 24], a distant-supervised method [38], or a random initialization method. The word s_i in the sentence S is mapped into a low-dimensional, real-valued embedding $\mathbf{e}_i \in \mathbb{R}^d$ formally defined as,

$$\mathbf{e}_i = \mathbb{L}\mathbf{o}_i \tag{1}$$

where $\mathbf{o}_i$ is the one-hot vector of s_i, the length is $|V|$. After that, we get a list of vectors $\mathbf{E} = \{\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_i, \ldots, \mathbf{e}_{i+L}, \ldots, \mathbf{e}_N\}$ corresponding to sentence S. If aspect S_a is a single word the aspect representation $\mathbf{v}_a$ is the embedding of aspect word. If S_a is a phrase, $\mathbf{v}_a$ takes the mean of aspect word embeddings, this simple representation has proven to be effective in a number of tasks, including aspect based sentiment classification task [32, 35, 37].

Memory Module. The vectors in $\mathbf{E}$ are stacked one after another to construct a long-term memory $\mathbf{M} = \{\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_N\}$, where

$\mathbf{M} \in \mathbb{R}^{d \times N}$. $\mathbf{M}$ stores information of the input sentence, where the contents of memory are treated as learnable variables and the model learns how to use the memory for prediction [8, 42].

Content Attention Module. Intuitively, words in a sentence have different contributions to the sentiment polarity of a given aspect occurring in the sentence. And the contribution of a word to the sentiment polarity of different aspects in the sentence should be different [37]. Bahdanau et al. [2] use an alignment model (attention model) to search for a set of important positions in a sentence to capture the most relevant information. Inspired by the effectiveness of this method, we use a content attention module (Fig. 2) to retrieve information that is most relevant to the sentiment polarity of a given aspect S_a from the memory $\mathbf{M}$ constructed above.

For calculating the attention weight of the memory slice $\mathbf{m}_i$ of $\mathbf{M}$, we use feed forward neural networks with two inputs (FwNN2) to score how important the word s_i is to the sentiment polarity of aspect S_a. All of the feed forward neural networks share parameters with each other. The score is calculated based on $\mathbf{m}_i$ and aspect representation $\mathbf{v}_a$, formally defined as,

$$c_i = \mathbf{W}_1 \tanh(\mathbf{W}_2 \mathbf{m}_i + \mathbf{W}_3 \mathbf{v}_a + \mathbf{b}_1) \tag{2}$$

where matrices $\mathbf{W}_1 \in \mathbb{R}^{1 \times d}$, $\mathbf{W}_2 \in \mathbb{R}^{d \times d}$ and $\mathbf{W}_3 \in \mathbb{R}^{d \times d}$ and vector $\mathbf{b}_1 \in \mathbb{R}^d$ are model parameters. After obtaining $\{c_1, c_2, \ldots, c_N\}$, the attention weight α_i of s_i is computed by

$$\alpha_i = \frac{\exp(c_i)}{\sum_{j=1}^{N} \exp(c_j)} \tag{3}$$

All the calculated attention weights form an attention weight vector $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_N)$ of the memory $\mathbf{M}$ with given aspect S_a. Finally, the aspect-specific sentence representation $\mathbf{v}_{ns}$ is calculated as follows, where $\mathbf{v}_{ns} \in \mathbb{R}^d$.

$$\mathbf{v}_{ns} = \mathbf{M}\boldsymbol{\alpha} \tag{4}$$

MLP. Depth is an important part of deep learning methods the multiple non-linear layers help to yield more abstract and useful representations [3]. We use an MLP with one hidden layer which is a simple and effective way to increase the depth of the model. The purpose is to use the hidden layer to represent the inputs in a more predictable way [15]. The MLP takes the aspect-specific sentence representation $\mathbf{v}_{ns}$ as its input, the module is defined as,

$$\mathbf{v}_{ms} = g(\mathbf{W}_4 \mathbf{v}_{ns} + \mathbf{b}_2) \tag{5}$$

where $\mathbf{v}_{ms} \in \mathbb{R}^d$ is the output of the MLP module, $\mathbf{W}_4 \in \mathbb{R}^{d \times d}$ is a weight matrix and $\mathbf{b}_2 \in \mathbb{R}^d$ is a bias vector, $g(\cdot)$ is a non-linear activation function and we use tanh here.

Softmax. The output $\mathbf{v}_{ms}$ of MLP is converted by a linear layer to a real-valued vector of length $|C|$, where C is the collection of possible sentiment categories. Then the vector is fed into a softmax layer to predict the sentiment polarity of the aspect S_a, formally,

$$pred = \text{softmax}(\mathbf{W}_5 \mathbf{v}_{ms} + \mathbf{b}_3) \tag{6}$$

where $pred \in \mathbb{R}^{|C|}$ is a conditional probability distribution over C, weight matrix $\mathbf{W}_5 \in \mathbb{R}^{|C| \times d}$ and bias vector $\mathbf{b}_3 \in \mathbb{R}^{|C|}$ are parameters for the linear layer.

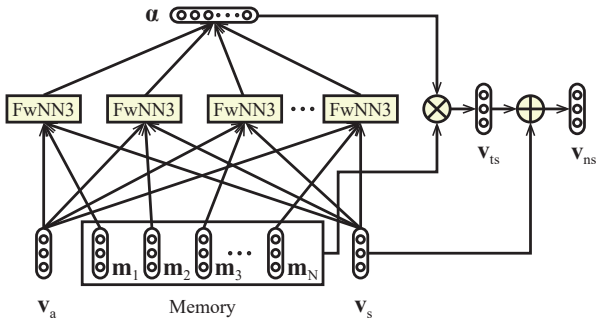


Figure 3: The sentence-level content attention module of BaseB model.

3.2 Baseline Model B (BaseB)

Content attention module in BaseA calculates an attention weight without considering the whole meaning of the entire sentence. However, considering only part of the sentence information may lead to some focused words not being related to the given aspect. This may affect the model's sentiment polarity prediction. In addition, for complex sentences, the aspect-specific sentence representation produced by content attention module may not have the overall meaning conveyed by the sentence will not be enough for the correct sentiment classification. In order to address these issues, we propose the use of the sentence information to enhance BaseA. The new model is BaseB, which adds two extensions to the content attention module of BaseA by using the sentence-level content attention mechanism, shown in Fig. 3.

Sentence-level Content Attention Module (SAM). The first extension adds the sentence representation to the calculation of the attention weight. The sentence representation smooths the score of the importance of a word for the sentiment polarity of an aspect from a global perspective, so as to improve the ability to accurately capture important sentiment features. A feed forward neural network with three inputs (FwNN3) is used to calculate score c_i of word s_i, that is,

$$c_i = \mathbf{W}_6 \tanh(\mathbf{W}_7 \mathbf{m}_i + \mathbf{W}_8 \mathbf{v}_a + \mathbf{W}_9 \mathbf{v}_s + \mathbf{b}_4) \tag{7}$$

where $\mathbf{m}_i \in \mathbb{R}^d$ is memory slice of s_i, $\mathbf{v}_a \in \mathbb{R}^d$ is the aspect representation, $\mathbf{v}_s \in \mathbb{R}^d$ is the sentence representation, $\mathbf{W}_6 \in \mathbb{R}^{1 \times d}$, $\mathbf{W}_7 \in \mathbb{R}^{d \times d}$, $\mathbf{W}_8 \in \mathbb{R}^{d \times d}$ and $\mathbf{W}_9 \in \mathbb{R}^{d \times d}$ are weight matrices and $\mathbf{b}_4 \in \mathbb{R}^d$ is a bias vector. We take the average of the embeddings of words participating in the sentence as sentence representation to preserve the sentence information, which is proved to be surprisingly effective [1]. All of the calculated scores of the sentence are denoted as $\{c_1, c_2, \ldots, c_N\}$.

After that, the Eq. (3) is used to compute the elements in the attention weight vector $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_N)$ of the sentence S. The output embedding vector $\mathbf{v}_{ts}$ is calculated by,

$$\mathbf{v}_{ts} = \mathbf{M}\boldsymbol{\alpha} \tag{8}$$

where $\mathbf{v}_{ts} \in \mathbb{R}^d$, $\mathbf{M}$ is the memory built in the memory module.

The second extension is to embed the entire sentence into the output vector which could improve the ability of the model to handle complex sentences. We achieve this by adding the sentence

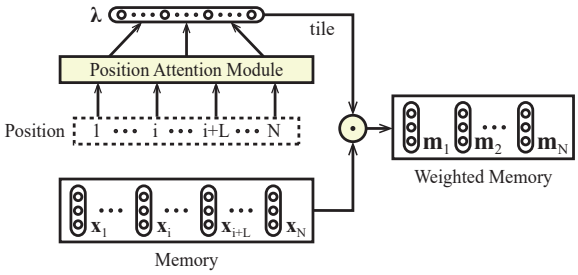


Figure 4: The position attention based memory module of BaseC model.

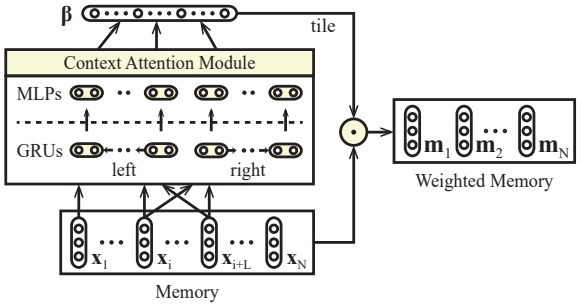


Figure 5: The context attention based memory module of Cabasc model.

representation $\mathbf{v}_s$ to the output vector $\mathbf{v}_{ts}$, computed as follows,

$$\mathbf{v}_{ns} = \mathbf{v}_{ts} + \mathbf{v}_s \tag{9}$$

where $\mathbf{v}_{ns}$ is the aspect-specific sentence representation as the output of the sentence-level content attention module in which the noticed sentiment features are highlighted and the sentence information is embedded.

3.3 Baseline Model C (BaseC)

The same word has same memory slice as generated in the above memory module. However, considering that due to the diversity of the meanings of words, a word may indicate different relative polarity in different contexts and a sentence might contain multiple aspects of a given topic. Therefore, each word may have a different importance depending on the aspect discussed in that sentence. To ease these problems, we use the position attention mechanism [4, 37] to extend the memory module of BaseB to be position attention based memory module, constructing a customized memory for a given aspect of a sentence, as shown in Fig. 4. The intuition behind the position attention mechanism is that the words around the aspect have a greater impact on the sentiment polarity of the aspect. This model is called BaseC.

Position Attention Based Memory Module. The position of a context word is defined as its absolute distance with the aspect in the sentence and the position of an aspect word is regarded as 0. If aspect is a phrase, then the positions of left context words are calculated with the first word of aspect, whiles the positions of right context words are calculated with the last word. Let $\boldsymbol{\lambda} \in \mathbb{R}^N$ be the position weight vector of sentence S, the i-th element of the vector is calculated as:

$$\lambda_i = 1 - p_i/N \tag{10}$$

where p_i is the position of word $s_i \in S$ and N is the sentence length.

The memory $\mathbf{M}$ is weighted by the position attention weights to produce weighted memory $\mathbf{M}_w = (\mathbf{m}_{w1}, \mathbf{m}_{w2}, \ldots, \mathbf{m}_{wN})$. Memory slice $\mathbf{m}_{wi} \in \mathbb{R}^d$ is calculated as:

$$\mathbf{m}_{wi} = \mathbf{q}_i \odot \mathbf{m}_i \tag{11}$$

where $\mathbf{m}_i \in \mathbb{R}^d$ is memory slice of s_i in $\mathbf{M}$, $\mathbf{q}_i \in \mathbb{R}^d$ is a vector obtained by tiling λ_i, that is copying λ_i to d times to form the $\mathbf{q}_i$ vector. $\odot$ means element-wise multiplication. Then $\mathbf{M}_w$ is fed into sentence-level content attention module.

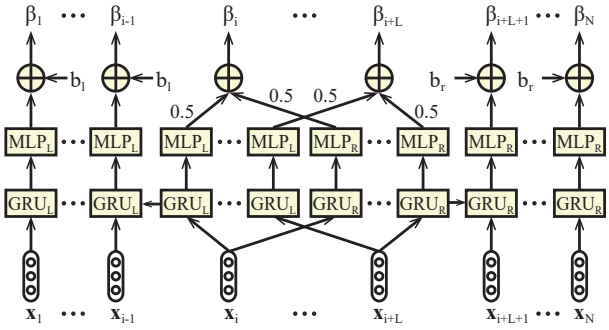


Figure 6: The context attention mechanism.

3.4 Content Attention Based Aspect Based Sentiment Classification Model (Cabasc)

We introduce the Cabasc model in this subsection. The position attention mechanism has been described in previous subsection, however, the mechanism only calculates the position attention weight based on the relative position of a context word to the aspect, ignoring the correlation between the word and the aspect which is more important than the relative position. In this way, the calculated attention weight is rough and is not flexible enough for this task. To address this deficiency, we propose a context attention mechanism in which the word order information, the aspect information and the correlation between the word and the aspect are modeled into the calculated attention weight. Cabasc is the same as BaseC except the memory module. The memory module of Cabasc is context attention based memory module as shown in Fig. 5, which differs from position attention based memory module which is used in the BaseC in that the Cabasc module uses a context attention mechanism rather than a position attention mechanism.

Context Attention Based Memory Module (CAM). The details of this module are visualized in Fig. 6. We first divide the input sentence S into the left context with aspect part $S_{ls} = \{s_1, \ldots, s_{i-1}, s_i, \ldots, s_{i+L}\}$ and the right context with aspect part $S_{rs} = \{s_i, \ldots, s_{i+L}, s_{i+L+1}, \ldots, s_N\}$. The embeddings of the words in these two parts are retrieved from matrix $\mathbb{L}$, and two lists of vectors $\mathbf{E}_{ls} = \{\mathbf{e}_1, \ldots, \mathbf{e}_{i-1}, \mathbf{e}_i, \ldots, \mathbf{e}_{i+L}\}$ and $\mathbf{E}_{rs} = \{\mathbf{e}_i, \ldots, \mathbf{e}_{i+L}, \mathbf{e}_{i+L+1}, \ldots, \mathbf{e}_N\}$ corresponding to S_{ls} and S_{rs} are obtained respectively.

In order to model the context information between the context words and aspect, as well as the aspect, we use two GRU neural networks [6], a left one GRU_L and a right one GRU_R, to model $\mathbf{E}_{ls}$ and $\mathbf{E}_{rs}$ respectively. The input of GRU_L is $\mathbf{E}_{ls}$, we run GRU_L from right to left. At time step t, the GRU_L observes an element $\mathbf{e}_t$ of $\mathbf{E}_{ls}$ and updates its internal hidden state $\mathbf{h}_t$ as follows:

$$\mathbf{r}_t = \sigma(\mathbf{W}_r\mathbf{e}_t + \mathbf{U}_r\mathbf{h}_{t-1}) \tag{12}$$

$$\mathbf{z}_t = \sigma(\mathbf{W}_z\mathbf{e}_t + \mathbf{U}_z\mathbf{h}_{t-1}) \tag{13}$$

$$\tilde{\mathbf{h}}_t = \tanh(\mathbf{W}_h\mathbf{e}_t + \mathbf{U}_h(\mathbf{r}_t \odot \mathbf{h}_{t-1})) \tag{14}$$

$$\mathbf{h}_t = \mathbf{z}_t \odot \mathbf{h}_{t-1} + (\mathbf{1} - \mathbf{z}_t) \odot \tilde{\mathbf{h}}_t \tag{15}$$

where $\mathbf{W}_r \in \mathbb{R}^{d\times d}$, $\mathbf{U}_r \in \mathbb{R}^{d\times d}$, $\mathbf{W}_z \in \mathbb{R}^{d\times d}$, $\mathbf{U}_z \in \mathbb{R}^{d\times d}$,$\mathbf{W}_h \in \mathbb{R}^{d\times d}$, $\mathbf{U}_h \in \mathbb{R}^{d\times d}$ are weight matrices and σ denotes the logistic sigmoid function. The update gate $\mathbf{r}_t$ controls the update extent of the output from a new hidden state $\tilde{\mathbf{h}}_t$ and the reset gate $\mathbf{z}_t$ controls how much information from the previous hidden state is allowed. After reading $\mathbf{E}_{ls}$, GRU_L produces a hidden state list $\mathbf{H}_{ls} = \{\mathbf{h}_{i+L_l}, \ldots, \mathbf{h}_{i_l}, \mathbf{h}_{i-1}, \ldots, \mathbf{h}_1\}$. GRU_R does the same thing, except that it takes $\mathbf{E}_{rs}$ as input and its run from left to right. The hidden state list of GRU_R is $\mathbf{H}_{rs} = \{\mathbf{h}_{i_r}, \ldots, \mathbf{h}_{i+L_r}, \mathbf{h}_{i+L+1}, \ldots, \mathbf{h}_N\}$.

An MLP is used to calculate the attention weight β_l of $\mathbf{h}_l$, where $\mathbf{h}_l$ is an element of $\mathbf{H}_{ls}$, formally,

$$\beta_l = \sigma(\mathbf{W}_{10}\mathbf{h}_l + b_5) + b_l \tag{16}$$

where $\mathbf{W}_{10} \in \mathbb{R}^{1\times d}$ is a weight matrix, $b_5 \in \mathbb{R}$ is a bias, and $b_l \in \mathbb{R}$ is a basic attention weight. An MLP that takes an element in $\mathbf{H}_{ls}$ as input is called MLP_L and all MLP_Ls share parameters. So the reverse attention weight list for $\mathbf{H}_{ls}$ is $\boldsymbol{\beta}_{ls} = \{\beta_1, \ldots, \beta_{i-1}, \beta_{i_l}, \ldots, \beta_{i+L_l}\}$. Then the attention weight list $\boldsymbol{\beta}_{rs} = \{\beta_{i_r}, \ldots, \beta_{i+L_r}, \beta_{i+L+1}, \ldots, \beta_N\}$ for $\mathbf{H}_{rs}$ is computed by a set of MLP_Rs. An attention weight β_r in $\boldsymbol{\beta}_{rs}$ is calculated as follows:

$$\beta_r = \sigma(\mathbf{W}_{11}\mathbf{h}_r + b_6) + b_r \tag{17}$$

where $\mathbf{W}_{11} \in \mathbb{R}^{1\times d}$ is a weight matrix, $b_6 \in \mathbb{R}$ is a bias, $\mathbf{h}_r \in \mathbf{H}_{rs}$, and $b_r \in \mathbb{R}$ is a basic attention weight. Each MLP_R takes an element of $\mathbf{H}_{rs}$ as input and these MLP_Rs share parameters.

The attention weights corresponding to the left context are extracted from $\boldsymbol{\beta}_{ls}$ to construct $\boldsymbol{\beta}_{lc} = \{\beta_1, \ldots, \beta_{i-1}\}$, the attention weights corresponding to the right context are extracted from $\boldsymbol{\beta}_{rs}$ to construct $\boldsymbol{\beta}_{rc} = \{\beta_{i+L+1}, \ldots, \beta_N\}$. $\boldsymbol{\beta}_a = \{\beta_i, \ldots, \beta_{i+L}\}$ are the attention weights corresponding to the aspect, one of the elements β_k is computed by,

$$\beta_k = (\beta_{k_l} + \beta_{k_r}) \times 0.5 \tag{18}$$

where $i \le k \le i + L$, $\beta_{k_l} \in \boldsymbol{\beta}_{ls}$ and $\beta_{k_r} \in \boldsymbol{\beta}_{rs}$.

We concatenate $\boldsymbol{\beta}_{lc}$, $\boldsymbol{\beta}_a$ and $\boldsymbol{\beta}_{rc}$ as the context attention weight vector $\boldsymbol{\beta} = \{\beta_1, \beta_2, \ldots, \beta_N\}$.

The weighted memory $\mathbf{M}_w = (\mathbf{m}_{w1}, \mathbf{m}_{w2}, \ldots, \mathbf{m}_{wN})$ is computed based on $\boldsymbol{\beta}$, formally,

$$\mathbf{m}_{wi} = \mathbf{y}_i \odot \mathbf{m}_i \tag{19}$$

where $\mathbf{m}_i$ is a memory slice of memory $\mathbf{M}$ and $\mathbf{y}_i \in \mathbb{R}^d$ is a vector obtained by tiling β_i, $\beta_i \in \boldsymbol{\beta}$. Accordingly, we take the average of $\mathbf{M}_w$ as sentence representation here.

Table 1: Statistics of the datasets.

Dataset		Pos.	Neg.	Neu.	Total
Laptop	train	994	870	464	2328
	test	341	128	169	638
Restaurant	train	2164	807	637	3608
	test	728	196	196	1120
Twitter	train	1561	1560	3127	6248
	test	173	173	346	692

3.5 Model Training

Our model is trained to minimize a cross-entropy loss objective in a supervised manner, the loss function is defined by,

$$loss = -\sum_i \log p_{t_i} \tag{20}$$

where p_{t_i} is the probability of the i-th training example as given by the model. We use back propagation to calculate the gradients of the parameters, and update them with stochastic gradient descent. The dropout technique is used to ease overfitting. We clamp the word embeddings with 300-dimensio-nal Glove[1] vectors [24] for our experiments, which the vocabulary size is 1.9M. We divide the development set from the training set and use it to select the hyper-parameters. The learning rate is set as 0.001. All the parameters in the model are initialized randomly with a normal distribution $N(0, 0.05^2)$. And the basic attention weights b_l and b_r are 0.5.

4 EXPERIMENTS AND DISCUSSIONS

4.1 Experimental Setting

Dataset. We experiment the proposed models on three publicly available datasets, with two from SemEval 2014 [25], which contain reviews of restaurant and laptop domains. The third one is a twitter dataset collected by Dong et al. [7]. The statistics of these datasets are presented in Table 1. Following Tang et al. [37], we remove conflict category in SemEval 2014 datasets to avoid datasets getting unbalanced. The evaluation metric is classification accuracy.

Baseline. We compare our models with the following baseline methods on three datasets:

SVM [13]: The classic SVM model using a series of manual features has the best results in SemEval-2014 Task 4.

TD-LSTM [36]: The model uses a forward LSTM and a backward LSTM to model the left context with an aspect and right context with an aspect, and concatenates the last hidden states from both directions as sentiment features for sentiment classification.

ATAE-LSTM [41]: An LSTM based model which appends each word input vector with the aspect embedding to strengthen the effects of aspects in hidden states, it then uses attention mechanism to generate the final representation from the hidden states.

MemNet [37]: A deep memory model which employs multi-hop attention on the memory stacked by input word embeddings. It uses the attention at a previous hop to help calculate more accurate attention distribution at a later hop.

IAN [19]: An LSTM based model which uses attention mechanism to capture important information to generate representations of aspect and context separately by interactive learning.

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

Table 2: Main results. The results with "*" are retrieved from the papers of compared methods. "N/A" means this result is not available.

Model	Laptops	Restaurants	Twitter
SVM	70.49*	80.16*	N/A
TD-LSTM	67.55	77.58	70.80
ATAE-LSTM	69.27	78.50	69.88
IAN	72.10*	78.60*	N/A
MemNet	71.89	79.69	69.65
RAN	74.49*	80.23*	69.36*
BaseA	70.84	78.83	68.93
BaseB	72.25	79.46	69.36
BaseC	72.72	79.73	69.79
Cabasc	**75.07**	**80.89**	**71.53**

Table 3: Experiment 1 ans 2 show the results of the dataset divided into 10 or 5 subsets respectively. Ave acc and std dev present the average accuracy and standard deviations of 10 or 5 rounds prediction results. Time is the average time cost for one training iteration. There are three datasets used, laptop (Lap.), restaurant (Res.) and twitter (Twi.).

		Experiment 1			Experiment 2		
Model	Metrics	Lap.	Res.	Twi.	Lap.	Res.	Twi.
RAN	ave acc	65.20	70.06	60.92	67.67	72.37	63.69
	std dev	1.09e-2	6.38e-3	8.82e-3	4.25e-3	9.27e-3	7.04e-3
	time(s)	4.20	3.91	2.48	5.41	4.93	3.13
Cabasc	ave acc	67.41	71.84	62.47	69.42	73.49	64.27
	std dev	8.76e-3	4.58e-3	5.07e-3	5.46e-3	4.93e-3	5.98e-3
	time(s)	2.29	2.04	1.31	2.69	2.51	1.74

RAN [4]: A state-of-the-art model which adopts recurrent attention to capture sentiment features separated by a long distance on position-weighted memory. The memory is built on the hidden states of a deep bidirectional LSTM.

4.2 Main Results

Table 2 illustrates the experimental results. Our proposed model (Cabasc) achieves state-of-the-art performances on three datasets. From Table 2 we can make the following observations.

TD-LSTM performs poorly, implying that the method used by the author to model the preceding and following contexts as aspect-dependent features maybe incapable of capturing the interactions between aspects and contexts. In addition, the hidden state in the last time step contains information about the sequence with a strong focus on the parts close to the aspect word [2], so the sentiment features of words with a long distance may be forgotten.

Further, the LSTM based model ATAE-LSTM and IAN perform better than TD-LSTM on laptop and restaurant datasets. One main reason maybe the introduction of an attention mechanism that can make the models notice important parts of a sentence for a given aspect. Compared to ATAE-LSTM, IAN has better results because does not only models the context representation, but also models the representation of an aspect by using attention mechanism, which makes better use of aspect information than ATAE-LSTM. MemNet which does not apply the classical recurrent neural network, performs comparably with IAN. It proves the effectiveness of multi-hop attention mechanism. Moreover, MemNet is computationally efficient because its network structure does not have the complex operations as in LSTM. RAN achieves the best performances among the baselines. RAN does not only uses the multi-hop attention mechanism and deep bidirectional LSTM, but also uses position attention mechanism to provide tailor-made memories for different aspects in a sentence and non-linearly combines the results from each computational hops.

Among our proposed models, the Cabasc model obtains the highest classification accuracies on three datasets. The basic BaseA performs the poorly on the three datasets. This is not surprising because it does not consider the correlation between each context word and the given aspect in a sentence, consequently not all of the focused words are relative to the given aspect and may hide the characteristics of the keywords. Compared with BaseA, BaseB achieves 1.41%, 0.63%, 0.43% improvements on three datasets respectively. The improvements demonstrate the effects of sentence-level content attention mechanism which calculates the attention weights from a global perspective by considering the information of the full sentence, and embeds the entire sentence information into the output embedding vector. Now on the basis of BaseB, its extension BaseC outperforms it on the three datasets. As we expect, with a customized memory which considers the position information between aspect and context words, BaseC is able to better predict the sentiment polarity for a given aspect in sentence. It is worth noting that the performance of BaseB is not much worse than BaseC, which shows that the ability of position attention is limited. We argue that it is because the importance of a context word is not only dependent on the word order, but also on the information of context and aspect. Cabasc uses context attention mechanism and outperforms BaseC. The results confirm that the context attention mechanism is more effective than position attention mechanism. The context attention mechanism simultaneously takes into account the order of the words and their correlations to the given aspect in a sentence and benefits the model in generating customized memory in response to each given aspect. This helps to improve the classification accuracy of the model.

Cabasc outperforms RAN which is the recently state-of-the-art model on three datasets, thus showing the effectiveness of our model. In terms of model complexity, Cabasc is simpler and is easier to train than RAN because of there is no multi-hop structure.

Moreover, SVM outperforms TD-LSTM and has an advanced result on restaurant dataset, which demonstrates the importance of high quality features for aspect based sentiment classification.

4.3 Comparison of Cabasc and RAN in an approximate production environment

Aspect based sentiment analysis has been heavily studied recently because it is widely used in various domains, such as investigating the sentiment of consumer towards products from product reviews [33]. As we know, the labeled dataset used in the field of sentiment analysis in academic literatures usually has a larger training set than test set. But labeling the data is quite expensive and labor-intensive, and in production environment the amount of labeled training data is usually smaller than test data.

In order to verify the performance of our proposed Cabasc model and the recently state-of-the-art RAN model in real production

Table 4: Test accuracies of whether embedding the entire sentence into the output embedding vector. The "is_add" and "un_add" in parentheses of the results indicate whether the entire sentence is embedded.

Method	Laptops	Restaurants
BaseA+CAM (un_add)	72.57	80.17
BaseA+CAM (is_add)	73.51	80.44
Cabasc (un_add)	72.72	80.53
Cabasc (is_add)	75.07	80.89

environment, we design two experiments to simulate a practical situation. We argue that the experimental results may to some extent reflect their performances in the real production environment. The first one combines the training set and test set as a new dataset, then divides the dataset into 10 subsets where each round successively uses one of the subsets as training set and others as test set. The second one combines the training set and test set and divides the dataset into 5 subsets. The results are shown in Table 3. Since the source code of the RAN model is still not publicly available, we used our implementation to test.

Table 3 Experiment 1 gives the average accuracies, standard deviations of the prediction results of Cabasc and RAN in the first experiment, Table 3 Experiment 2 gives the results of the second experiment. We find that in both experiments our model has better average accuracies than the RAN model on all the datasets, and lower standard deviations on almost all the datasets. Meanwhile, the time cost of each training iteration about Cabasc is almost half of RAN, and the two experimental results also show that more training data could lead to better performance. Based on the above experimental results and the main results in subsection 4.2, , we advocate that the proposed model is more effective than RAN in aspect based sentiment classification. We argue that it may result from the higher complexity of RAN which makes it is hard to train with less training data, because it adopts multiple computational hops which increase the complexity. The results imply that Cabasc may be more suitable for practical application than RAN.

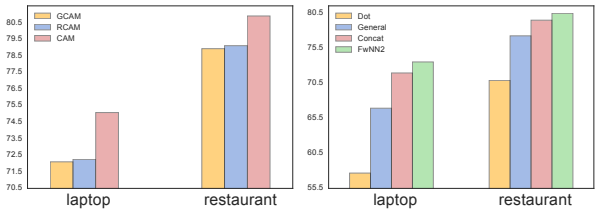
4.4 Effects of embedding the entire sentence into the output embedding vector

In this section, we design a series of contrast models to verify the validity of embedding the entire sentence into the output vector:

- **BaseA+CAM (un_add)**: Using the CAM in BaseA the same as Cabasc.
- **BaseA+CAM (is_add)**: Based on BaseA+CAM (un_add), the entire sentence is embedded into the output vector.
- **Cabasc (un_add)**: On the basis of Cabasc, but not embedding the entire sentence into the output embedding vector.
- **Cabasc (is_add)**: The Cabasc model proposed in the paper.

The results are shown in Table 4. We can see that models in which the entire sentence is embedded into the output embedding vector, have better performances than those without it. This proves that embedding the entire sentence into the output embedding vector can help judge the sentiment polarity of an aspect.

In order to verify the effect of embedding the entire sentence into the output embedding vector to handle complex sentence, we



(a) **(b)**

Figure 7: (a) Test accuracies of using different context attention mechanisms in Cabasc. (b) Test accuracies of using different content attention computational methods.

study contrary sentences as representatives of complex sentences. In order to facilitate research, we treat the sentences which have "but" or "however"[16] as contrary sentences. We count the contrary sentences in laptop test set, and the statistical result as 66, accounting for 10.34% in the test set. By analyzing the classification results of the above experiments, we find that the error rate of contrary sentences in BaseA+CAM (is_add) decreases from 54.54% to 40.90%, and in Cabasc (is_add) decreases from 53.03% to 37.87% compared with each corresponding un_add model, respectively. This indicates that embedding entire sentence into the output vector can help the model to better understand the complex sentence.

4.5 Effects of different Context Attention Mechanisms

We design and investigate the effects of different context attention mechanisms by replacing the used mechanism of CAM in Cabasc. Fig. 7 (a) shows the performances of the following mechanisms:

- **RCAM**: Making a modifier to CAM in Cabasc which runs the GRU_L from left to right and GRU_R from right to left to model the context information.
- **GCAM**: Making a modifier to CAM in Cabasc without using two GRUs to model the left and right context respectively. The module only uses one GRU over the entire sentence and feeds the hidden states into the MLP layer.
- **CAM**: Using the CAM described in subsection 3.4.

From Fig. 7 (a), we can see that RCAM outperforms GCAM. It proves the effect of considering the left and right context information respectively. RCAM performs worse than CAM because there is no interaction between aspect and context words in RCAM. CAM achieves the best results,we argue that it may be due to the fact that the CAM simultaneously takes into account the aspect information and the context information between aspect and context words, resulting in aspect-dependent context attention weights which contribute to the aspect based sentiment classification.

4.6 Comparison to other content attention computational methods in content attention module

Luong et al. [18] propose three kinds of attention computational methods in the field of machine translation and showed that the

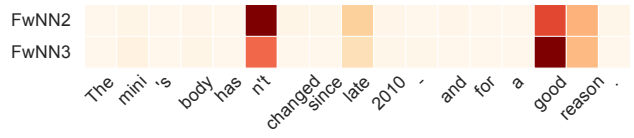


Figure 8: Content attention visualization. The color depth indicates the importance degree of a word. Attention weight in an attention vector is used as the color-coding.

General method performs the best. For investigating the effect of these methods in aspect based sentiment classification, we adopt the above three computational methods in the sentence-level content attention module from BaseA+CAM (is_add) which is designed in subsection 4.4 and compare the results with our proposed FwNN2:

$$\begin{cases} s(\mathbf{v}_a, \mathbf{m}_{wi}) = \mathbf{v}_a^T \mathbf{m}_{wi} & \text{Dot} \\ s(\mathbf{v}_a, \mathbf{m}_{wi}) = \mathbf{v}_a^T \mathbf{W}_{12} \mathbf{m}_{wi} & \text{General} \\ s(\mathbf{v}_a, \mathbf{m}_{wi}) = \mathbf{v}_1^T \tanh(\mathbf{W}_{13}[\mathbf{v}_a; \mathbf{m}_{wi}]) & \text{Concat} \end{cases} \quad (21)$$

Fig. 7 (b) shows the results. Dot and General compute the attention weight based on the similarity between the vectors of word and aspect. Concat and FwNN2 are similar because both use the feed forward neural network to compute the potential relatedness [2, 37] between the word and the sentiment polarity of the given aspect. From Fig. 7 (b), we see that Dot and General perform worse than Concat and FwNN2. It may because the potential correlation between words is more suitable than vector similarity in aspect based sentiment classification tasks. Fwnn2 outperforms Concat which may result from the increased parameters in FwNN2, which makes the method more powerful.

4.7 Case Study

We pick some examples and visualize the attention results to show what are noticed by the different attention mechanisms.

The FwNN2 and FwNN3 in Fig. 8 present the visualization results of the content attention distributions from BaseA and BaseB (un_add), where (un_add) means not embedding the entire sentence into the output embedding vector. The sentence in Fig. 8 is "The mini's body hasn't changed since late 2010- and for a good reason.", in which the corresponding aspect is "body". From Fig. 8, we find that multiple words "late", "good" and "n't" are focused by content attention module. However, the words "n't" and "late" are not related to the polarity of the given aspect of the sentence. As a result, the model makes incorrect prediction "negative", which may be because the wrongly focused words hide the sentiment features of "good". The sentence-level content attention mechanism is used in BaseB (un_add). The word "good" which is most relevant to sentiment polarity of "body" in this sentence makes a significant improvement on attention weight compared with that from BaseA, and the weight of the unrelated word "n't" is reduced. Accordingly, the model predicts the correct sentiment label "positive". This shows that the sentence-level attention mechanism can capture important parts of sentence more accurately.

Fig. 9 visualizes the attention distributions of position attention in BaseC and context attention in Cabasc. The example is "The two

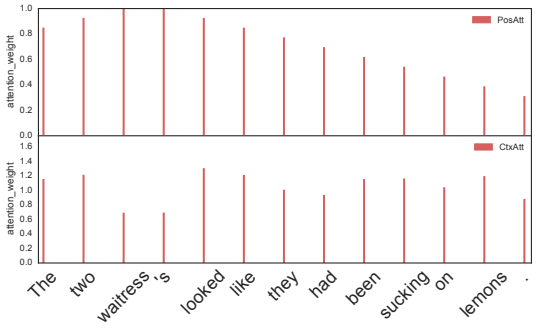


Figure 9: Position and context attention visualization.

waitress's looked like they had been sucking on lemons" and the aspect is "waitress's". Obviously in the position attention mechanism, larger distance between word and aspect, smaller position attention weight of the word. In this case, the keywords "sucking" and "lemon" which are important to the sentiment polarity of aspect but each has a so small weight that shows the position attention mechanism maybe rough and not flexible enough. Using context attention mechanism, the context attention weights of the two keywords are not monotonically decreasing by the increasing of relative distance from aspect, and become more reasonable. The results show that the context attention could be better than position attention to construct customized memory for a given aspect.

5 CONCLUSION

In this paper, we develop a content attention based aspect based sentiment classification model for the ABSC task. Two novel attention mechanisms, namely sentence-level content attention mechanism and context attention mechanism have been introduced to tackle the semantic-mismatch problem. The sentence-level content attention mechanism captures the important information about the given aspect from a global perspective by considering the information of entire sentence when calculating attention weights, and generates an output vector which embeds the entire sentence to help the model improve its ability of handling complex sentences. Context attention mechanism models the information of word order and the correlations between the words and the aspect into attention weights, the weights are used to generate a customized memory for each aspect. The proposed model is evaluated on three datasets, experimental results demonstrate the validity of the proposed attention mechanisms, and show that the proposed model achieves the state-of-the-art performance.

6 ACKNOWLEDGMENTS

We thank the anonymous reviewers for taking time to read and make valuable comments on this paper. This work was supported by NSFC under grant 61133016 and 61772117, the General Equipment Department Foundation (61403120102), and the Sichuan Hi-Tech industrialization program (2017GZ0308).

REFERENCES

[1] Yossi Adi, Einat Kermany, Yonatan Belinkov, Ofer Lavi, and Yoav Goldberg. 2017. Fine-grained analysis of sentence embeddings using auxiliary prediction tasks.

In *Proc. of the 5th ICLR*. CoRR, Toulon, France.
[2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proc. of the 3rd International Conference on Learning Representations*. CoRR, Scottsdale, USA.
[3] Yoshua Bengio, Aaron Courville, and Pascal Vincent. 2013. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35, 8 (2013), 1798–1828.
[4] Peng Chen, Zhongqian Sun, Lidong Bing, and Wei Yang. 2017. Recurrent attention network on memory for aspect sentiment analysis. In *Proc. of the 2017 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Copenhagen, Denmark, 463–472.
[5] Zhiyuan Chen, Arjun Mukherjee, and Bing Liu. 2014. Aspect extraction with automated prior knowledge learning. In *Proc. of the 52nd Annual Meeting of the Association for Computational Linguistics*. ACL, Baltimore, Maryland, 347–358.
[6] Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: encoder-decoder approaches. In *Proc. of 8th Workshop on Syntax, Semantics and Structure in Statistical Translation*. Association for Computational Linguistics, Doha, Qatar, 103–111.
[7] 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 *Proc. of the 52nd Annual Meeting of the Association for Computational Linguistics (Short Papers)*. ACL, Baltimore, Maryland, 49–54.
[8] Alex Graves, Greg Wayne, Malcolm Reynolds, Tim Harley, Ivo Danihelka, Agnieszka Grabska-Barwińska, Sergio Gómez Colmenarejo, Edward Grefenstette, Tiago Ramalho, John Agapiou, Adrià Puigdomènech Badia, Karl Moritz Hermann, Yori Zwols, Georg Ostrovski, Adam Cain, Helen King, Christopher Summerfield, Phil Blunsom, Koray Kavukcuoglu, and Demis Hassabis. 2016. Hybrid computing using a neural network with dynamic external memory. *Nature* 538, 7626 (2016), 471–476.
[9] Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems 28*. Curran Associates, Inc., Montréal, Canada, 1693–1701.
[10] Ozan Irsoy and Claire Cardie. 2014. Opinion mining with deep recurrent neural networks. In *Proc. of the 2014 Conference on Empirical Methods in Natural Language Processing*. ACL, Doha, Qatar, 720–728.
[11] Long Jiang, Mo Yu, Ming Zhou, Xiaohua Liu, and Tiejun Zhao. 2011. Target-dependent twitter sentiment classification. In *Proc. of the 49th Annual Meeting of the Association for Computational Linguistics*. ACL, Portland, Oregon, USA, 151–160.
[12] Yoon Kim, Carl Denton, Luong Hoang, and Alexander M Rush. 2017. Structured attention networks. In *Proc. of the 5th International Conference on Learning Representations*. CoRR, Toulon, France.
[13] Svetlana Kiritchenko, Xiaodan Zhu, Colin Cherry, and Saif M. Mohammad. 2014. NRC-Canada-2014: Detecting aspects and sentiment in customer reviews. In *Proc. of the 8th International Workshop on Semantic Evaluation*. Dublin, Ireland, 437–442.
[14] Ankit Kumar, Ozan Irsoy, Peter Ondruska, Mohit Iyyer, James Bradbury, Ishaan Gulrajani, Victor Zhong, Romain Paulus, and Richard Socher. 2016. Ask me anything: Dynamic memory networks for natural language processing. In *Proc. of The 33rd International Conference on Machine Learning*, Vol. 48. PMLR, New York, NY, USA, 1378–1387.
[15] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *Nature* 521, 7553 (2015), 436–444.
[16] Bing Liu. 2012. Sentiment analysis and opinion mining. *Synthesis Lectures on Human Language Technologies* 5, 1 (2012), 1–167.
[17] Yue Lu and Chengxiang Zhai. 2008. Opinion integration through semi-supervised topic modeling. In *Proc. of the 17th International Conference on World Wide Web*. ACM, Beijing, China, 121–130.
[18] Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proc. of the 2015 Conference on Empirical Methods in Natural Language Processing*. ACL, Lisbon, Portugal, 1412–1421.
[19] Dehong Ma, Sujian Li, Xiaodong Zhang, and Houfeng Wang. 2017. Interactive attention networks for aspect-level sentiment classification. In *Proc. of the Twenty-Sixth IJCAI*. Melbourne, Australia, 4068–4074.
[20] Qiaozhu Mei, Xu Ling, Matthew Wondra, Hang Su, and ChengXiang Zhai. 2007. Topic sentiment mixture: modeling facets and opinions in weblogs. In *Proc. of the 16th international conference on World Wide Web*. ACM, Banff, Alberta, Canada, 171–180.
[21] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26*. Curran Associates, Inc., Lake Tahoe, USA, 3111–3119.
[22] Samaneh Moghaddam and Martin Ester. 2013. The FLDA model for aspect-based opinion mining: addressing the cold start problem. In *Proc. of the 22nd International Conference on WWW*. ACM, Rio de Janeiro, Brazil, 909–918.
[23] Thien Hai Nguyen and Kiyoaki Shirai. 2015. PhraseRNN: Phrase recursive neural network for aspect-based sentiment analysis. In *Proc. of the 2015 Conference on Empirical Methods in Natural Language Processing*. ACL, Lisbon, Portugal, 2509–2514.
[24] Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proc. of the 2014 Conference on Empirical Methods in Natural Language Processing*. ACL, Doha, Qatar, 1532–1543.
[25] Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. 2014. Semeval-2014 task 4: Aspect based sentiment analysis. In *Proc. of the 8th International Workshop on Semantic Evaluation*. Dublin, Ireland, 27–35.
[26] Soujanya Poria, Erik Cambria, and Alexander Gelbukh. 2016. Aspect extraction for opinion mining with a deep convolutional neural network. *Knowledge-Based Systems* 108, Supplement C (2016), 42–49.
[27] Guang Qiu, Bing Liu, Jiajun Bu, and Chun Chen. 2011. Opinion word expansion and target extraction through double propagation. *Computational Linguistics* 37, 1 (2011), 9–27.
[28] Tim Rocktäschel, Edward Grefenstette, Karl Moritz Hermann, Tomáš Kočiský, and Phil Blunsom. 2016. Reasoning about entailment with neural attention. In *Proc. of the 4th ICLR*. San Juan, Puerto Rico.
[29] Sebastian Ruder, Parsa Ghaffari, and John G Breslin. 2016. A hierarchical model of reviews for aspect-based sentiment analysis. In *Proc. of the 2016 Conference on Empirical Methods in Natural Language Processing*. ACL, Austin, Texas, USA, 999–1005.
[30] Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for sentence summarization. In *Proc. of the 2015 Conference on EMNLP*. ACL, Lisbon, Portugal, 379–389.
[31] Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2017. Bi-directional attention flow for machine comprehension. In *Proc. of the 5th International Conference on Learning Representations*. CoRR, Toulon, France.
[32] Richard Socher, Danqi Chen, Christopher D Manning, and Andrew Y. Ng. 2013. Reasoning with neural tensor networks for knowledge base completion. In *NIPS 2013*. Curran Associates, Inc., Lake Tahoe, USA, 926–934.
[33] Kaisong Song, Ling Chen, Wei Gao, Shi Feng, Daling Wang, and Chengqi Zhang. 2016. PerSentiment: A personalized sentiment classification system for microblog users. In *Proc. of the 25th International Conference Companion on World Wide Web*. International World Wide Web Conferences Steering Committee, Montréal, Canada, 255–258.
[34] Sainbayar Sukhbaatar, Arthur Szlam, Jason Weston, and Rob Fergus. 2015. End-to-end memory networks. In *Advances in Neural Information Processing Systems 28*. Curran Associates, Inc., Montréal, Canada, 2440–2448.
[35] 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. In *Proc. of the Twenty-Fourth International Joint Conference on Artificial Intelligence*. Buenos Aires, Argentina, 1333–1339.
[36] Duyu Tang, Bing Qin, Xiaocheng Feng, and Ting Liu. 2016. Effective LSTMs for target-dependent sentiment classification. In *Proc. of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*. Osaka, Japan, 3298–3307.
[37] Duyu Tang, Bing Qin, and Ting Liu. 2016. Aspect level sentiment classification with deep memory network. In *Proc. of the 2016 Conference on Empirical Methods in Natural Language Processing*. ACL, Austin, Texas, USA, 214–224.
[38] Duyu Tang, Furu Wei, Nan Yang, Zhou Ming, Ting Liu, and Bing Qin. 2014. Learning sentiment-specific word embedding for twitter sentiment classification. In *Proc. of the 52nd Annual Meeting of the Association for Computational Linguistics*. ACL, Baltimore, Maryland, 1555–1565.
[39] Duy-Tin Vo and Yue Zhang. 2015. Target-dependent twitter sentiment classification with rich automatic features. In *Proc. of the 24th International Joint Conference on Artificial Intelligence*. Buenos Aires, Argentina, 1347–1353.
[40] Shuai Wang, Zhiyuan Chen, and Bing Liu. 2016. Mining aspect-specific opinion using a holistic lifelong topic model. In *Proc. of the 25th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, Montréal, Canada, 167–176.
[41] Yequan Wang, Minlie Huang, Xiaoyan Zhu, and Li Zhao. 2016. Attention-based LSTM for aspect-level sentiment classification. In *Proc. of the 2016 Conference on Empirical Methods in Natural Language Processing*. ACL, Austin, Texas, USA, 606–615.
[42] Jason Weston, Sumit Chopra, and Antoine Bordes. 2015. Memory networks. In *Proc. of the 3rd International Conference on Learning Representations*. CoRR, Scottsdale, Arizona, USA.
[43] Meishan Zhang, Yue Zhang, and Duy-Tin Vo. 2016. Gated neural networks for targeted sentiment analysis. In *Proc. of the Thirtieth AAAI*. AAAI Press, Phoenix, Arizona, USA, 3087–3093.