Multi-grained Attention Network for Aspect-Level Sentiment Classification

Feifan Fan[1] **Yansong Feng** [12*] **Dongyan Zhao**[13]
[1]Institute of Computer Science and Technology, Peking University, China
[2]MOE Key Laboratory of Computational Linguistics, Peking University, China
[3]Beijing Institute of Big Data Research, China
{fanff, fengyansong, zhaody}@pku.edu.cn

Abstract

We propose a novel multi-grained attention network (MGAN) model for aspect level sentiment classification. Existing approaches mostly adopt coarse-grained attention mechanism, which may bring information loss if the aspect has multiple words or larger context. We propose a fine-grained attention mechanism, which can capture the word-level interaction between aspect and context. And then we leverage the fine-grained and coarse-grained attention mechanisms to compose the MGAN framework. Moreover, unlike previous works which train each aspect with its context separately, we design an aspect alignment loss to depict the aspect-level interactions among the aspects that have the same context. We evaluate the proposed approach on three datasets: laptop and restaurant are from SemEval 2014, and the last one is a twitter dataset. Experimental results show that the multi-grained attention network consistently outperforms the state-of-the-art methods on all three datasets. We also conduct experiments to evaluate the effectiveness of aspect alignment loss, which indicates the aspect-level interactions can bring extra useful information and further improve the performance.

1 Introduction

Aspect level sentiment classification is a fundamental task in sentiment analysis (Pang et al., 2008; Liu, 2012), which aims to infer the sentiment polarity (e.g. positive, neutral, negative) of sentence with respect to the aspects. For example, in sentence "*I like coming back to Mac OS but this laptop is lacking in speaker quality compared to my $400 old HP laptop*", the polarity of the sentence towards the aspect "*Mac OS*" is positive while the polarity is negative in terms of aspect "*speaker quality*".

*corresponding author.

Many statistical methods, such as support vector machine (SVM) (Wagner et al., 2014; Kiritchenko et al., 2014), are employed with well-designed handcrafted features. In recent years, neural network models (Socher et al., 2011; Dong et al., 2014; Nguyen and Shirai, 2015) are studied to automatically learn low-dimensional representations for aspects and their context. Attention mechanism (Wang et al., 2016; Li et al., 2017; Ma et al., 2017) is also be studied to characterize the effect of aspect on enforcing the model to pay more attention on the important words of the context. Previous works (Tang et al., 2016b; Chen et al., 2017) mainly employed the simple averaged aspect vector to learn the attention weights on the context words. Ma et al. [2017] further proposed the bidirectional attention mechanism, which interactively learns the attention weights on context/aspect words, with respect to the averaged vector of aspect/context, respectively.

These above attention methods are all at the coarse-grained level, which simply averages the aspect/context vector to guide learning the attention weights on the context/aspect words. The simple average pooling mechanism might cause information loss, especially for the aspect with multiple words or larger context. For example, in sentence "*I like coming back to Mac OS but this laptop is lacking in speaker quality compared to my $400 old HP laptop*", the simple averaged vector of long context might lose information when steering the attention weights on aspect words. Similarly, the simple averaged vector of aspect (i.e. *"speaker quality"*) may deviate from the intuitive core meaning (*i.e. "quality"*) when enforcing the model to pay varying attentions on the context words. From another perspective, previous works all regard the aspect and its context words as one instance, and train each instance separately. However, they do not consider the relationship among

Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 3433–3442
Brussels, Belgium, October 31 - November 4, 2018.

the instances that have the same context words. The aspect-level interactions among the instances with same context might bring extra useful information. Considering the above example, intuitively, the aspect *"speaker quality"* should pay more attention on *"lacking"* and less attention on *"like"*, compared with aspect *"Mac OS"*, since they have different sentiment polarities.

In this paper, we propose a multi-grained attention network to address the above two issues in aspect level sentiment classification. Specifically, we propose a fine-grained attention mechanism (*i.e. F-Aspect2Context* and *F-Context2Aspect*), which is employed to characterize the word-level interactions between aspect and context words, and relieve the information loss occurred in coarse-grained attention mechanism. In addition, we utilize the bidirectional coarse-grained attention (*i.e. C-Aspect2Context* and *C-Context2Aspect*) and combine them with fine-grained attention vectors to compose the multi-grained attention network for the final sentiment polarity prediction, which can leverage the advantages of them. More importantly, in order to make use of the valuable aspect-level interaction information, we design an aspect alignment loss in the objective function to enhance the difference of the attention weights towards the aspects which have the same context and different sentiment polarities. As far as we know, we are the first to explore the interactions among the aspects with the same context.

To evaluate the proposed approach, we conduct experiments on three datasets: laptop and restaurant are from the SemEval 2014 Task 4 and the third one is a tweet collection. Experimental results show that our method achieves the best performance on all three datasets.

2 Related Work

Aspect-level sentiment analysis is a branch of sentiment classification, which requires considering both the sentence and aspect information.

Traditional approaches (Jiang et al., 2011; Kiritchenko et al., 2014; Vo and Zhang, 2015) regard this task as the text classification problem and design effective features, which are utilized in statistical learning algorithms for training a classifier. Kiritchenko et al. [2014] proposed to use SVM based on n-gram features, parse features and lexicon features, which achieved the best performance in SemEval 2014. Vo and Zhang [2015] designed sentiment-specific word embedding and sentiment lexicons as rich features for prediction. These methods highly depend on the effectiveness of the laborious feature engineering work and easily reach the performance bottleneck.

In recent works, there are growing studies on neural network based methods due to their capability of encoding original features as continuous and low-dimensional vectors without feature engineering. Recursive Neural Network (Socher et al., 2011; Dong et al., 2014; Nguyen and Shirai, 2015) are studied to conduct semantic compositions on tree structures, and generate representations for prediction. Methods on LSTM (Hochreiter and Schmidhuber, 1997) were proposed to model the context information and use an aggregated vector for prediction. TD-LSTM (Tang et al., 2016a) adopted LSTM to model the left context and right context of the aspect, and concatenate them as the representation for prediction. However, these works only focused on modeling the contexts without considering the aspects, which performed an important role in estimate the sentiment polarity.

Attention mechanisms (Wang et al., 2016; Lei et al., 2016; Li et al., 2017) are studied to enhance the influence of aspects on the final representation for prediction. Many approaches (Tang et al., 2016b; Chen et al., 2017) adopted the averaged aspect vector to learn the attention weights on the hidden vectors of context words. Ma et al. [2017] further proposed bidirectional attention mechanism, which also learns the attention weights on aspect words towards the averaged vector of context words. These attention methods only consider the coarse-grained level attention, through using the simple averaged aspect/context vector to steer the attention weights learning on the context/aspect words, which might cause some information loss on the long aspect or context case.

In contrast, motivated by the bidirectional attention flow approaches (Seo et al., 2017; Pan et al., 2017) in machine comprehension, we propose a fine-grained attention mechanism which is responsible for linking and fusing information from the aspect and the context words. Furthermore, we leverage both the coarse-grained and fine-grained attentions to compose the multi-grained attention network (MGAN). In addition, existing works train each instance separately. However, we

observe that the interactions among the aspects, which have the same context words, could bring extra useful information. Thus we design the aspect alignment loss in the objective function to depict such kind of relationship, which is the first work to explore the aspect-level interactions.

3 Our Approach

3.1 Task Definition

Given a sentence $s = \{w_1, w_2, \cdots, w_N\}$ consisting of N words, and an aspect list $A = \{a_1, \cdots, a_k\}$, where the aspect list size is k and each aspect $a_i = \{w_{i_1}, \cdots, w_{i_M}\}$ is a subsequence of sentence s, which contains $M \in [1, N)$ words. Aspect-level sentiment classification evaluates sentiment polarity of the sentence s with respect to each aspect a_i.

We present the overall architecture of the proposed Multi-grained Attention Network (MGAN) model in Figure 1. It consists of the Input Embedding layer, the Contextual Layer, the Multi-grained Attention Layer and the Output Layer.

3.2 Input Embedding Layer

Input Embedding Layer maps each word to a high dimensional vector space. We employ the pre-trained word vector, GloVe (Pennington et al., 2014), to obtain the fixed word embedding of each word. Specifically, we denote the embedding lookup matrix as $\mathbb{L} \in \mathbb{R}^{d_v \times |V|}$, where d_v is the word vector dimension and $|V|$ is the vocabulary size.

3.3 Contextual Layer

We employ a bidirectional Long Short-Term Memory Network (BiLSTM) on top of the embedding layer to capture the temporal interactions among words. Specifically, at time step t, given the input word embedding x, the update process of forward LSTM network can be formalized as follows:

$$i_t = \sigma(\overrightarrow{W}_i \cdot [\overrightarrow{h}_{t-1}, \overrightarrow{x}_t] + \overrightarrow{b}_i) \quad (1)$$

$$f_t = \sigma(\overrightarrow{W}_f \cdot [\overrightarrow{h}_{t-1}, \overrightarrow{x}_t] + \overrightarrow{b}_f) \quad (2)$$

$$o_t = \sigma(\overrightarrow{W}_o \cdot [\overrightarrow{h}_{t-1}, \overrightarrow{x}_t] + \overrightarrow{b}_o) \quad (3)$$

$$g_t = tanh(\overrightarrow{W}_g \cdot [\overrightarrow{h}_{t-1}, \overrightarrow{x}_t] + \overrightarrow{b}_g) \quad (4)$$

$$\overrightarrow{c}_t = f_t * \overrightarrow{c}_{t-1} + i_t * g_t \quad (5)$$

$$\overrightarrow{h}_t = o_t * tanh(\overrightarrow{c}_t) \quad (6)$$

Where σ is the sigmoid activation function, i_t, f_t, o_t are the input gate, forget gate and output gate, respectively. $\overrightarrow{W}_i, \overrightarrow{W}_f, \overrightarrow{W}_o, \overrightarrow{W}_g \in \mathbb{R}^{d*(d+d_v)}$, $\overrightarrow{b}_i, \overrightarrow{b}_f, \overrightarrow{b}_o, \overrightarrow{b}_g \in \mathbb{R}^d$, and d is the hidden dimension size. The backward LSTM does the similar process and we can get the concatenated output $h_t = [\overrightarrow{h}_t, \overleftarrow{h}_t] \in \mathbb{R}^{2d}$. Given the word embeddings of a context sentence s and a corresponding aspect a_j, we will employ the BiLSTM separately and get the sentence contextual output $H \in \mathbb{R}^{2d*N}$ and aspect contextual output $Q \in \mathbb{R}^{2d*M}$.

In addition, considering that the context words with closer distance to an aspect may have higher influence to the aspect, we utilize the position encoding mechanism to simulate the observation. Formally, the weight for a context word w_j, which has l word-level distance from the aspect (here we treat the aspect phrase as a single unit), is defined as follows:

$$w_t = 1 - \frac{l}{N - M + 1} \quad (7)$$

Specifically, we treat the weights of words within the aspect as 0 in order to focus on the context words in the sentence. Then we can obtain the final contextual outputs of context words $H = [H_1 * w_1, \cdots, H_N * w_N]$.

3.4 Multi-grained Attention Layer

Attention mechanism is a common way to capture the interactions between the aspect and context words. Previous methods (Tang et al., 2016b; Ma et al., 2017; Chen et al., 2017) only adopt coarse-grained attentions, which simply use the averaged aspect/context vector as the guide to learn the attention weights on context/aspect. However, the simple average pooling in generating the guide vector might bring some information loss, especially for the aspect with multiple words or larger context. We propose the fine-grained attention mechanism, which is responsible for linking and fusing information from the aspect and context words. This mechanism is designed to capture the word-level interactions which estimate how each aspect/context word affect each context/aspect word. In addition, we concatenate both the fine-grained and coarse-grained attention vectors to obtain the final representation. From other perspective, we observe the relationship among aspects can introduce extra valuable information. Hence, we propose an aspect alignment loss, which is designed to strengthen the at-

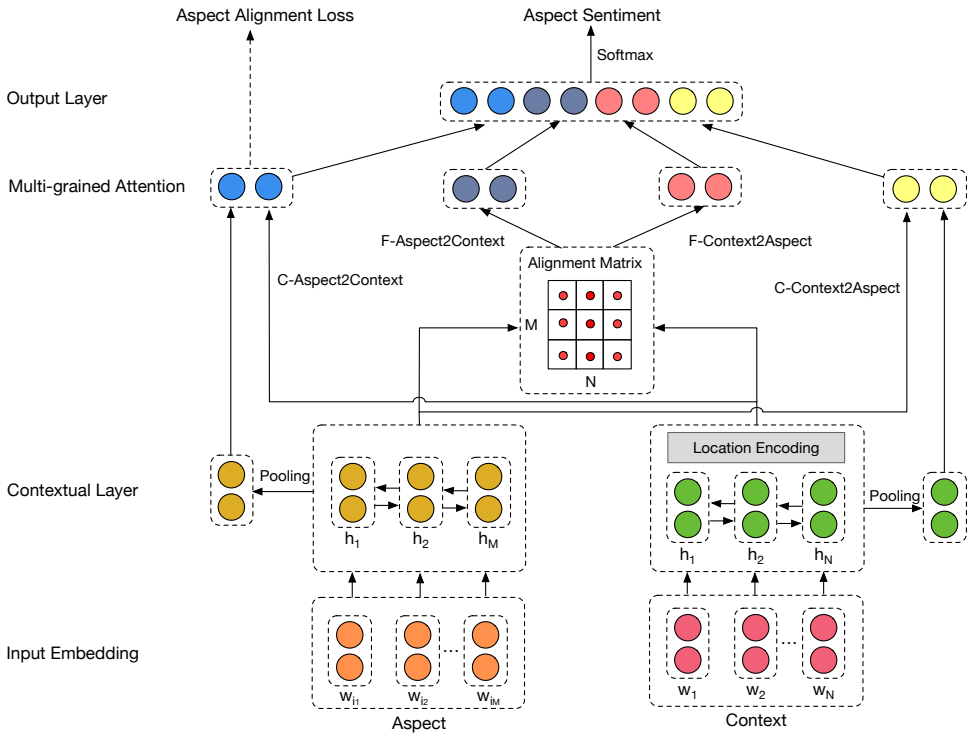

Figure 1: The architecture of the proposed multi-grained attention network.

tention difference among aspects with same context and different sentiment polarities.

Coarse-grained Attention

Coarse-grained attention is a widely used mechanism to capture the interactions between aspect and context, which utilizes an averaged aspect vector to steer the attention weights on the context words. Follow the work in (Ma et al., 2017), we employ the bidirectional attention mechanism, namely *C-Aspect2Context* and *C-Context2Aspect*.

(1) C-Aspect2Context learns to assign attention scores to the context words with respect to the averaged aspect vector. Here we employ an average pooling layer above aspect contextual output Q to generate the averaged aspect vector $Q_{avg} \in \mathbb{R}^{2d}$. For each word vector H_i in context, we can compute the attention score a_i^{ca} as follows:

$$s_{ca}(Q_{avg}, H_i) = Q_{avg} * W_{ca} * H_i \tag{8}$$

$$a_i^{ca} = \frac{exp(s_{ca}(Q_{avg}, H_i))}{\sum_{k=1}^{N} exp(s_{ca}(Q_{avg}, H_k))} \tag{9}$$

Where the score function s_{ca} computes the weight which indicates the importance of a context word towards aspect sentiment. $W_{ca} \in \mathbb{R}^{2d*2d}$ is the attention weight matrix. Then the weighted combination of the context output $m^{ca} \in \mathbb{R}^{2d}$ is calculated as follows:

$$m^{ca} = \sum_{i=1}^{N} a_i^{ca} \cdot H_i \tag{10}$$

(2) C-Context2Aspect learns to assign attention weights on aspects words, which follows the similar learning process with *C-Aspect2Context*. We utilize the average pooling mechanism to obtain the averaged context vector H_{avg}, and compute the weights for each word w_i in the aspect phrase. We compute the final weighted combination of aspect vector $m^{cc} \in \mathbb{R}^{2d}$ as follows:

$$s_{cc}(H_{avg}, Q_i) = H_{avg} * W_{cc} * Q_i \tag{11}$$

$$a_i^{cc} = \frac{exp(s_{cc}(H_{avg}, Q_i))}{\sum_{k=1}^{M} exp(s_{cc}(H_{avg}, Q_k))} \tag{12}$$

$$m^{cc} = \sum_{i=1}^{M} a_i^{cc} \cdot Q_i \tag{13}$$

where $W^{cc} \in \mathbb{R}^{2d*2d}$ is the attention weight matrix.

Fine-grained Attention

As introduced above, we propose a fine-grained attention mechanism to characterize the word-level interactions and evaluate how each aspect/context

word affect each context/aspect word. Considering the previous example "*I like coming back to Mac OS but this laptop is lacking in speaker quality compared to my $400 old HP laptop*", the word *"quality"* in aspect *"speaker quality"* should have more effect on the context words compared with word *"speaker"*. Accordingly, the context words should pay more attention on *"quality"* instead of *"speaker"*.

Formally, we define an alignment matrix $U \in \mathbb{R}^{N*M}$, between the contextual output of and the context H and the aspect Q, where U_{ij} indicates the similarity between i-th context word and j-th aspect word. The similarity matrix U is computed by

$$U_{ij} = W_u([H_i; Q_j; H_i * Q_j]) \tag{14}$$

Where $W_u \in \mathbb{R}^{1*6d}$ is the weight matrix, $[;]$ is the vector concatenation across row, $*$ is the elementwise multiplication. Then we use U to calculate the attention vectors in both directions.

(1) F-Aspect2Context estimates which context word has the closest similarity to one of the aspect word and are hence critical for determining the sentiment. We can compute the attention weights a^{fa} on context words by

$$s_i^{fa} = max(U_{i,:}) \tag{15}$$

$$a_i^{fa} = \frac{exp(s_i^{fa})}{\sum_{k=1}^{N} exp(s_k^{fa})} \tag{16}$$

where s_i^{fa} obtains the maximum similarity across column. And then we can get the attended vector $m^{fa} \in \mathbb{R}^{2d}$ as follows:

$$m^{fa} = \sum_{i=1}^{N} a_i^{fa} \cdot H_i \tag{17}$$

(2) F-Context2Aspect measures which aspect words are most relevant to each context word. Let $a_i^{fc} \in \mathbb{R}^M$ be the attention weights on aspect contextual output Q with respect to the i-th context word vector H_i. The attended aspect vector $q^{fc} \in \mathbb{R}^{2d*N}$ is defined as follows:

$$a_{ij}^{fc} = \frac{exp(U_{ij})}{\sum_{k=1}^{M} exp(U_{ik})} \tag{18}$$

$$q_i^{fc} = \sum_{j=1}^{M} a_{ij}^{fc} \cdot Q_j \tag{19}$$

Then we use an average pooling layer on q^{fc} to get the attended vector $m^{fc} \in \mathbb{R}^{2d}$:

$$m^{fc} = Pooling([q_1^{fc}, \cdots, q_N^{fc}]) \tag{20}$$

3.5 Output Layer

At last, we concatenate both the coarse-grained and fine-grained attention vectors as the final representation $m \in \mathbb{R}^{8d}$, which will be fed to a softmax layer for determining the aspect sentiment polarity.

$$m = [m^{ca}; m^{cc}; m^{fa}; m^{fc}] \tag{21}$$

$$p = softmax(W_p * m + b_p) \tag{22}$$

where $p \in \mathbb{R}^C$ is the probability distribution for the polarity of aspect sentiment, $W_p \in \mathbb{R}^{C*8d}$ and $b_p \in \mathbb{R}^C$ are the weight matrix and bias, respectively. Here we set $C = 3$, which is the number of aspect sentiment classes.

3.6 Model Training

Aspect Alignment Loss

Existing approaches train each aspect with its context separately, without considering the relationship among the aspects. However, we observe the aspect-level interactions can bring extra valuable information. In order to enhance the attention differences of aspects, which have the same context and different sentiment polarities, we design the aspect alignment loss on the C-Aspect2Context attention weights. C-Aspect2Context is employed to find the important context words in terms of a specific aspect. With the constraint of aspect alignment loss, each aspect will pay more attention on the important words through the comparisons with other related aspects. In terms of the previous example, the aspect *"speaker quality"* should pay more attention on *"lacking"* and less attention on *"like"*, compared with aspect *"Mac OS"* due to their different sentiment polarities.

Specifically, for each aspect pair a_i and a_j in aspect list A, we compute the square loss on the coarse-grained attention vector a_i^{ca} and a_j^{ca}, and also estimate the distance $d_{ij} \in [0, 1]$ between a_i and a_j as the loss weight.

$$d_{ij} = \sigma(W_d([Q_i; Q_j; Q_i * Q_j]) \tag{23}$$

$$\mathcal{L}_{align} = -\sum_{i=1}^{M-1} \sum_{j=i+1, y_i \neq y_j}^{M} \sum_{k=1}^{N} d_{ij} \cdot (a_{ik}^{ca} - a_{jk}^{ca})^2 \tag{24}$$

Where σ is the sigmoid function, $W_d \in \mathbb{R}^{1*6d}$ is weight matrix for computing the distance, y_i and y_j are the true labels of the aspect a_i and a_j, a_{ik}^{ca} and a_{jk}^{ca} are the attention weights on k-th context word towards aspect a_i and a_j, respectively.

For training the multi-grained attention network (MGAN), we should optimize all the parameters Θ from the LSTM networks: $[W_i, W_o, W_f, W_g, b_i, b_o, b_f, b_g]$, the attention and alignment loss parameters: $[W_{ca}, W_{cc}, W_u, W_d]$ and softmax parameters: $[W_p, b_p]$. The final loss function is consisting of the cross-entropy loss, aspect alignment loss and regularization item as follows:

$$\mathcal{L} = -\sum_{i=1}^{C} y_i log(p_i) + \beta \mathcal{L}_{align} + \lambda \left\| \Theta \right\|^2 \quad (25)$$

Where $\beta \geq 0$ and $\lambda \geq 0$ controls the influence of the aspect alignment loss and the L_2 regularization item, respectively. We employ the stochastic gradient descent (SGD) optimizer to compute and update the training parameters. In addition, we utilize dropout strategy to avoid overfitting.

4 Experiments

In this section, we conduct experiments to evaluate our two hypotheses: (1) whether the word-level interaction between aspect and context can help relieve the information loss and improve the performance. (2) whether the relationship among the aspects, which have the same context and different sentiment polarities, can bring extra useful information.

4.1 Experiment Setting

We conduct experiments on three datasets, as shown in Table 1. The first two are from the SemEval 2014 Task 4[1] (Pontiki et al., 2014), which contains the reviews in laptop and restaurants, respectively. The third one is a tweet collection, which are gathered by (Dong et al., 2014). Each aspect with the context is labeled by three sentiment polarities, namely *Positive*, *Neutral* and *Negative*. In addition, we adopt *Accuracy* and *Macro-F1* as the metrics to evaluate the performance of aspect-level sentiment classification, which is widely used in previous works (Tang et al., 2016b; Ma et al., 2017; Chen et al., 2017; Wang et al., 2016).

In our experiments, word embeddings for both context and aspect words are initialized by Glove (Pennington et al., 2014). The dimension of word embedding d_v and hidden state d are set to 300. The weight matrix and bias are initialized by sampling from a uniform distribution $U(0.01, 0.01)$. The coefficient λ of L_2 regularization item is 10^{-5}, the parameter β of aspect alignment loss and drop out rate are set to 0.5.

Dataset	Positive		Neutral		Negative	
	Train	Test	Train	Test	Train	Test
Laptop	994	341	870	128	464	169
Restaurant	2164	728	807	196	637	196
Twitter	1561	173	3127	346	1560	173

Table 1: The statistics of the datasets.

4.2 Compared Methods

To evaluate the performance of proposed approach, we compared with the following methods:
Majority is the basic baseline, which chooses the largest sentiment polarity in the training set to each instance in the test set.
Feature+SVM (Kiritchenko et al., 2014) uses n-gram features, parse features and lexicon features based on SVM, which achieves the state-of-the-art performance in SemEval 2014.
LSTM (Wang et al., 2016) utilizes one LSTM network to learn the hidden states and obtain the averaged vector to predict the sentiment polarity.
ATAE-LSTM (Wang et al., 2016) learns attention embeddings and combine them with the LSTM hidden states to predict the polarity.
TD-LSTM (Tang et al., 2016a) employs two directional LSTM networks, which estimate the left context and right context of the target aspect, respectively. Finally it takes the last hidden states of LSTM networks for prediction.
MemNet (Tang et al., 2016b) applys multi-hop attentions on the word embeddings, learns the attention weights on context word vectors with respect to the averaged query vector.
IAN (Ma et al., 2017) interactively learns the coarse-grained attentions between the context and aspect, and concatenate the vectors for prediction.
BILSTM-ATT-G (Liu and Zhang, 2017) models left and right context with two attention-based LSTMs and utilizes gates to control the importance of left context, right context and the entire sentence for prediction.
RAM(Chen et al., 2017) learns multi-hop attentions on the hidden states of bidirectional LSTM networks for context words, and proposes to use GRU network to get the aggregated vector from the attentions. Similar with **MemNet**, the atten-

[1]The detailed task introduction can be found in http://alt.qcri.org/semeval2014/task4/.

tion weights on context words are steered by the simple averaged aspect vector.

We also list the variants of MGAN model, which are used to analyze the effects of coarse-grained attention, fine-grained attention and aspect alignment loss, respectively.

MGAN-C only employs the coarse-grained attentions for prediction, which is similar with **IAN**.

MGAN-F only utilizes the proposed fine-grained attentions for prediction.

MGAN-CF adopts both the coarse-grained and fine-grained attentions, while without applying the aspect alignment loss.

MGAN is the complete multi-grained attention network model.

4.3 Overall Performance Comparison

Table 2 shows the performance comparison results of MGAN with other baseline methods. We can have the following observations.

(1) **Majority** performs worst since it only utilizes the data distribution information. **Feature+SVM** can achieve much better performance on all the datasets, with the well-designed feature engineering. Our method **MGAN** outperforms **Majority** and **Feature+SVM** since **MGAN** could learn the high quality representation for prediction.

(2) **ATAE-LSTM** is better than **LSTM** since it employs attention mechanism on the hidden states and combines with attention embedding to generate the final representation. **TD-LSTM** performs slightly better than **ATAE-LSTM**, and it employs two LSTM networks to capture the left and right context of the aspect. **TD-LSTM** performs worse than our method **MGAN** since it could not properly pay more attentions on the important parts of the context.

(3) **IAN** achieves slightly better results with the previous LSTM-based methods, which interactively learns the attended aspect and context vector as final representation. Our method consistently performs better than **IAN** since we utilize the fine-grained attention vectors to relieve the information loss in **IAN**. **MemNet** continuously learns the attended vector on the context word embedding memory, and updates the query vector at each hop. **BILSTM-ATT-G** models left context and right context using attention-based LSTMs, which achieves better performance than **MemNet**. **RAM** performs better than other baselines. It employs bidirectional LSTM network to generate contextual memory, and learns the multiple attended vector on the memory. Similar with **MemNet**, it utilizes the averaged aspect vector to learn the attention weights on context words.

Our proposed **MGAN** consistently performs better than **MemNet**, **BILSTM-ATT-G** and **RAM** on all three datasets. On one hand, they only consider to learn the attention weights on context towards the aspect, and do not consider to learn the weights on aspect words towards the context. On the other hand, they just use the averaged aspect vector to guide the attention, which will lose some information, especially on the aspects with multiple words. From another perspective, our method employs the aspect alignment loss, which can bring extra useful information from the aspect-level interactions.

4.4 Analysis of MGAN model

Table 3 shows the performance comparison among the variants of MGAN model. We can have the following observations.

(1) the proposed fine-grained attention mechanism **MGAN-F**, which is responsible for linking and fusing the information between the context and aspect word, achieves competitive performance compared with **MGAN-C**, especially on laptop dataset. To investigate this case, we collect the percentage of aspects with different word lengths in Table 4. We can find that laptop dataset has the highest percentage on the aspects with more than two words, and the second-highest percentage on two words. It demonstrates **MGAN-F** has better performance on aspects with more words, and make use of the word-level interactions to relieve the information loss occurred in coarse-grained attention mechanism.

(2) **MGAN-CF** is better than both **MGAN-C** and **MGAN-F**, which demonstrates the coarse-grained attentions and fine-grained attentions could improve the performance from different perspectives. Compared with **MGAN-CF**, the complete **MGAN** model gains further improvement by bringing the aspect alignment loss, which is designed to capture the aspect level interactions. Specifically, we collect the statistics of sentence-level with different aspect amounts, which is shown in Table 5. We can observe that both laptop and restaurant datasets have relatively high percentage on the sentences with multiple aspects.

Method	Laptop		Restaurant		Twitter	
	Acc	Macro-F1	Acc	Macro-F1	Acc	Macro-F1
Majority	0.5350	0.3333	0.6500	0.3333	0.5000	0.3333
Feature-SVM	0.7049	-	0.8016	-	0.6340	0.6330
ATAE-LSTM	0.6870	-	0.7720	-	-	-
TD-LSTM	0.7183	0.6843	0.7800	0.6673	0.6662	0.6401
IAN	0.7210	-	0.7860	-	-	-
MemNet	0.7237	-	0.8032	-	0.6850	0.6691
BILSTM-ATT-G	0.7312	0.6980	0.7973	0.6925	0.7038	0.6837
RAM	0.7449	0.7135	0.8023	0.7080	0.6936	0.6730
MGAN	**0.7539**	**0.7247**	**0.8125**	**0.7194**	**0.7254**	**0.7081**

Table 2: The performance comparisons of different methods on the three datasets, where the results of baseline methods are retrieved from published papers. The best performances are marked in bold.

Method	Laptop		Restaurant		Twitter	
	Acc	Macro-F1	Acc	Macro-F1	Acc	Macro-F1
MGAN-C	0.7273	0.6933	0.8054	0.7099	0.7153	0.6952
MGAN-F	0.7398	0.7082	0.8000	0.7092	0.7110	0.6918
MGAN-CF	0.7445	0.7121	0.8089	0.7135	0.7254*	0.7081*
MGAN	**0.7539**	**0.7247**	**0.8125**	**0.7194**	**0.7254**	**0.7081**

Table 3: The performance comparisons of MGAN variants. $*$ means MGAN-CF and MGAN can be regarded as the same method on twitter dataset.

Dataset	#words=1	#words=2	#words>2
Laptop	61.60%	29.16%	9.24%
Restaurant	74.47%	17.32%	8.21%
Twitter	29.99%	69.91%	0.10%

Table 4: The percentage of aspects with different word length on three datasets. Here we give the overall statistic of each dataset.

The improved performance on the two datasets shows the importance of capturing the aspect-level interactions. In terms of twitter dataset, almost all of the sentences only has one aspect. In this case, the method **MGAN** can be regarded as **MGAN-CF**.

Dataset	#aspects=1	#aspects=2	#aspects>2
Laptop	63.94%	23.32%	12.74%
Restaurant	50.89%	28.60%	20.51%
Twitter	99.91%	0.09%	0.00%

Table 5: The percentage of sentences with different aspect numbers on three datasets. Aspects with the same context are regarded as the same sentence.

4.5 Case Study

In order to demonstrate the effect of aspect alignment loss, we visualize the attention weights of the C-Aspect2Context mechanism. Figure 2 shows the attention weights of two aspects "resolution" and "fonts", whose sentiment polarities are positive and negative, respectively. From the above two bars, we can observe that the C-Aspect2Context can enforce the model to pay more attentions on the important words with respect to the aspect. For example, in terms of the aspect"resolution", the words "has", "higher" and "but" have higher attention weights compared with other words. In contrast, aspect "fonts" pays more attentions on words "but", "fonts" and "small". In addition, we evaluate the effect of aspect alignment loss, which enhances the attention difference between the aspect "resolution" and "fonts". For the two bars at bottom, we can find that aspect "fonts" has more attention on "small" and less attention on "higher", compared with the aspect "resolution". This phenomenon shows that with the constraint of aspect alignment loss, C-Aspect2Context can not only learn the important context words for each aspect, but also can make the attention gaps on the important words be as large as possible for aspects with different polarities.

5 Conclusion

In this paper, we propose a multi-grained attention network (MGAN) for aspect-level sentiment classification. Specifically, we propose a fine-grained attention mechanism, which is responsible for linking and fusing the words from the aspect and context, to capture the word-level interaction. And we combine it with the coarse-grained atten-

Figure 2: The attention visualizations on aspect "resolution" and "fonts". The above two bars are from the C-Aspect2Context attention mechanism, and the two bars at bottom are from the C-Aspect2Context attention mechanism with the constraint of aspect alignment loss.

tion mechanism to compose the MGAN model. In addition, we design an aspect alignment loss to characterize the aspect-level interactions among aspects, which have the same context and different sentiment polarities, to explore extra valuable information. Experimental results demonstrate the effectiveness of our approach on three public datasets.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (61672057, 61672058); KLSTSPI Key Lab. of Intelligent Press Media Technology. For any correspondence, please contact Yansong Feng.

References

Peng Chen, Zhongqian Sun, Lidong Bing, and Wei Yang. 2017. Recurrent attention network on memory for aspect sentiment analysis. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 452–461.

Li Dong, Furu Wei, Chuanqi Tan, Duyu Tang, Ming Zhou, and Ke Xu. 2014. Adaptive recursive neural network for target-dependent twitter sentiment classification. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 49–54.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Long Jiang, Mo Yu, Ming Zhou, Xiaohua Liu, and Tiejun Zhao. 2011. Target-dependent twitter sentiment classification. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 151–160. Association for Computational Linguistics.

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

Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2016. Rationalizing neural predictions. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 107–117.

Cheng Li, Xiaoxiao Guo, and Qiaozhu Mei. 2017. Deep memory networks for attitude identification. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, pages 671–680. ACM.

Bing Liu. 2012. Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies*, 5(1):1–167.

Jiangming Liu and Yue Zhang. 2017. Attention modeling for targeted sentiment. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, volume 2, pages 572–577.

Dehong Ma, Sujian Li, Xiaodong Zhang, and Houfeng Wang. 2017. Interactive attention networks for aspect-level sentiment classification. In *IJCAI*.

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

Boyuan Pan, Hao Li, Zhou Zhao, Bin Cao, Deng Cai, and Xiaofei He. 2017. Memen: Multi-layer embedding with memory networks for machine comprehension. *arXiv preprint arXiv:1707.09098*.

Bo Pang, Lillian Lee, et al. 2008. Opinion mining and sentiment analysis. *Foundations and Trends® in Information Retrieval*, 2(1–2):1–135.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. 2014. Semeval-2014 task 4: Aspect based sentiment analysis. *Proceedings of the*

8th International Workshop on Semantic Evaluation (SemEval 2014), pages 27–35.

Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2017. Bidirectional attention flow for machine comprehension. In *Proceedings of ICLR*.

Richard Socher, Jeffrey Pennington, Eric H Huang, Andrew Y Ng, and Christopher D Manning. 2011. Semi-supervised recursive autoencoders for predicting sentiment distributions. In *Proceedings of the conference on empirical methods in natural language processing*, pages 151–161. Association for Computational Linguistics.

Duyu Tang, Bing Qin, Xiaocheng Feng, and Ting Liu. 2016a. Effective lstms for target-dependent sentiment classification. pages 3298–3307.

Duyu Tang, Bing Qin, and Ting Liu. 2016b. Aspect level sentiment classification with deep memory network. pages 214–224.

Duy-Tin Vo and Yue Zhang. 2015. Target-dependent twitter sentiment classification with rich automatic features. In *IJCAI*, pages 1347–1353.

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

Yequan Wang, Minlie Huang, Li Zhao, et al. 2016. Attention-based lstm for aspect-level sentiment classification. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 606–615.