Information Processing and Management 56 (2019) 463–478

Contents lists available at ScienceDirect

Information Processing and Management

journal homepage: www.elsevier.com/locate/infoproman

Aspect-based sentiment analysis with alternating coattention networks

Chao Yang[*,a], Hefeng Zhang[a], Bin Jiang[a], Keqin Li[a,b]

[a] College of Computer Science and Electronic Engineering, Hunan University, Lushan Road (S), Yuelu District, Changsha, China
[b] Department of Computer Science, State University of New York, New Paltz, NY 12561, USA

ARTICLE INFO

Keywords:
Aspect-based sentiment analysis
Coattention
LSTM
Memory network
Location weighted function

2010 MSC:
00-01
99-00

ABSTRACT

Aspect-based sentiment analysis aims to predict the sentiment polarities of specific targets in a given text. Recent researches show great interest in modeling the target and context with attention network to obtain more effective feature representation for sentiment classification task. However, the use of an average vector of target for computing the attention score for context is unfair. Besides, the interaction mechanism is simple thus need to be further improved. To solve the above problems, this paper first proposes a coattention mechanism which models both target-level and context-level attention alternatively so as to focus on those key words of targets to learn more effective context representation. On this basis, we implement a Coattention-LSTM network which learns nonlinear representations of context and target simultaneously and can extracts more effective sentiment feature from coattention mechanism. Further, a Coattention-MemNet network which adopts a multiple-hops coattention mechanism is proposed to improve the sentiment classification result. Finally, we propose a new location weighted function which considers the location information to enhance the performance of coattention mechanism. Extensive experiments on two public datasets demonstrate the effectiveness of all proposed methods, and our findings in the experiments provide new insight for future developments of using attention mechanism and deep neural network for aspect-based sentiment analysis.

1. Introduction

In recent years, sentiment analysis has become one of the most active research fields in Natural Language Processing (NLP) (Liu, 2012). It also has extensive applications in data mining, information retrieval, question answering, summarization, intelligent recommender system and so on (Bo & Lillian, 2008). The rapid development of this field is due to various social media applications on the Internet, such as product reviews, forum discussions, micro-blog and WeChat, etc. Specially, in e-commerce systems, large amounts of review texts which reflect customers' positive or negative feelings are posted towards different aspects of products and services they received, for example, design, quality, price, logistics and so on. Sentiment analysis is therefore proposed as a technique which can help detect users' attitudes and predict their demands, so as to promote users' further browsing and consuming behaviors. It is an important tool to help users to focus on useful information and relieve the information overload in this big data era. However, the traditional sentiment analysis is always a sentence-level or document-level task which aims at finding the overall sentiment associated with one entity such as the overall topic of the review. If we intent to find the sentiment for the aspects of the entity, the general sentiment analysis cannot satisfy this task. Here, the aspect can be any characteristic or property of one specific entity. For

* Corresponding author.
E-mail address: yangchaoedu@hnu.edu.cn (C. Yang).

https://doi.org/10.1016/j.ipm.2018.12.004
Received 29 July 2018; Received in revised form 2 November 2018; Accepted 3 December 2018
0306-4573/

example, in product reviews, the product itself is usually the entity, and all things related to the product (e.g., price, quality, etc.) are aspects of it (Schouten & Frasincar, 2016). Therefore, aspect-based sentiment analysis (ABSA) is raised for this task (Li, Zhou, & Li, 2015; Pontiki et al., 2014; Tubishat, Idris, & Abushariah, 2018).

ABSA aims at inferring the sentiment polarity (e.g. positive, negative, neutral) of a sentence expressed toward a target which is the aspect of one specific entity. For example, given a sentence "The fish is fresh but the variety of fish is nothing out of ordinary.", the polarity of the target "fish" is positive while the polarity of the target "variety of fish" is negative. The ABSA task involves two subtasks: aspect detection and sentiment classification. The ABSA task in this paper assumes that the aspects are known, so we only focus on the sentiment classification task. ABSA is a fundamental task in natural language processing and catches many research attentions. Early works generally use machine learning algorithms to tackle this task. That is, a set of features are firstly extracted to describe the relationship between context and target, and then a sentiment classifier, such as SVM, is trained from sentences with polarity labels based on these manual features (Kiritchenko, Zhu, Cherry, & Mohammad, 2014; Wagner et al., 2014; Zhang & Lan, 2015). Recently, neural networks have become effective solutions for sentiment analysis because of their efficiency of automatic feature extraction. The accuracy of sentiment analysis based on neural networks such as CNN (Kalchbrenner, Grefenstette, & Blunsom, 2014; Kim, 2014; Zhang, Zhao, & LeCun, 2015), LSTM (Qian, Huang, Lei, & Zhu, 2017; Ruder, Ghaffari, & Breslin, 2016; Zhou et al., 2016), Tree-LSTM (Tai, Socher, & Manning, 2015; Zhu, Sobhani, & Guo, 2015) and attention networks (Lin et al., 2017; Yang et al., 2016) has reached or surpassed the methods depending on the manual features. However, the ABSA task is different from general sentiment classification. In ABSA task, multiple targets could appear in one sentence, and each target has related words to modify it. When judging the sentiment of current target, other targets and related words would become noises. Therefore, we need to fully consider the relationship between target and context words when we design neural networks. Meanwhile, using attention mechanism to learn context feature associated with the target for sentiment analysis has also shown its effectiveness in ABSA task (Chen, Sun, Bing, & Yang, 2017; Liu & Zhang, 2017; Ma, Li, Zhang, & Wang, 2017; Tang, Qin, & Liu, 2016b; Tay, Tuan, & Hui, 2018; Wang, Huang, Zhao, & Zhu, 2016).

When computing the relationship between the target and the context, most existing methods based on attention mechanism obtain the target representation by an average pooling method. However, if the target contains multiple words, for example, "glass of wine/MS Office 2011 for Mac", the average function would introduce noise words such as "of/for". In addition, the instances in the target have an imbalance effect on the classification. In the above examples, "wine" is more important than "glass", and "Office" is more important than "MS", "2011" and "Mac". If we can use the essential information of the target, like "wine/Office", to compute the attention score for context, the quality of context features will be improved. Up to now, existing models cannot effectively solve this problem. The IAN (Ma et al., 2017) interactively learns attentions in contexts and targets but it adopts a parallel interaction mechanism, which makes inefficient use of target attention representation to further extract related feature from contexts. Similar to IAN, we argue that the representation learning of target and context is necessary and helpful to each other. We therefore propose an alternative coattention mechanism to resolve this problem by considering both the target-level attention and context-level attention. It learns the attentions in target and context sequentially that it first generates the attention representation for target based on the context and then generates the attention representation for context based on the learned target representation. We believe that the coattention mechanism is a more intuitive and effective interaction mechanism for ABSA task. We then propose the Coattention-LSTM which first learns the nonlinear representation for target and context, and then extracts more effective sentiment features from context based on the interactive coattention learning process, so as to achieve better performance.

In addition, for the reason that extracting sentiment feature related with certain targets is hard for the single layer attention networks, some works (Chen et al., 2017; Tang et al., 2016b) apply the memory networks to iterate the attention computation unit multiple times in order to learn multiple levels of abstraction. In the memory networks, each hop layer is to seek the evidences related with target representation vector from context utilizing conventional attention function. However, these works based on memory network have the same drawback that they use an average method to process targets. Therefore, we use the coattention mechanism as the basic computation unit to design an interaction memory network named Coattention-MemNet which can learn more effective features of target and context alternately and the model can also learn more abstract features through the iteration mechanism based on memory network.

Some works (Chen et al., 2017; Li, Bing, Lam, & Shi, 2018; Schouten & Frasincar, 2018; Tang et al., 2016b) have shown that the location information is useful in ABSA task and can help the model to learn better text features for classification. In order to take the location information into consideration, we propose a location weighted function for coattention mechanism. Similar to other methods, it is designed based on the relative distance between context words and target, but it adds the location weights into attention layer instead of decoded features to directly limit the attention weights to focus on sentiment features around target. Finally, we apply the location weighted function to both Coattention-LSTM and Coattention-MemNet models.

This article has the following research contributions:

- To reduce the effect of noise words of targets such as preposition and fully utilize the key words of targets to learn context representation, we replace the conventional attention with a new alternating coattention mechanism which models both the target-level and context-level attention in a more intuitive and effective interaction way for ABSA tasks.
- To verify the effectiveness of coattention mechanism, we propose the Coattention-LSTM. It learns the nonlinear representations of context and target simultaneously and can extract more effective sentiment features from coattention mechanism.
- In order to overcome the drawback of deep memory network applied in ABSA task which explores key context clues with an average target vector, we propose an interaction memory network based on the coattention mechanism named Coattention-MemNet to learn the key features from the target and context alternately with an iteration mechanism.

- In order to take the location information into consideration, we propose a location weighted function for coattention mechanism and apply the function to both Coattention-LSTM and Coattention-MemNet.

Finally, we evaluate our proposed methods on two public datasets: SemEval 2014 datasets and Twitter dataset. The results show that our proposed methods achieve the above research contributions.

The rest of the paper is organized as follows. After introducing related works in Section 2, we elaborate our proposed methods in Section 3. We then perform experimental evaluation in Section 4. Finally, we conclude the whole paper and give an outlook of future work in Section 5.

2. Literature review

This section will be divided into three parts. The first part introduces the works on ABSA task, which mainly use the LSTM (Hochreiter & Schmidhuber, 1997) as encoder. The second part introduces the effectiveness of memory networks (Weston, Chopra, & Bordes, 2015) in natural language processing and their applications in ABSA tasks. The final part introduces the works with coattention networks.

2.1. Aspect-based sentiment analysis based on LSTM

ABSA aims at inferring the sentiment polarity of a sentence expressed toward a target which is one aspect of a specific entity. Different from the concept "aspect" of search result diversification task which denotes the multiple possible intents, interpretations, or subtopics associated with a given query (Dang & Croft, 2012; Liang, Ren, & de Rijke, 2014; Liang, Yilmaz, Shen, Rijke, & Croft, 2017) in information retrieval field, the aspect here is exactly the characteristic or property of an entity in the review and multiple aspects can appear in one sentence. Therefore, the main challenge of ABSA task is how to effectively model the relationship between target and context. Jiang, Yu, Zhou, Liu, and Zhao (2011) pointed out that 40% of classification errors in this task were due to the ignorance of the target information. In order to overcome the difficulty, lots of works have been proposed. Early works mainly use machine learning algorithms (Kiritchenko et al., 2014; Wagner et al., 2014; Zhang & Lan, 2015) which extract a set of features to describe the relationship between target and context, for example, context-target bigrams and parse path. However, in addition to the difficulties for extracting effective features, some effective features such as parse features are heavily dependent on the performance of parsing tools, and complex features such as parse path have a low coverage.

Neural networks, especially the LSTM network, can encode sentences without feature engineering and have been applied in many natural language processing tasks (Sundermeyer, Schlüter, & Ney, 2012; Sutskever, Vinyals, & Le, 2014; Tang, Qin, & Liu, 2015). Tang, Qin, Feng, and Liu (2016a) proposed TD-LSTM and TC-LSTM which developed two dependent LSTMs to model the left and right contexts divided by target, where target information was also added into input via word embeddings. Similarly, Zhang, Zhang, and Vo (2016) utilized the Gated Neural Networks to control the importance of left and right context to target. The difference between these two methods and those using LSTM directly is that one of the divided sentences could only contain one target with a high probability. However, the context is hard to be reasonably divided by target because of the flexible expression and the above two methods do not effectively capture the underlying interaction between the target and the context.

Since the success of applying attention network on translation task (Bahdanau, Cho, & Bengio, 2015; Luong, Pham, & Manning, 2015), lots of works make use of attention mechanism to model the relationship between target and context. Wang et al. (2016) proposed AE-LSTM and AETE-LSTM which adopted the attention mechanism to focus on the related words of context when given different targets as input. Tay et al. (2018) proposed AF-LSTM which learned to attend based on associative relationships between context words and target. It thus allowed the model to adaptively focus on the correct context words when given a target. Although the use of attention mechanism has improved the performance of ABSA tasks, all these works simply process the target through the average pooling method when computing the attention score for context. If a target contains multiple words, the performance of attention would decrease. Ma et al. (2017) proposed IAN to learn attention representations for context and target based on two attention networks in parallel and then concatenated them for sentiment classification. Although IAN was a significant work which considered the interactive learning of context and target, it still used average vectors to compute attention score for both target and context. Furthermore, the attention representations learned for target and context were directly concatenated as the final representation. The interaction learning between context and target was too simple and the target attention representation has not been used efficiently. Ma, Peng, and Cambria (2018b) proposed a hierarchical attention model for aspect-based sentiment analysis task, which included both the target-level attention and sentence-level attention. However, the target-level attention was a self-attention network which took nothing but the hidden output itself as input. Without the guidance of context, the target-level attention was difficult to learn. Following the core idea of IAN, we think that the context information will help the learning of target-level attention. As such, our model could ameliorate the weaknesses of IAN and the hierarchical attention model with an alternative coattention mechanism.

In addition, there still exists some works that use knowledge based methods to tackle the ABSA task. Schouten and Frasincar (2018) and Schouten, Frasincar, and de Jong (2017) proposed rule-based ontology methods which designed ontologies by utilizing common domain knowledge to help improve the ABSA result. Such ontology-enhanced aspect-based sentiment analysis method could explicitly solve the sentiment polysemy, for example, small price is predicted as positive, while small serving is predicted as negative. Ma et al. (2018b) embedded the affective commonsense knowledge into a hierarchical LSTM attention model to help the LSTM filter out the irrelevant information. It is a good idea to introduce external knowledge to guide the model to learn relevant rules so that the model can learn more effective features. However, the knowledge based methods are very dependent on the knowledge base which is hard to establish and the knowledge rule is also difficult to design into neural networks effectively.

2.2. Aspect-based sentiment analysis based on memory network

Memory network by Weston et al. (2015) is a general machine learning framework. Sukhbaatar, Szlam, Weston, and Fergus (2015) improved it and proposed a memory network that can be trained in an end to end way. Until now, memory network has been successfully applied in question answering, language model, dialogue system and mention recommendation tasks (Bordes, Boureau, & Weston, 2017; Huang, Zhang, & Huang, 2017; Kumar et al., 2016; Sukhbaatar et al., 2015). The efficiency of memory network lies on its multiple computational layers (which is termed "hops") structure. Memory network encodes the inputs into vectors and stores them in the memory. The query is also embedded to the vector as the initial query representation vector. The function of each hop layer is to seek the evidences related to query representation vector from memory. The computation process of each layer can be divided into two steps. 1) Update the query representation. The query representation of each hop is different. The current layer of query representation is updated with the sum of initial query representation and relevant evidences from the previous layers. 2) Update relevant evidences based on input memory. The attention mechanism is used as computation unit to update the evidences related to the current layer's query representation. This process will be repeated multiple times until more abstractive evidences could be obtained based on previously extracted evidences.

Inspired by these works, Tang et al. (2016b) first applied memory network to ABSA task. The model used the embeddings of pretrained context words as memory and regarded the attention with an average target representation as the computation unit which will be repeated multiple hops to seek the evidences related to target from context. Chen et al. (2017) built the memory using a two layer Bi-LSTM network and used a gate mechanism of GRU in the connection between hop layers to control the history memory. However, these works use the conventional attention as the computation unit and ignore the importance of targets modeling. It means that the existing memory networks could only seek the evidences related to the target (or query) from context memory and could not interactively seek the evidences from context and target (or query). In this paper, we design an interaction memory network which could alternately learn the evidences from both the target and context. Yin, Song, and Zhang (2017) proposed an iterative attention module which attends and reads memories of questions and documents alternatively with a multiple-hops mechanism, in order to generate aspect-specific sentence-level and document-level representations for the document-level aspect-based sentiment classification task. However, learning aspect representation needs to select seed words for each aspect. For example, the words "value", "price", "worth", "cost", and "$" are selected as seeds for aspect "Price". While how to decide the selection of seed words is a meta problem. Besides, in Yin's model, the memory vectors update function adopts a simple strategy without considering any historical information. Different from iterative attention module, our model is proposed for the sentence-level aspect-based sentiment classification task and we do not need to select any seed word for each aspect. The procedures of iterative attention are also different from Yin's work. That is, we update the memory vector of context according to the aspect memory vector and historical context memory vector.

2.3. Coattention networks

Many researches have proved the effectiveness of coattention mechanism in several fields. Lu, Yang, Batra, and Parikh (2016) first proposed the coattention mechanism for VQA that jointly performed question-guided visual attention and image-guided question attention. The coattention mechanism includes two strategies that parallel coattention mechanism attends the image and question simultaneously, and alternating coattention mechanism sequentially alternate between generating image and question attention. The results on the VQA dataset proved that the performance of alternating coattention is better than parallel coattention. Zhang, Wang, Huang, Huang, and Gong (2017) applied the alternating coattention mechanism in hashtag recommendation for multimodal microblog. Moreover, Ma, Zhang, Wang, Cui, and Huang (2018a) applied the alternating coattention in mention recommendation for multimodal microblog and proposed the cross-attention memory network. The motivation of all these works is that the attentions for text and image are important for the task and there exists some relationship between the textual and visual information since they are helpful and associated with each other.

Xiong, Zhong, and Socher (2017) introduced the Dynamic Coattention Network (DCN) for machine comprehension. Their work was also based on the coattention mechanism that the DCN first fused co-dependent representations of the question and the document in order to focus on relevant parts of both. Then a dynamic pointing decoder iterated over potential answer spans which enabled the model to recover from initial local maxima corresponding to incorrect answers. This research was the first work that introduced the coattention mechanism to machine comprehension, but was monomodal and only involves text information. To some extent, machine comprehension is similar to the ABSA task in that the context and target (or question) information are related and helpful to each other. Similar to DCN, we first propose an alternating coattention mechanism to extract the co-dependent representations of the target and the context. Different from the DCN which adopts a dynamic iteration mechanism based on LSTM for machine comprehension, we propose a coattention iteration mechanism to extract high level co-dependent representations cyclically based on memory networks in ABSA task.

3. Methodology

In this section, we first define the research task and depict an overview of coattention networks. Then, we introduce a new interactive network and a memory network which could alternately learn the evidences from the target and context named

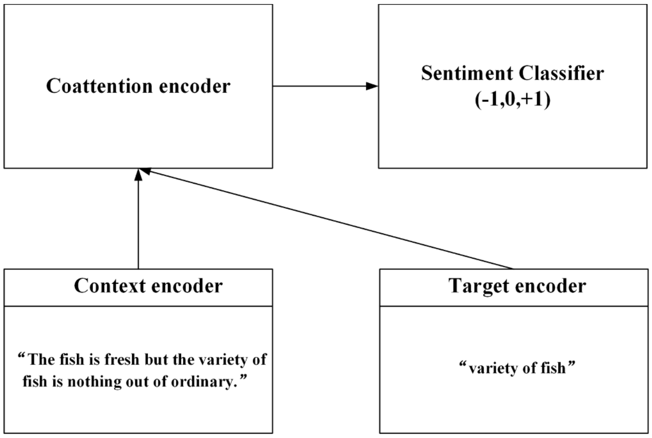

Fig. 1. Overview of the Coattention networks.

Coattention-LSTM and Coattention-MemNet, respectively. We also design a location enhanced method to further improve the sentiment classification result.

3.1. Task definition and notation

Given a sentence $s = [w_c^1, w_c^2, w_c^3, \ldots, w_c^n]$ consisting of n words and a target $t = [w_t^1, w_t^2, w_t^3, \ldots, w_t^m]$ consisting of m words occurring in sentence s, the ABSA task aims at analyzing the sentiment of sentence s towards the target t. For example, the sentiment polarity of sentence "The fish is fresh but the variety of fish is nothing out of ordinary." towards "fish" is neutral, while the polarity towards "variety of fish" is negative.

3.2. An overview of coattention networks

Fig. 1illustrates an overview of the coattention networks for ABSA task, which includes one context encoder, one target encoder, one coattention encoder and a sentiment classifier. The context and target encoders are used to encode the words of context and target into vectors which capture the features of each word. The coattention encoder learns the representation which captures the interactions between context and target words based on the coattention mechanism. Finally, the sentiment classifier predicts the sentiment polarity of the context towards the target.

In this paper, we propose two models with coattention mechanism that based on LSTM network and memory network respectively, namely Coattention-LSTM and Coattention-MemNet. In the following, we will introduce these two models in details.

3.3. Coattention-LSTM

The design and implementation of the Coattention-LSTM is shown in Fig. 2. In the figure, P means an average pooling function and C_{init} is the pooling result of context hidden states. A means the attention function, T_r is the attention results of target hidden states and C_r is the attention results of context hidden states. The numbers denote the order of these computation units. The components of Coattention-LSTM are introduced as follows.

3.3.1. Context and target encoder

We suppose that a context consists of n words $[w_c^1, w_c^2, w_c^3, \ldots, w_c^n]$ and a target has m words $[w_t^1, w_t^2, w_t^3, \ldots, w_t^m]$. We map the context and target words into two embedding matrices: $[e_c^1, e_c^2, e_c^3, \ldots, e_c^n]$ and $[e_t^1, e_t^2, e_t^3, \ldots, e_t^m]$. Each e^i is a word vector from a pre-trained word matrix $M^{v \times d}$. v denotes the number of words involved in corpus and d denotes the dimension of the vectors.

We use one LSTM to encode the context and get the hidden states for each word: $h_c^i = LSTM(h_c^{i-1}, e_c^i)$, and then we define the context encoding matrix as $C = [h_c^1, h_c^2, h_c^3, \ldots, h_c^n]$. Similarly, we use another LSTM to encode the target and get the hidden states for each word: $h_t^i = LSTM(h_t^{i-1}, e_t^i)$, and then we define the target encoding matrix as $T = [h_t^1, h_t^2, h_t^3, \ldots, h_t^m]$.

3.3.2. Coattention encoder

We propose a coattention mechanism as the basic of coattention encoder. The coattention mechanism is designed to sequentially alternate between target-level and context-level attention. More concretely, the process of alternating coattention is executed in three steps: 1) summarize the context into a single vector C_{init}; 2) attend to the target based on the context summary vector C_{init}; 3) attend to the context based on the attended target features T_r.

We can first get the initial representation of context based on the context encoding matrix C:

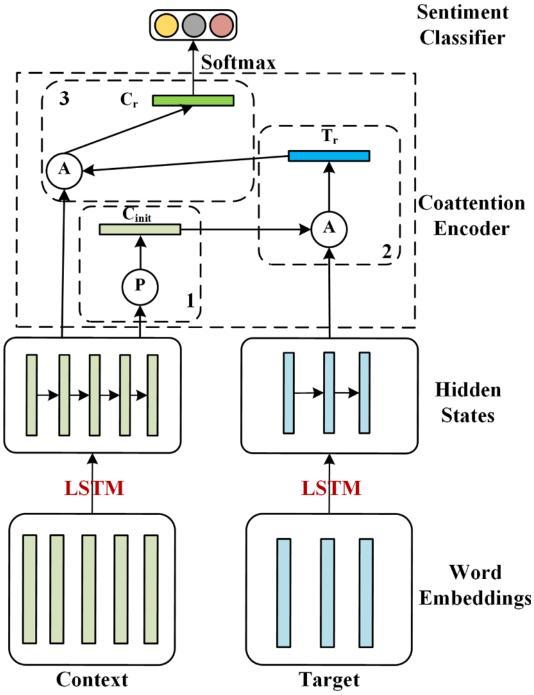

Fig. 2. Coattention-LSTM.

$$C_{init} = \sum_{i=1}^{n} h_c^i / n \tag{1}$$

In order to describe the attending process, we first define a score function to describe the relationship between each target word and the initial context representation. The score also denotes the importance of each word in target for context:

$$f(h_t^i, C_{init}) = \tanh(h_t^i \cdot W_t \cdot C_{init} + b_t) \tag{2}$$

where W_t and b_t are weight matrix and bias respectively.

Then we get the normalized attention score for target through a softmax function:

$$\alpha_i = \frac{\exp(f(h_t^i, C_{init}))}{\sum_{j=1}^{m} \exp(f(h_t^j, C_{init}))} \tag{3}$$

Afterwards, we can get the target attention representation T_r through a weighted combination of target hidden states:

$$T_r = \sum_{i=1}^{m} \alpha_i \cdot h_t^i \tag{4}$$

Similarly, we compute the attention score for context based on the target attention representation T_r:

$$\beta_i = \frac{\exp(f(h_c^i, T_r))}{\sum_{j=1}^{n} \exp(f(h_c^j, T_r))} \tag{5}$$

Finally, we obtain the context attention representation C_r through a weighted combination of context hidden states:

$$C_r = \sum_{i=1}^{n} \beta_i \cdot h_c^i \tag{6}$$

We use C_r as the output of Coattention encoder for sentiment classification.

3.3.3. Sentiment classifier

Similar to other models (Chen et al., 2017; Ma et al., 2017; Tang et al., 2016b; Tay et al., 2018; Wang et al., 2016), we use a softmax layer as the sentiment classifier.

First, we use a linear function mapping the output of coattention representation C_r to a l-dimensional space, where l stands for the number of sentiment polarities:

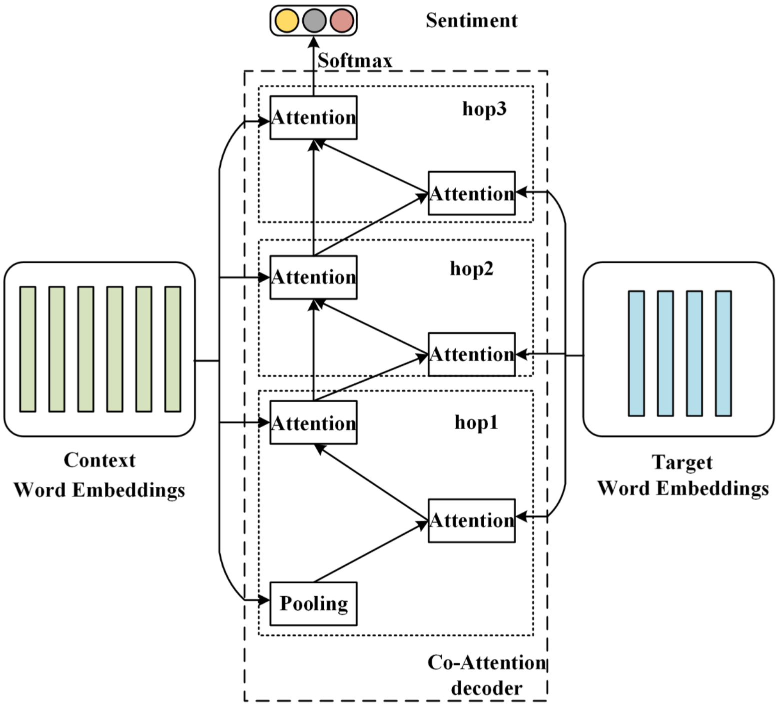

Fig. 3. Coattention-MemNet.

$$x = W_x \cdot C_r + b_x \tag{7}$$

where W_x and b_x are weight matrix and bias respectively.

Then, we use a softmax function to compute the probability of labeling sentence belonging to sentiment polarity i.

$$p_i = \frac{\exp(x_i)}{\sum_{j=1}^{l} \exp(x_j)} \tag{8}$$

The loss is computed with the cross entropy function.

$$L = -\sum_{i=1}^{l} y_i \log(p_i) \tag{9}$$

3.4. Coattention-MemNet

As shown in Fig. 3, similar to deep memory network used by Tang et al. (2016b), we use the embedding layer as the context and target encoder, and the encoding result of context is stacked as the external memory. The difference is that the target is obtained through the attention function instead of an average method. We use the coattention mechanism to replace conventional attention so that the Coattention-MemNet can learn important features from the target and context alternately. The components of Coattention-MemNet are described below.

3.4.1. Context and target encoder

We use the embedding layer which first maps n words of context to a context encoding matrix $C = [e_c^1, e_c^2, e_c^3, ..., e_c^n]$, and then maps m words of target to a target encoding matrix $T = [e_t^1, e_t^2, e_t^3, ..., e_t^m]$. Each e_i is a word vector from a pre-trained word matrix $M^{v \times d}$. v denotes the number of words involved in corpus and d denotes the dimension of vector.

3.4.2. Coattention encoder

The coattention mechanism in Coattention-MemNet is similar to Coattention-LSTM, which contains target-level attention and context-level attention. The process of each hop layer is executed in three steps: 1) to obtain the context vector of current hop layer; 2) to attend the target memory based on the current context vector and update target representation of current layer; 3) to attend to the context memory and use the attended context representation as the context vector representation of next hop layer. As shown in Fig. 3, the computation process of hop 1 layer is the same as Coattention-LSTM. After hop 1 layer, current context vector is the context representation outputted by the last layer and the context attention representation is computed based on the context memory, the updated target features and the current context features updated with history context information.

To explain the coattention encoder more concretely, we define C_{hop} and T_{hop} as the context vector representation and the target attention representation of current hop layer respectively. We define C_{hop_att} as the context attention representation computed based on C_{hop} and T_{hop}.

Specifically, we first define a score function to describe the relationship between each target word and the context vector representation. Here, we adopt the additive attention because of its good performance in high dimension:

$$f(e_t^i, C_{hop}) = \tanh(V_t[e_t^i; C_{hop}] + b_t) \tag{10}$$

where V_t and b_t are weight and bias respectively, *tanh* is a non-linear function.

Then we get the normalized attention score for target through a softmax function.

$$\alpha_i = \frac{\exp(f(e_t^i, C_{hop}))}{\sum_{j=1}^{m} \exp(f(e_t^j, C_{hop}))} \tag{11}$$

Afterwards, we can get the target attention representation T_{hop} of current hop layer through a weighted combination of target word embeddings.

$$T_{hop} = \sum_{i=1}^{m} \alpha_i \cdot e_t^i \tag{12}$$

Similarly, we can compute the attention score for context based on the target attention representation T_{hop}.

$$\beta_i = \frac{\exp(f(e_c^i, T_{hop}))}{\sum_{j=1}^{n} \exp(f(e_c^j, T_{hop}))} \tag{13}$$

Then, we can get the context attention representation of current hop layer through a weighted combination of context word embeddings, and it is used as the context vector representation of next hop layer C_{hop+1}:

$$C_{hop+1} = C_{hop\_att} = \sum_{i=1}^{n} \beta_i \cdot e_c^i \tag{14}$$

As shown in Fig. 3, in the hop 1 layer, the average pooling result of the context embeddings is used to initialize the current context representation. After computing the target attention representation and context attention representation, the final context attention result of current layer is used as the context vector representation of next layer. This means that the context evidences related to target will be passed into the next layer. After hop 1 layer, the current context representation is first used to obtain target attention representation of current layer and then combined with calculated target attention representation to compute attention score for context.

Therefore, the target attention representation after hop 1 layer is computed in the same way as in hop 1 layer. The attention score for context after hop 1 layer is computed as:

$$\beta_i = \frac{\exp(f(e_c^i, T_{hop} + C_{hop}))}{\sum_{j=1}^{n} \exp(f(e_c^j, T_{hop} + C_{hop}))} \tag{15}$$

Though the hop layer will be repeated multiple times, the parameters of each hop layer are shared.

Finally, we use the context attention representation $C_{lasthop\_att}$ obtained from the last hop layer as the coattention representation for classification.

3.4.3. Sentiment classifier

Similar to Coattention-LSTM, we use a softmax layer as the classifier and the cross entropy function as the loss function.

3.5. Location-enhanced coattention

The coattention process introduced above includes two steps. It first obtains the important features of targets through the target-level attention with the average context. Then, it extracts the key sentiment features relevant to the attended target information based on the context-level attention. There exists one defect in target level attention and context level attention respectively. In the target-level attention, our method computes the attention with an average context, which causes that the different target features is computed by the same context vector representation. In addition, the context-level attention with global attention does not always work well especially on the case when multiple aspects with different polarities occur in one sentence. For example, in the sentence "great food but the service is dreadful!", "food" and "service" are two aspects with different polarities. Because the sentiment words could modify more than one aspect, such as in the given example, "great" could modify "food" and "service", which makes the attention difficult to distinguish which sentiment word is more related to the specific aspect.

To solve the two problems, we try to add the location prior information into the two attention process with the intuition that the closer to the target a word is, the higher the contribution to the classification it should be. On the one hand, adding the location prior information into the target-level attention will let the model depend more on the context words around the specific target to compute key features of the target. On the other hand, adding the location prior information into the context-level attention will guide the model to focus on the sentiment words around the target words rather than the global words.

The method proposed in this paper is inspired by the works of Chen et al. (2017) and Li et al. (2018).

We first define a variable D_i to measure the relative distance between the current word and the target. For example, given the sentence "Great food but the service was dreadful!" and its target "food", the distance of "great" to "food" is 1, the distance of "food" is 0 and the "dreadful" is 5.

We also define W_i to describe the location importance of words relative to target:

$$W_i = 1 - \frac{|len - D_i|}{len} \tag{16}$$

where *len* is the length of the sentence.

When adding the location prior information into the target-level attention, we use W_i to compute the initialization context representation instead of the average pooling method.

We first normalize W_i:

$$L_i = \frac{W_i}{\sum_{i=1}^{n} W_i} \tag{17}$$

Then, we could obtain the initialization context representation with the weighted sum function:

$$C_{init} = \sum_{i=1}^{n} L_i \cdot C_i \tag{18}$$

where C_i is the encoded features of context word i based on LSTM or embedding layer.

When adding the location prior information into the context-level attention, we use the location weights to limit the attention weights to focus on the words around the target:

$$R_i = W_i \cdot \beta_i \tag{19}$$

After normalization, we can obtain the new context attention weights:

$$\beta_i = \frac{R_i}{\sum_{i=1}^{n} R_i} \tag{20}$$

In the Coattention-LSTM, the location weights can be directly added into the target-level attention and context-level attention. In Coattention-MemNet, we just add the location weights into the target-level attention of 1-hop layer and the context-level attention of last-hop layer, which can not only add the location prior information into the network but also avoid the location weights to limit the iterative learning process of attention weights.

4. Experiments

In this section, we describe the experiment settings and design several experiments to verify the effectiveness of our methods. First, we compare our methods with recent neural network based methods to demonstrate the effectiveness of our models. Then, we analyze the effect of coattention mechanism in Coattention-LSTM and Coattention-MemNet by designing comparison experiments and visualizing some examples. Finally, we compare the models before and after adding location information to verify the effectiveness of location-enhanced method, and we also visualize some prediction examples to analyze the effectiveness and weakness of it.

4.1. Experiment preparation

4.1.1. Dataset

We conduct experiments on SemEval 2014 Task 4[1] and Twitter (Dong et al., 2014) to validate the effectiveness of our model. As shown in Table 1, the SemEval 2014 datasets consist of reviews in two categories: Restaurant and Laptop. The reviews are labeled with three sentiment polarities: positive, neutral and negative. We have statistics on the length of the target words in the datasets. The results are shown in Table 2. It can be seen that more than 1/4, 1/3 and 2/3 of targets on the restaurant, laptop and Twitter dataset contain multiple words, respectively.

4.1.2. Evaluation metric and parameters

We use accuracy as the metric for evaluating the performance of the models. The accuracy is defined as:

$$Acc = T/(T + F) \tag{21}$$

where T is the number of correctly predicted samples and F is the number of mispredicted samples. The accuracy refers to the

[1] http://alt.qcri.org/semeval2014/task4/.

Table 1
Statistics of SemEval 2014 and Twitter Datasets.

Dataset	Positive		Nerual		Negative	
	Train	Test	Train	Test	Train	Test
Restaurant	2164	728	637	196	807	196
Laptop	994	341	464	196	870	128
Twitter	1561	173	3127	346	1560	173

Table 2
Statistics of target length on SemEval 2014 and Twitter Datasets.

Dataset	Len = 1	Len = 2	Len > 2
Restaurant-Train	2720/75.38%	604/16.74%	284/7.87%
Restaurant-Test	801/71.52%	215/19.20%	104/9.29%
Laptop-Train	1473/63.27%	649/27.88%	206/8.85%
Laptop-Test	351/52.78%	209/31.43%	78/11.73%
Twitter-Train	1893/30.25%	4360/69.67%	5/0.08%
Twitter-Test	198/28.61%	492/71.10%	2/0.29%

proportion of correctly classified samples over all samples. Generally, a well performed system has a high accuracy.

In this paper, we use the 300-dimensional Glove vectors[2] to initialize the word embeddings which are trained from web data and the vocabulary size is 1.9M (Pennington, Socher, & Manning, 2014). All out-of-vocabulary words are initialized by sampling from the uniform distribution $U(-0.01, 0.01)$. All weights are initialized with uniform distribution $U(-0.01, 0.01)$, and all biases are set to zeros. The dimension of hidden states is set to 128. We train the model with a SGD optimizer, and the learning rate is set to 0.01.

4.2. Compared methods

We compare our models with the following related methods.

Majority is a basic baseline method, which assigns the largest sentiment polarity in the training set to each sample in the test set.

LSTM uses one LSTM network to model the context, and the average value of all the hidden states is used as final representation to estimate the probability of sentiment.

TD − LSTM adopts two LSTM networks to model the left context with target and the right context with target respectively. The left and right target-dependent representations are concatenated to predict the sentiment polarity of the target (Tang et al., 2016a).

AE − LSTM first models the context words via LSTM networks and then combines the word hidden states with aspect embeddings to generate the attention vectors (Wang et al., 2016).

ATAE − LSTM is developed based on AE-LSTM. ATAE-LSTM combines the aspect embeddings with each word embedding vector to represent the context (Wang et al., 2016).

AF − LSTM learns to attend based on associative relationships between sentence words and aspects (Tay et al., 2018).

IAN interactively learns attentions in the contexts and targets, and generates the representations for targets and contexts separately (Ma et al., 2017).

MemNet applies attention multiple times, and the output of the last layer attention is fed to softmax for prediction (Tang et al., 2016b).

4.3. Result comparisons

Table 3shows the performance comparison results. As we can see from the results, Majority method has the worst performance, meaning that the majority sentiment polarity occupies 53.5%, 65.0% and 50.0% of all samples on the Restaurant, Laptop and Twitter testing datasets respectively. The LSTM method with an average pooling has the worst performance among all the neural network methods, because it ignores the target information. TD-LSTM outperforms LSTM over about 1–1.5%, since it is developed from the standard LSTM and models the left and right contexts with target separately. AE-LSTM outperforms TD-LSTM over about 0.5–1%, because it models the relationship between target and context using attention. ATAE-LSTM and AF-LSTM learn to attend based on associative relationships between context words and aspects, and the performance has a slight improvement. IAN outperforms ATAE-LSTM and AF-LSTM over about 2–3%, since it first considers modeling the target with attention which really helps to improve the classification accuracy. MemNet introduces the memory network into the ABSA task for the first time. After stacking multiple computing units, the accuracy rate increases about 2% over AE-LSTM. Especially, the Coattention-LSTM we proposed in this paper shows an improvement about 1–1.5% over IAN. Furthermore, the Coattention-MemNet has an improvement about 1–2% over

[2] https://nlp.stanford.edu/projects/glove/.

Table 3
The performance (classification accuracy) of different methods[a].

Model	Restaurant	Laptop	Twitter
Majority	0.535	0.650	0.500
LSTM	0.743	0.665	0.665
TD-LSTM	0.756	0.681	0.666
AE-LSTM	0.762	0.689	–
ATAE-LSTM	0.772	0.687	–
AF-LSTM	0.754	0.688	–
IAN	0.781*	0.721*	0.698*
MemNet	0.787*	0.708*	0.685*
Coattention-LSTM	0.788	**0.735**	**0.715**
Coattention-MemNet	**0.797**	0.729	0.705

[a] The results with * are reproduced under the same conditions with the original paper. Other results are mainly from Ma et al. (2017).

MemNet. The results show that our models achieve better performance in the ABSA task. Next, we will give a detailed analysis of the two networks.

4.4. Analysis of networks

4.4.1. Analysis of coattention-LSTM

a) The effect of coattention in Coattention-LSTM

In this part, we design multiple models to verify that Coattention-LSTM proposed in Section 3 is a more effective interaction network for ABSA task. These models are described below.

No − Target only uses one LSTM network to model the context and the hidden state of last word is used for classification.

Target − Context uses two LSTMs to model context and target separately and the concatenated result of final hidden states of two LSTMs is used for classification.

Target2Context uses two LSTMs to model context and target separately. It first gets the average target representation through an average pooling to the hidden states of target, and then obtains the context attention representation for classification based on target attention representation.

Self − target2Context uses two LSTMs to model context and target separately. It first learns the target representation via a self-local attention and then generates the context representation for classification.

Interactiveattention uses two LSTMs to model context and target separately. It first generates the context attention representation with the average target representation and then generates the target attention representation with the average context representation. Finally, the concatenated result is used for classification.

Coattention uses two LSTMs to model context and target separately. It first obtains the target attention representation by the average context representation and then generates the context attention representation for classification with the target attention representation.

The performance of all compared models are reported in Table 4. We discuss the results as follows.

First, we compare the results obtained by No-Target and Target-Context. The results show that the performance of these two models is close. It indicates that directly adding target representation into final representation only has a slight improvement. We also find that the concatenation of context and target representations would sometimes introduce sentiment noise. For example, in the sentence “My only complaint might be the fortune cookies - I’ve never had a cookie predict bad luck for me before.”, the sentiment polarity of the target “fortune cookies” is negative. However, the word “fortune” in the target “fortune cookies” is a positive sentiment word. If the target representation is concatenated into the final representation for sentiment classification, it will introduce sentiment noise and may obtain a wrong classification result. We believe that the sentiment of targets should be determined only by the related contexts. Therefore, we use the context attention representation for classification, which will make the network more reasonable and robust.

Second, we compare the results obtained by Target2Context with Target-Context and No-Target. We observe that Target2Context

Table 4
Analysis of Coattention-LSTM.

Model	Restaurant	Laptop	Twitter
No-Target	0.770	0.706	0.682
Target-Context	0.772	0.705	0.684
Target2Context	0.775	0.712	0.692
Self-target2Context	0.768	0.702	0.692
Interactive Attention	0.781	0.721	0.698
Coattention	**0.788**	**0.735**	**0.715**

Table 5
Illustration of attention weights for target.

Model	taglierini	with	truffles
IAN	0.41	0.31	0.28
Coattention-LSTM	0.76	0.12	0.12

outperforms the other two models without attention. Such results prove that attention is effective for the ABSA task.

Third, we compare the performance of Coattention with Target2Context and Self-target2Context. The results show that Self-target2Context has the worst performance. The reason is that Self-target2Context learns target attention representation using self attention without the guidance of context. It leads to the fact that the self attention on targets cannot obtain an effective target attention representation. Therefore, we argue that context is helpful to the learning of target attention representation and target learning is also helpful to the learning of context attention representation, which is also the main reason that we design the coattention network.

Finally, we compare the performance of Coattention with Target2Context and Interactive Attention. We find that Interactive Attention outperforms Target2Context over only about 0.5–1% and Coattention outperforms Target2Context over about 1.5–2.5%. The Interactive Attention considers modeling target with attention and adopts parallel interaction mechanism so that the network has a better performance than Target2Context. However, it still uses an average vector of target to compute attention score for context which leads to an ineffective use for target attention result. Besides, the attention representation learned for target is also directly concatenated to the final representation. Coattention with an alternating coattention mechanism could overcome these shortcomings so that it can outperforms Interactive Attention. b) Visualize the effect of Coattention-LSTM

To prove that our model can reduce the effect of noise words such as preposition (“in”, “of”, etc.) and fully utilize the key words of targets to learn more effective context representation, we compare the visualization of IAN and our model given a sentence “Definitely try the taglierini with truffles - it was incredible” and its target “taglierini with truffles”. Obviously, “with” is a noise word and “taglierini” is the central word of the target which is more important than “truffles”.

From the results in Table 5, we can find that both IAN and our model assign the highest weight on “taglierini”. It means that both IAN and our model can reduce the effect of noise word and learn the key features of target. However, the effective target features are directly used for classification and not further used to interact with context so that the effect of context features learned by IAN is rather poorer than our model, which is illustrated in the Table 6. From Table 6, we find that IAN scatters high weights on the four words as “definitely”, “try”, “taglierini” and “incredible”, while our model concentrates high weights on two sentiment words as “definitely” and “incredible”. Thus, our model can learn more effective features for target and context than IAN. From Tables 5 and 6, we conclude that our model can learn more effective features on both target and context. Therefore, the Coattention-LSTM is a more effective interaction network for the ABSA task.

4.4.2. Analysis of Coattention-MemNet

a) The effect of coattention in Coattention-MemNet

In this part we discuss that Coattention-MemNet is effective for ABSA task by comparing our model with MemNet. As shown in Table 7, we compare the performance of MemNet and our model when the attention layer is set from 1 to 4. We also have also tried to make the hop layers deeper, but the results of both the MemNet and our model over 4 layers are worse or close to the best result within 4 layers. Therefore, we set the attention layer from 1 to 4. From the results in Table 7, we can find that our model achieves better results which outperforms MemNet over 4.6%, 5.6% and 1.9% on Restaurant, Laptop and Twitter datasets when the network has only one hop layer, respectively. Compared with results in Table 3, the performance of our model with one hop layer still exceeds many methods. It shows that the coattention is effective in this task again. With the computation unit repeated, performance continues to be improved about 1–1.5%, which indicates that multiple coattention layers are effective for this task. When the hop layer is repeated 3 times, our model obtains the best results which outperforms MemNet over 1%∽2% when MemNet obtains the best accuracy. The results show that Coattention-MemNet is more effective than MemNet for the ABSA task. b) Visualize the effect of Coattention-MemNet

Table 6
Illustration of Attention Weights for Context.

Words	IAN	Coattention-LSTM
Definitely	0.22	0.12
try	0.22	0.06
the	0.03	0.06
taglierini	0.14	0.07
with	0.03	0.06
truffles	0.03	0.06
it	0.03	0.06
was	0.03	0.06
incredible	0.22	0.44

Table 7
Analysis of Coattention-MemNet.

Hop layers	Restaurant		Laptop		Twitter	
	MemNet	Coattention-MemNet	MemNet	Coattention-MemNet	MemNet	Coattention-MemNet
Hops = 1	0.742	0.788	0.660	0.716	0.672	0.691
Hops = 2	0.777	0.794	0.708	0.726	0.685	0.689
Hops = 3	0.787	0.797	0.707	0.729	0.679	0.705
Hops = 4	0.781	0.795	0.699	0.728	0.677	0.703

We further examine whether the Coattention-MemNet can alternately learn the evidences from the target and context through a visualization of three layers network. Given a sentence “Tired windows 8 and hated it!!!” and its target “windows8”, TableEs 8 and 9 show the attention weights for target and context in different hop layers. From the results in Table 8, we can see that the target-level attention assigns the same weight on the “windows” and “8” in hop 1 layer, and assigns a higher weight on “8” in hop 2 and 3 layer. Generally, the central word of the target “windows8” should be “windows”. However, there have different versions operating systems in the Laptop dataset such as “windows7”, “windows XP” and “windows vista”. Hence, the network may learn to focus more on the “8” to distinguish between these versions. The results show that our model can learn effective target attention features.

From the results in Table 9, we can find that the context-level attention assigns higher weights on “Tired” and “hated” in hop 1 layer. It decreases the weight of “Tired” and increases the weight of “hated” in hop 2 and 3 layer that “hated” has a stronger negative tendency than “Tired”. Associating with the results in Table 8, we conclude that our model can learn evidences from the target and context alternately and the model can adjust the learned features to make them more effective based on the multiple hops structure. The Coattention-MemNet first obtains the target features at hop 1 layer based on the initial context representation which is obtained with an average pooling function. And then the model can obtain the context features based on the learned target features at hop 1 layer. Afterwards, the target-level attention network adjusts the weights on target words based on the new context features and obtains more effective target features at hop layer 2. The context-level attention also adjusts the weights on context words based on new target features and the history context features, then it obtains more effective sentiment features. Therefore, Coattention-MemNet is an effective network which can learn the key features from the target and context alternately and obtain the multiple levels of abstraction via a multiple iterations.

4.5. Analysis of location-enhanced coattention

We further examine the effectiveness of the location-enhanced coattention mechanism. The results in Table 10 show the method achieve better performance in most cases. Especially, the results on restaurant have about 1% improvement. The results indicate that the location weighted function proposed in this paper is effective.

To verify that the location weighted function could limit the attention weights to focus on the sentiment words around the target, we only use the embedding layer to encode the context and target words and use the coattention mechanism with location weighted function to predict the sentiment polarities, which is exactly a one-hop layer Coattention-MemNet with location weights. The prediction examples is listed in Table 11. Each row of the table includes a sentence and some triples. The first item of the triple is one target of the given sentence and those important words of the targets are colored red. The second item is a label and predicted result of the target with the notations P, N and O, corresponding to represent positive, negative and neutral sentiment result respectively. The third item is the words that the model gives more attention than other words of context.

In the first example, the model could focus on the important sentiment words “great” and “dreadful” even given the different targets. However, because of the limitation of location weights, the model could give more weight to “great” (0.29) than “dreadful” (0.13) when the target is “food”, and the model gives a more weight to “dreadful” (0.25) than “great” (0.16) when the target is “service”. Finally, the model can predict the polarities correctly.

From the Examples 2, 3 and 4 of Table 11, we can see that the model could first attend the key words of target when the target contains multiple instances because of the effect of coattention mechanism. Then, we find that the context-level attention could attend the important sentiment features around the target words and the model could give correct polarities because the location weights limit the model to ignore the long distance sentiment features. For this reason, the model gives a wrong label to the “bar menu” in the sentence of Example 5. It needs a global sentiment features “disappointingly” to predict a correct label for the“bar menu”, but our model could only obtain the local sentiment features “been taken off” even if the attention is a global function but limited by the location weights. This is one shortcoming of our method, and we will solve this problem in future.

Table 8
Illustration of attention weights for target in different hop layer.

Target	Hop 1	Hop 2	Hop3
Windows	0.50	0.12	0.13
8	0.50	0.88	0.87

Table 9
Illustration of attention weights for context in different hop layer.

Context	Hop1	Hop2	Hop3
Tired	0.29	0.22	0.21
windows	0.05	0.05	0.06
8	0.05	0.05	0.06
and	0.05	0.05	0.05
hated	0.35	0.42	0.41
it	0.05	0.05	0.05
!	0.05	0.05	0.06
!	0.05	0.05	0.06
!	0.05	0.05	0.06

Table 10
Effect of location.

Model	Restaurant	Laptop	Twitter
Coattention-LSTM	0.788	0.735	0.715
Coattention-LSTM + location	0.796	0.729	0.718
Coattention-MemNet	0.797	0.729	0.705
Coattention-MemNet + location	0.806	0.734	0.708

Table 11
Example predictions.

Sentence	Targets and their relevant words
1. Great food but the service was dreadful!	(food, P✓, Great dreadful) (service, N✓, Great dreadful)
2. Great beer selection too, something like 50 beers.	(beer selection, P✓, Great beer selection too) (beers, O✓, too something beers)
3. The staff members are extremely friendly and even replaced my drink once when I dropped it outside.	(staff members, P✓, extremely friendly) (drink, O✓, even replaced drink)
4. Probably my worst dining experience in new york, and I'm a former waiter so I know what I'm talking about.	(dining experience, N✓, worst) (waiter, O✓, former talking)
5. Disappointingly, their wonderful Saketini has been taken off the bar menu.	(Saketini, P✓, Disappointingly wonderful Saketini) (bar menu, N✗, been taken off)

5. Conclusion and future work

In this paper, we first propose an alternating coattention mechanism which designs an alternating learning process for both target-level and context-level feature extraction in a more intuitive and effective way for aspect-based sentiment analysis task. Then we propose Coattention-LSTM network based on coattention mechanism which could learn both attention representations for target and context alternately, and we show that coattention mechanism in Coattention-LSTM could reduce the effect of noise words of targets and fully utilize the key words of targets to learn more effective context representation for sentiment analysis. The results on SemEval 2014 Datasets and Twitter Dataset indicate that Coattention-LSTM could outperform most of existing LSTM-attention methods. Further, we propose Coattention-MemNet which can learn the key features from the target and context alternately and adjust the learned features to make them more effective based on the multiple hops structure. The performance on SemEval 2014 Datasets and Twitter Dataset shows that the Coattention-MemNet achieves better performance than the conventional memory network. Finally, we propose a location weighted function for coattention mechanism based on the relative location information, which can generate different context features for multiple targets in one comment and limit the network to focus on the words around the target. We apply the location weighted function in both Coattention-LSTM and Coattention-MemNet and the results on SemEval 2014 Datasets and Twitter Dataset indicate the effectiveness of this function.

Since our methods use LSTM or word embedding layer to represent the words and attention is only a weighted sum function, it is difficult for them to learn the complex relationship between words such as the negations modifiers and the implicit sentiment phrases. Therefore, we need to improve the LSTM or attention, in order to make the network learn complex relationship between context words or map the results of attention layer to the classification space in a non-linear manner. In future work, we plan to design a context learning function capture the relationship between local context words with a complexity between word embedding and LSTM. We will also consider adding external knowledge into the network to solve the problem of negations modifier modeling and identify unknown sentiment words and phrases.

Acknowledgments

This work was supported in part by National Natural Science Foundation of China under Grant no. 61702176 and Natural Science Foundation of Hunan Province, China under Grant no. 2017JJ3038.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.ipm.2018.12.004.

References

Bahdanau, D., Cho, K., & Bengio, Y. (2015). *Neural machine translation by jointly learning to align and translate. International conference on learning representations (iclr 2015)*.

Bo, P., & Lillian, L. (2008). Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval, 2*(1–2), 7–9.

Bordes, A., Boureau, Y.-L., & Weston, J. (2017). *Learning end-to-end goal-oriented dialog. International conference on learning representations (ICLR 2017)*.

Chen, P., Sun, Z., Bing, L., & Yang, W. (2017). *Recurrent attention network on memory for aspect sentiment analysis. Proceedings of the 2017 conference on empirical methods in natural language processing (EMNLP 2017)*452–461.

Dang, V., & Croft, W. B. (2012). *Diversity by proportionality: An election-based approach to search result diversification. Proceedings of the 35th international ACM SIGIR conference on research and development in information retrieval (SIGIR 2012)*65–74.

Dong, L., Wei, F., Tan, C., Tang, D., Zhou, M., & Xu, K. (2014). *Adaptive recursive neural network for target-dependent twitter sentiment classification. Proceedings of the 52nd annual meeting of the association for computational linguistics (ACL 2014)*49–54.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation, 9*(8), 1735–1780.

Huang, H., Zhang, Q., & Huang, X. (2017). *Mention recommendation for twitter with end-to-end memory network. Proceedings of the twenty-sixth international joint conference on artificial intelligence (IJCAI 2017)*1872–1878.

Jiang, L., Yu, M., Zhou, M., Liu, X., & Zhao, T. (2011). *Target-dependent twitter sentiment classification. Proceedings of the 49th annual meeting of the association for computational linguistics (ACL 2011)*151–160.

Kalchbrenner, N., Grefenstette, E., & Blunsom, P. (2014). *A convolutional neural network for modelling sentences. Proceedings of the 52nd annual meeting of the association for computational linguistics (ACL 2014)*655–665.

Kim, Y. (2014). *Convolutional neural networks for sentence classification. Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP 2014)*1746–1751.

Kiritchenko, S., Zhu, X., Cherry, C., & Mohammad, S. (2014). *Nrc-canada-2014: Detecting aspects and sentiment in customer reviews. Proceedings of the 8th international workshop on semantic evaluation (SEMEVAL 2014)*437–442.

Kumar, A., Irsoy, O., Ondruska, P., Iyyer, M., Bradbury, J., Gulrajani, I., et al. (2016). *Ask me anything: Dynamic memory networks for natural language processing. International conference on machine learning (ICML 2016)*1378–1387.

Li, S., Zhou, L., & Li, Y. (2015). Improving aspect extraction by augmenting a frequency-based method with web-based similarity measures. *Information Processing & Management, 51*(1), 58–67.

Li, X., Bing, L., Lam, W., & Shi, B. (2018). *Transformation networks for target-oriented sentiment classification. Proceedings of the 56th annual meeting of the association for computational linguistics (ACL 2018)*946–956.

Liang, S., Ren, Z., & de Rijke, M. (2014). *Personalized search result diversification via structured learning. Proceedings of the 20th ACM SIGKDD international conference on knowledge discovery and data mining (SIGKDD 2014)*751–760.

Liang, S., Yilmaz, E., Shen, H., Rijke, M. D., & Croft, W. B. (2017). Search result diversification in short text streams. *ACM Transactions on Information Systems, 36*(1).

Lin, Z., Feng, M., dos Santos, C. N., Yu, M., Xiang, B., Zhou, B., et al. (2017). *A structured self-attentive sentence embedding. International conference on learning representations (ICLR 2017)*.

Liu, B. (2012). Sentiment analysis and opinion mining. *Synthesis Lectures on Human Language Technologies, 5*(1), 7–16.

Liu, J., & Zhang, Y. (2017). *Attention modeling for targeted sentiment. Proceedings of the 15th conference of the European chapter of the association for computational linguistics (EACL 2017)*572–577.

Lu, J., Yang, J., Batra, D., & Parikh, D. (2016). *Hierarchical question-image co-attention for visual question answering. Advances in neural information processing systems (NIPS 2016)*289–297.

Luong, M.-T., Pham, H., & Manning, C. D. (2015). *Effective approaches to attention-based neural machine translation. Proceedings of the 2015 conference on empirical methods in natural language processing (EMNLP 2015)*1412–1421.

Ma, D., Li, S., Zhang, X., & Wang, H. (2017). *Interactive attention networks for aspect-level sentiment classification. Proceedings of the twenty-sixth international joint conference on artificial intelligence (IJCAI 2017)*4069–4074.

Ma, R., Zhang, Q., Wang, J., Cui, L., & Huang, X. (Zhang, Wang, Cui, Huang, 2018a). *Mention recommendation for multimodal microblog with cross-attention memory network. Proceedings of the 41th international ACM SIGIR conference on research and development in information retrieval (SIGIR 2018)*195–204.

Ma, Y., Peng, H., & Cambria, E. (Peng, Cambria, 2018b). *Targeted aspect-based sentiment analysis via embedding commonsense knowledge into an attentive lstm. Proceedings of the thirty-second AAAI conference on artificial intelligence (AAAI 2018)*5876–5883.

Pennington, J., Socher, R., & Manning, C. D. (2014). *Glove: Global vectors for word representation. Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP 2014)*1532–1543.

Pontiki, M., Galanis, D., Pavlopoulos, J., Papageorgiou, H., Androutsopoulos, I., & Manandhar, S. (2014). *Semeval-2014 task 4: Aspect based sentiment analysis. Proceedings of the 8th international workshop on semantic evaluation (SEMEVAL 2014)*27–35.

Qian, Q., Huang, M., Lei, J., & Zhu, X. (2017). *Linguistically regularized lstms for sentiment classification. Proceedings of the 55th annual meeting of the association for computational linguistics (ACL 2017)*1679–1689.

Ruder, S., Ghaffari, P., & Breslin, J. G. (2016). *A hierarchical model of reviews for aspect-based sentiment analysis. Proceedings of the 2016 conference on empirical methods in natural language processing (EMNLP 2016)*999–1005.

Schouten, K., & Frasincar, F. (2016). Survey on aspect-level sentiment analysis. *IEEE Transactions on Knowledge and Data Engineering, 28*(3), 813–830.

Schouten, K., & Frasincar, F. (2018). *Ontology-driven sentiment analysis of product and service aspects. European semantic web conference (ESWC 2018)*608–623.

Schouten, K., Frasincar, F., & de Jong, F. (2017). *Ontology-enhanced aspect-based sentiment analysis. International conference on web engineering (ICWE 2017)*302–320.

Sukhbaatar, S., Szlam, A., Weston, J., & Fergus, R. (2015). *End-to-end memory networks. Advances in neural information processing systems (NIPS 2015)*2440–2448.

Sundermeyer, M., Schlüter, R., & Ney, H. (2012). *Lstm neural networks for language modeling. Proceedings of the thirteenth annual conference of the international speech communication association (ISCA 2012)*194–197.

Sutskever, I., Vinyals, O., & Le, Q. V. (2014). *Sequence to sequence learning with neural networks. Advances in neural information processing systems (NIPS 2014)*3104–3112.

Tai, K. S., Socher, R., & Manning, C. D. (2015). *Improved semantic representations from tree-structured long short-term memory networks. Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing (IJCNLP 2015)*1556–1566.

Tang, D., Qin, B., Feng, X., & Liu, T. (Qin, Feng, Liu, 2016a). *Effective lstms for target-dependent sentiment classification. Proceedings of the 26th international conference on computational linguistics (COLING 2016)*3298–3307.

Tang, D., Qin, B., & Liu, T. (2015). *Document modeling with gated recurrent neural network for sentiment classification. Proceedings of the 2015 conference on empirical*

*methods in natural language processing (EMNLP 2015)*1422–1432.
Tang, D., Qin, B., & Liu, T. (Qin, Liu, 2016b). *Aspect level sentiment classification with deep memory network. Proceedings of the 2016 conference on empirical methods in natural language processing (EMNLP 2016)*214–224.
Tay, Y., Tuan, L. A., & Hui, S. C. (2018). *Learning to attend via word-aspect associative fusion for aspect-based sentiment analysis. Proceedings of the thirty-second AAAI conference on artificial intelligence (AAAI 2018)*5956–5963.
Tubishat, M., Idris, N., & Abushariah, M. A. (2018). Implicit aspect extraction in sentiment analysis: Review, taxonomy, oppportunities, and open challenges. *Information Processing & Management, 54*(4), 545–563.
Wagner, J., Arora, P., Cortes, S., Barman, U., Bogdanova, D., Foster, J., et al. (2014). *Dcu: Aspect-based polarity classification for semeval task 4. Proceedings of the 8th international workshop on semantic evaluation (SEMEVAL 2014)*223–229.
Wang, Y., Huang, M., Zhao, L., & Zhu, X. (2016). *Attention-based lstm for aspect-level sentiment classification. Proceedings of the 2016 conference on empirical methods in natural language processing (EMNLP 2016)*606–615.
Weston, J., Chopra, S., & Bordes, A. (2015). *Memory networks. International conference on learning representations (ICLR 2015)*.
Xiong, C., Zhong, V., & Socher, R. (2017). *Dynamic coattention networks for question answering. International conference on learning representations (ICLR 2017)*.
Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., & Hovy, E. (2016). *Hierarchical attention networks for document classification. Proceedings of the 2016 conference of the north american chapter of the association for computational linguistics: Human language technologies (NAACL 2016)*1480–1489.
Yin, Y., Song, Y., & Zhang, M. (2017). *Document-level multi-aspect sentiment classification as machine comprehension. Proceedings of the 2017 conference on empirical methods in natural language processing (EMNLP 2017)*2044–2054.
Zhang, M., Zhang, Y., & Vo, D.-T. (2016). *Gated neural networks for targeted sentiment analysis. Proceedings of the thirtieth AAAI conference on artificial intelligence (AAAI 2016)*3087–3093.
Zhang, Q., Wang, J., Huang, H., Huang, X., & Gong, Y. (2017). *Hashtag recommendation for multimodal microblog using co-attention network. Proceedings of the twenty-sixth international joint conference on artificial intelligence (IJCAI 2017)*3420–3426.
Zhang, X., Zhao, J., & LeCun, Y. (2015). *Character-level convolutional networks for text classification. Advances in neural information processing systems (NIPS 2015)* 649–657.
Zhang, Z., & Lan, M. (2015). *Ecnu: Extracting effective features from multiple sequential sentences for target-dependent sentiment analysis in reviews. Proceedings of the 9th international workshop on semantic evaluation (SEMEVAL 2015)*736–741.
Zhou, P., Qi, Z., Zheng, S., Xu, J., Bao, H., & Xu, B. (2016). *Text classification improved by integrating bidirectional lstm with two-dimensional max pooling. Proceedings of the 26th international conference on computational linguistics (COLING 2016)*3485–3495.
Zhu, X., Sobhani, P., & Guo, H. (2015). *Long short-term memory over recursive structures. International conference on machine learning (ICML 2015)*1604–1612.