Journal of AI and Data Mining
Published online:

A Novel Hierarchical Attention-based Method for Aspect-level Sentiment Classification

A. Lakizadeh* and Z. Zinaty

Computer Engineering Department, University of Qom, Qom, Iran.

Received 23 April 2020; Revised 12 July 2020; Accepted 08 August 2020
* *Corresponding author: lakizadeh@qom.ac.ir (A. Lakizadeh).*

Abstract
Aspect-level sentiment classification is an essential issue in the sentiment analysis that intends to resolve the sentiment polarity of a specific aspect mentioned in the input text. The recent methods have discovered the roles of some aspects in sentiment polarity classification and have developed various techniques to assess the sentiment polarity of each aspect in the text. However, these studies do not pay enough attention to the need for vectors to be optimal for the aspects. In order to address this issue, in the present work, we suggest a Hierarchical Attention-based Method (HAM) for the aspect-based polarity classification of the text. HAM works in a hierarchically manner. Firstly, it extracts an embedding vector for the aspects. Next, it employs these aspect vectors with information content to determine the sentiment of the text. The experimental findings on the SemEval2014 dataset show that HAM can improve the accuracy by up to 6.74% compared to the state-of-the-art methods in the aspect-based sentiment classification task.

Keywords: *Deep Learning, Sentiment Analysis, Word Embedding, Long Short-term Memory.*

1. Introduction

The increasing amount of comments on the Internet has drawn both the research and industry's attentions towards the sentiment analysis. In the recent years, the sentiment analysis has been one of the focal points in Natural Language Processing (NLP). Mainly, lots of reviews have been posted by costumers in the e-commerce systems to give their feedback about a service they have received or a product that they have purchased.

Hence, the sentiment analysis is suggested as a helpful method that can aid in observing the users' opinions and predict their needs. Such data is beneficial to study the users' future demands and their consuming behaviors. Thus the users would be able to concentrate on the information that is useful and to neglect those that are less critical for them [1].

Even though most of the time opinion mining is helpful at both the document level and the sentence level, it is not accurate enough for understanding the exact polarity of the text. A positive feedback on a posted review does not necessarily signify the positive attitude of the user on everything about the entity. Also a negative feedback does not imply that he entirely hates that entity. Looking into the aspect level is required in order to reach a more accurate sentiment analysis. The fundamental task is to extract and summarize the people's feedback about what they have received or purchased and about its different aspects [2]. By aspect, we mean any property or feature of a particular entity. For instance, in terms of product reviews, the product is the entity, and everything associated with it (e.g. price, color, material) are its aspects [1]. As an example, in the sentence "Great salad but the soup tastes bad," the idea over the "salad" is obviously positive, while the idea over the "soup" is negative. In this example, the comments include different aspects of a restaurant. Estimating the aspect sentiment polarities of such comments is called the aspect term sentiment analysis (ATSA) or target sentiment analysis (TSA). In this work, aspect refers to both the aspect category and the aspect term/target. Here, the aim is the aspect-based sentiment analysis (ABSA), which includes ATSA [3]. Two thriving deep learning techniques of word

embedding are Word2Vec and Global Vectors (Glove). These two methods have been used by numerous researchers in their sentiment analysis research works.

Although the sentiment classification methods at the aspect-level that have been presented so far are very effective, a number of limitations are associated with these methods, which should be improved. For example, Word2Vec and Glove should have sizeable corpora in order to train and present an acceptable vector for each word. Being small in size, some datasets force the researchers to use such pre-trained word vectors that sometimes are not the right choice for their data. Also these embedding vectors ignore the context of the document. For instance, the word vectors for "beetle" when denoting either a car or an animal are the same. Another critical problem is neglecting the sentiment information from the given text. A consequence of this issue is that the words with opposite polarity are mapped into close vectors, a disaster for the sentiment analysis [4]. A background research work that shows that 40% of the sentiment classification errors are the result of ignoring targets in sentiment classification. Some novel approaches have become aware of the importance of aspects so they have developed various techniques to precisely model the contexts via generating aspect-specific representations. However, these studies still have limitations; for example, they consider only the one-word aspects [5] and do not try to embed the aspects separately. In order to overcome these challenges, the authors in [6] have developed ATAE-LSTM, an attention-based LSTM with an aspect embedding method to model together with the context and aspect via concatenating the aspect vector to the word embedding of the context words in the embedding layer. Also the authors in [5] have proposed "Interactive Attention Networks" (IAN) and "Aspect Fusion LSTM" (AF-LSTM) [7] so that they are able to model the context independently and use the aspect to compute the context's attention vector. "Recurrent attention network on memory (RAM)" [8] offers some information about the relative position of the context words and the particular target into their hidden state vectors. Using two stacked recurrent neural networks and a gate mechanism, Li et al. [9] have suggested a merged model to extract the opinion target and predict the target sentiment. One of the recurrent neural networks predicts combined tags, and the other one predicts a new target boundary.

In the present work, we suggest a Hierarchical Attention Model (HAM) for the aspect-based polarity classification. It works in two stages; firstly, it extracts the embedding vectors for aspects, and secondly, simultaneously, it employs these aspect vectors with information content to determine the sentiment of the input text. The main contribution of HAM is to use the two LSTM networks for modeling the aspect and context such that neural architectures can learn continuous features and the complicated relationship between an aspect and its text words. In this model, the aspects are modeled with an LSTM network, where the aspects can also contain multiple words. The LSTM network is more successful at modeling long aspects than short aspects.

Conversely, average/max pooling, which is used by other techniques, usually loses more information in modeling long aspects in comparison with the shorter aspects. This confirms the efficiency of modeling the aspects separately through the LSTM networks [5, 6]. HAM uses an aspect in the context modeling process and selects a crucial information in the context according to the aspect and keeps the critical information in the context words' hidden states. In the proposed model, the vector formed of aspect information is able to influence the context modeling procedure and filter the pointless information according to the given aspect. Thus the proposed model can generate more effective context hidden states based on the given aspect. The experimental results confirm that the proposed method can improve the accuracy of the text sentiment classification compared to the state-of-the-art methods.

2. Related works

The following section includes a brief introduction of the most recent studies on the aspect-level sentiment analysis. The traditional studies can be split into three groups: classical machine learning approaches, neural network approaches, and attention-based network approaches.

2.1. Classical machine learning methods

The traditional machine learning methods for the ABSC task are mainly based on feature engineering. As a result, collecting data and analyzing them, designing features according to the dataset features, and also obtaining enough language resources in order to develop models are time-consuming tasks. Jiang *et al*. [10] have suggested the statistical techniques that are mainly based on the success level of feature engineering measures. Kaji *et al*. [11] have suggested using structural clues that can help to extract polar sentences from the HTML documents, and building lexicon after extracting polar sentences [12]. The methods based on the traditional machine

learning are not strong enough to be generalized so applying them to a wide variety of datasets is not straightforward [13]. Also they usually require expensive artificial features like n-grams, part-of-speech tags, lexicon dictionaries, and dependency parser information [14].

2.2. Neural networks methods

The Deep Neural Network (DNN) method is possible to extract the original features into a continuous and low-dimensional vector representation without manual feature engineering. Word embedding is the foundation of most DNN-based techniques within which the words or phrases from the text are mapped to vectors of real numbers. Word2vec, PV, and Glove are the pre-trained word embedding [13]. Tang *et al*. [15] have suggested Target-Dependent LSTM (TD-LSTM) and Target-Connection LSTM (TC-LSTM), which by using them, the aspect information would be taken into account to improve the classification accuracy. Using two stacked recurrent neural networks and a gate mechanism, Li *et al*. [9] have suggested a merged model to extract the opinion target and predict the target sentiment. One of the recurrent neural networks predicts merged tags, and the other one predicts an extra target boundary [12]. Two gated neural networks have been proposed by Zhang *et al*. (2016), one of which has been employed to capture tweet-level syntactic and semantic information, and the other one has been employed to model the interactions between the left context and the right context of a particular target. Using the gating mechanism, the target affects the selection of sentiment signals in the context [16].

2.3. Attention networks methods

Wang *et al*. have suggested the AE-LSTM, AT-LSTM, and ATAE-LSTM methods. These methods mix the attention mechanisms with LSTM to semantically model sentences, which uses attention mechanisms in order to take the importance of different contextual information of a specific aspect and solve the ASA problem. Its results show that feeding the embedding of aspect or aspect terms is essential in capturing the corresponding sentiment polarity [17]. The authors in AF-LSTM [7] learned to attend based on the associative relationships between the context words and targets. Ma *et al*. have suggested the IAN model, which by using two attention networks, interactively learns the representations of the target and context. When modeling the context, the IAN model just utilizes the context words as the input; therefore, when analyzing the comments that contain several aspects, they result in similar context hidden states vectors. Also IAN models the context separately when using the information of aspect in the context's attention calculation. Moreover, the attention representations that are learned for target and context are exactly linked as the final representation. The interaction learning between the context and target is not complicated at all, and the target attention representation has not been employed appropriately [1, 3, 18].

Tang *et al*. have designed MemNet, which is made up of a multi-hop attention mechanism that has an external memory. This external memory helps to find the importance of each word in the context concerning the specific target. The memory represented is on focus by word embedding to make a better semantic information. However, in these studies, a conventional attention is used as a computation unit, and the significance of target modeling is disregarded [1,18]. Ma, Peng, and Cambria have recommended a hierarchical attention model designed to do the aspect-based sentiment analysis tasks including both the target-level attention and the sentence-level attention. However, the target-level attention was a self-attention network whose only input was the hidden output itself. The target-level attention is hard to learn without the guidance of context, and mutually, the context information will help learning the target-level attention [1].

3. Our methodology

The proposed model consists of two parts for modeling the aspect and the context, given that a context is composed of n-words $[w_c^1, w_c^2, ..., w_c^n]$ and an aspect with m words $[w_t^1, w_t^2, ..., w_t^m]$ denotes a specific word. This model aims to predict the sentiment polarity of the sentence w_c over the target w_t. Figure 1 illustrates the overall architecture of the suggested Hierarchical Attention Model (HAM) for the aspect-level polarity classification. In order to depict a word, we embed each word into a low-dimensional real-valued vector called word embedding. Two popular Embedding methods are Glove and BERT embedding. Therefore, the models are called HAM-GLOVE and HAM-BERT.

3.1. Word embedding type

3.1.1. Glove embedding

HAM-GLOVE adopts the 300-dimensional Glove vectors in order to initialize the word embeddings [19]. By sampling from the uniform distribution,

all the words that are out of vocabulary are initialized $U(-0.01, 0.01)$. Suppose $L \in R^{d_{emb} \times |v|}$ to be the pre-trained Glove embedding matrix, where d_{emb} is the dimension of the word vectors and $|v|$ is the vocabulary size. Then map each word $W^i \in R^{|v|}$ to its associated embedding vector $e_i \in R^{d_{emb} \times 1}$, which is a column in the embedding matrix L.

3.1.2. Bert embedding

The pre-trained BERT is used by BERT embedding in order to create word vectors of the sequence [20]. To make the training and fine-tuning of BERT model easy, the given context and aspect are transformed to "[CLS] + context + [SEP]" and "[CLS] + aspect + [SEP]", respectively.

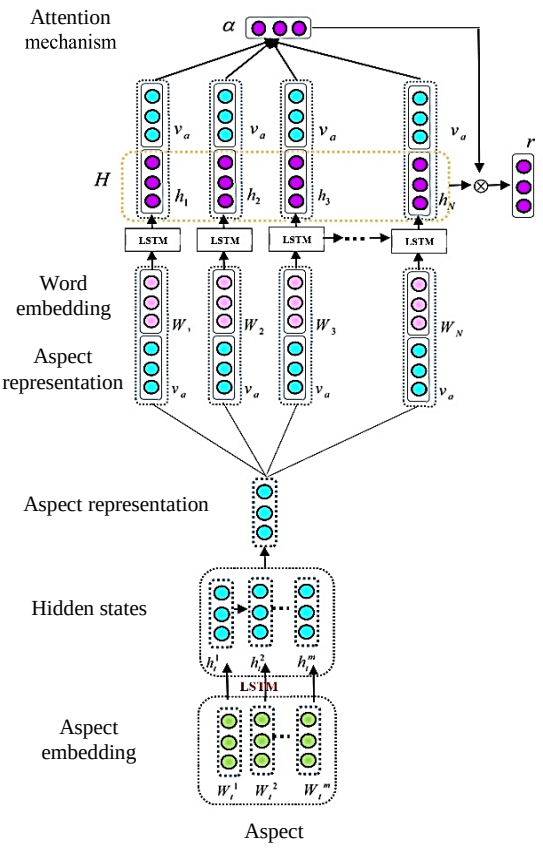

Figure 1. Architecture of the proposed model.

3.2. Aspect modeling

For a better modeling of the aspect's meaning, the LSTM networks are used to obtain the aspect's hidden states $\left[h_t^1, h_t^2, ..., h_t^m\right]$, and the initial representations of aspect (for example v_a) are obtained by averaging the hidden states.

$$v_a = \sum_{i=1}^{m} h_t^i / m \quad (1)$$

3.3. Context and aspect representation

In order to optimize the merits of the aspect information, the aspect vector is attached to the context word embedding vector. Since the words in a sentence are strongly dependent on each other, we used the LSTM networks to learn the hidden word semantics. By the way, in learning long-term dependencies, LSTM works very well, and it is also able to avoid the gradient vanishing and expansion problems. The structure of this model is illustrated in Fig 1. In this model, firstly, the aspect terms w_t are entered into the LSTM networks. In this way, the hidden output representations $\left(h_1, h_2, ..., h_N\right)$ can have information from the input aspect (v_a). Then in the next step, we modeled the interdependence between words and the input aspect. Formally, given the input word embedding and aspect representation are concatenated together as w^k, the previous cell state c^{k-1} and the previous hidden state h^{k-1}, the current cell state c^k, and the current hidden state h^k in the LSTM networks are updated as:

$$i^k = \sigma(W_i^w . w^k + W_i^h . h^{k-1} + b_i) \quad (2)$$

$$f^k = \sigma(W_f^w . w^k + W_f^h . h^{k-1} + b_f) \quad (3)$$

$$o^k = \sigma(W_o^w . w^k + W_o^h . h^{k-1} + b_o) \quad (4)$$

$$\hat{c}^k = tanh(W_c^w . w^k + W_c^h . h^{k-1} + b_c) \quad (5)$$

$$c^k = f^k \odot c^{k-1} + i^k \odot \hat{c}^k \quad (6)$$

$$h^k = o^k \odot tanh(c^k) \quad (7)$$

where i, f, and o stand for the input gate, forget gate, and the output gate, respectively, that model the interactions between the memory cells and their environments. σ is a sigmoid function. W and b indicate the weight matrices and biases, respectively. The symbol $.$ represents the matrix multiplication and $\odot$ stands for the elementwise multiplication. The hidden states $\left[h_1, h_2, ..., h_N\right]$ are considered as the word representation for context according to the specific aspect.

3.4. Attention mechanism

The attention mechanism is used to select the relevant information contributing to the sentiment polarity. It will generate an attention weight vector and a weighted hidden representation r.

$$M = tanh\left(\begin{bmatrix} W_h H \\ W_v v_a \otimes e_N \end{bmatrix}\right) \quad (8)$$

$$\alpha = softmax(w^T M) \quad (9)$$

$$r = H\alpha^T \quad (10)$$

where $M \in R^{(d+d_a)\times N}$, $\alpha \in R^N$, $r \in R^d$, $W_h \in R^{d\times d}$, $W_v \in R^{d_a \times d_a}$, and $w \in R^{d+d_a}$ are the projection parameters. r is a weighted representation of sentences with a given aspect. $H \in R^{d\times N}$ is a matrix that includes the hidden vectors $[h_1,...,h_N]$ produced by LSTM, d is the size of the hidden layers, N is the length of the input sentence, v_a represents the embedding of aspect, and $e_n \in R^N$ is a vector of 1s. The operator in 8 (a circle with a multiplication sign inside, OP for short here) means: $v_a \otimes e_N = [v_a; v_a; \ldots; v_a]$, i.e. the operator frequently concatenates v_a for N times. $W_v v_a \otimes e_N$ is repeating the linearly transformed v_a again and again until there are words in a sentence. The final sentence representation is given by:

$$h^* = tanh(W_p r + W_x h_N) \quad (11)$$

where $h^* \in R^d$, W_p, and W_x are the projection parameters that are supposed to be learned while the training process is running. The attention mechanism permits the model to take the most important part of a sentence when considering different aspects. h^* serves as a feature representation of a sentence given the input aspect. A linear layer is added to change the sentence vector to e, i.e. a real-valued vector whose length equals class number $|c|$. Then a $softmax$ layer is employed to transform e into a conditional probability distribution.

$$y = softmax(W_s h^* + b_s) \quad (12)$$

where y is the predicted sentiment polarity distribution, and W_s and b_s are the learnable parameters for the $softmax$ layer.

3.5. Regularization and model training

In terms of the text sentiment analysis, neutral polarity is a vague sentimental state, and training samples with neutral's labels is untrustworthy. Thus we use a Label Smoothing Regularization (LSR) term in the loss function, which fines low the entropy output distributions [21]. By preventing a network from assigning the full probability to each training example, LSR can lower the over-fitting chance during training and replaces the 0 and 1 targets for a classifier with smoothed values like 0.1 or 0.9. For a training sample x with the original ground-truth label distribution $G(g|x)$, we compute $G'(g|x)$ with:

$$G'(g|x) = (1-e)G(g|x) + eu(g) \quad (13)$$

where $u(g)$ represents a known distribution of label k independent of training samples, which mostly follows a simple uniform distribution, then $u(k) = \frac{1}{c}$, $e \in [0,1]$ and is the smoothing parameter. LSR corresponds to the KL distance between the known label distribution $u(g)$ and the predicted distribution p_θ. The LSR term is explained as:

$$L_{lsr} = -D_{KL}(u(g) \| p_\theta) \quad (14)$$

The proposed model is trained by improving the cross-entropy loss as much as possible with the L_{lsr} and L_2 regularizations. The training loss is as follows:

$$loss = -\sum_{i=1}^{C} y_i log(\hat{y}_i) + L_{lsr} + \lambda \|\theta\|^2 \quad (15)$$

where C is the number of classes, y_i serves as the correct sentiment polarity, and $\hat{y}_i$ presents the predicted sentiment polarity for a specific sentence. Moreover, λ is the L_2 regularization factor and θ is the parameter set of the proposed model.

4. Experiments and results

4.1. Datasets

In order to assess the suggested model, some tests were carried out on the SemEval 2014 Task4 dataset [22]. There exist two domain-specific datasets for laptops and restaurants, namely restaurants14 and laptop14. The number of the training and test samples of each sentiment polarity on the restaurant and laptop datasets is shown in table 1 [12].

Table 1. Statistical information of semeval-2014.

Property	Datasets			
	Restaurant		Laptop	
	Train	Test	Train	Test
#samples	1978	600	1462	411
#AvgLen	16.2856	15.4167	18.5855	14.9562
#TermSet	1.191	520	939	389
#AvgTermLen	2.0722	1.9942	1.9191	1.9434
#ATPS	1.8210	1.8667	1.5821	1.5523
Pos./Neg. /Neu.	2164/805 /633	728/196/ 196	987/866 /460	341/128/ 169

According to table 1, the average number of the aspects in the same sentence is about 1.8, and the average length of the aspect is about 2. This data shows that each sentence often involves more than one aspect, and each aspect usually contains more than one word.

4.2. Evaluation metrics

The following metrics are adopted to evaluate the performance of the suggested model. The accuracy is defined as:

$$Accuracy = \frac{TP + TN}{N} \quad (16)$$

in which, *TP,TN* are the number of correctly predicted samples, and N is the total number of testing samples. Since there is a three-class classification task and the classes are imbalanced, as one can see in table 1, $Macro - F1$ is calculated and the value of $Macro - F1$ is obtained as follows:

$$F1_i = 2.\frac{Precision_i.Recall_i}{Precision_i + Recall_i} \quad (17)$$

$$Macro\text{-}F1 = \frac{1}{3}\sum_i F1_i \quad (18)$$

where $i \in$ [positive, neutral, negative] .

4.3. Implementation details

In our experiments, we show the details of the configurations and use hyper-parameters in tables 2 for both the HAM-Glove and HAM-BERT models on the Restaurants and Laptop datasets. Randomly, a sample containing 20% of the original training data is employed as the development data to tune the algorithm parameters.

Table 2. Configuration and hyper-parameters of word embeddings.

Property	Word Embedding	
	Glove	BERT
Dimension	300	768
Hidden states	300	300
Initializer	Uniform (-0.1,0.1)	Uniform (-0.1,0.1)
Optimizer	Adam	Adam
Drop out	0.5	0.1
Learning rate	1e-3	2e-5
L2 Regularize	1e-5	From{1e-2, 1e-3}
Framework	PyTorch	PyTorch

Adjusting the process of Bert is a delicate process; a small learning rate would maximize the Bert's performance. The experiments performed demonstrated that a too large batch size makes the volatility of regularization between layers to bring down the performance of the model. Thus the optimal batch size from {16, 25, 32} for HAM models was adopted.

4.4. Comparison models

In order to have a fair comparison of the HAM with other methods, we report the best value that is published for each method on the same datasets. This prevents the possible implementation errors. The results obtained indicated that, to a great extent, HAM could improve the state-of-the-art performance on the two datasets, especially the HAM-BERT model. A comparison was made between the HAM design models and the following baselines: LSTM [23], CNN [24], TC-LSTM [15], AT-LSTM [6], ATAE-LSTM [6], ATAE-BiLSTM [6], MemNet [25], IAN [5], RAM [8], AF-LSTM(CORR), AF-LSTM(CONV) [7], GCAE [26], DAuM [27], IARM [28], CEA [29], MTKFN-Senti [30], AA-LSTM [3], ATAE-LSTM(AA) [3], Co-attention-LSTM [1], PG-CNN [31], BERT-AVG [32], ANTM+$BERT_B$ [17], BERT-CLS [32], SPAN- Collapsed [33], Base model + BG, Base model + BG + SC, Base model + BG + OE [9], MTKFN-struct [30], TAG-collapsed [33], BERT-Soft, BERT-Hard, BERT-Original [34], IGCN [35], BERT-LSTM, BERT-Attention [36].

4.5. Results

Tables 3 and 4 represent the performance comparison of HAM with other models. The HAM-BERT model achieves an impressive improvement compared to the state-of-the-art methods. According to the results indicated in tables 3 and 4, the model includes only the LSTM network, which has achieved the worst performance among the baseline methods. The reason is that it treats the aspects equally with other context words and does not fully use the aspect information, so it must get the same sentiment polarity, although given different aspects. TC-LSTM takes both the combination of aspect vector (average over multiple word vectors) and word embedding as the input, which results in a worse function than the proposed model. Representation of the aspect in the TC-LSTM model can cause the information to be lost, especially when the aspects have multiple words.

Compared to the ATAE-LSTM model, the proposed model (HAM-GLOVE) improves the performance in terms of the accuracy measure about 2.32% and 3.76%, and the proposed model (HAM-BERT) improves 4.73% and 9.56% in the restaurant and laptop categories, respectively, in terms of the accuracy measure. According to the results obtained, the aspect should be modeled individually, and the aspect representations can contribute to judge the sentiment polarity of a target, and the collocated context and aspect could affect each other. It means that the interaction between the aspect and the content is crucial when classifying the aspect sentiment polarity, and the unidirectional attentions do not suffice for the final representation. In the AF-LSTM model, instead of allowing the attention layer to focus on the learning of the relative importance of context words, it is

given to the extra burden of modeling the relationship between aspect and context words, and its performance has a slight improvement. Also our method outperforms the IAN model. Although GCAE incorporates the gating mechanism to control the sentiment information flow based on the input aspect, the information flow is generated by an aspect independent encoder. In terms of the accuracy measure, our model would enhance the performance compared to GCAE, by 1.7% and 5.3% in the two datasets (restaurant and laptop), respectively. Since MemNet does not model the hidden semantic of embedding, its overall performance is not satisfying; the last attention results in a simple linear combination of word embedding. The RAM method utilizes several recurrent attention models in order to gain weight in distinctive context words. In comparison with RAM, the proposed model improves the performance in terms of the accuracy measures about 1.7% and 3.6% in the restaurant and laptop categories, respectively.

Table 3. Comparison results on the restaurants dataset.

Methods	year	Reported from	Accuracy	Macro F1
LSTM	1997	[37]	74.30	63.00
CNN	2014		75.18	60.25
TC-LSTM	2016		77.41	66.72
AT-LSTM			78.04	63.37
ATAE-LSTM			76.79	63.72
ATAE-Bi-LSTM			75.98	63.43
MemNet			73.39	61.09
IAN	2017		76.70	65.12
RAM			77.41	66.76
AF-LSTM(CORR)	2018	[38]	75.96	64.00
AF-LSTM(CONV)			76.46	65.54
GCAE		[37]	77.41	65.06
DAuM		[38]	77.91	66.47
IARM			77.73	66.66
CEA			78.44	66.78
MTKFN-Senti	2019	[30]	77.74	66.30
AA-LSTM		[3]	78.21	66.24
ATAE-LSTM (AA)			78.31	66.46
Coattention-LSTM		[1]	78.80	-
PG-CNN		[31]	78.90	-
HAM-GLOVE			**79.11**	**66.81**
BERT-AVG	2019	[32]	78.70	-
Coattention-MemNet		[1]	79.7	-
ANTM+$BERT_B$		[17]	80.78	71.00
BER-CLS		[32]	81.20	-
SPAN-collapsed		[33]	-	57.85
IGCN	2020	[35]	81.34	-
HAM-BERT			**81.52**	**71.46**

According to table 4, using the BERT representations can boost the performance of our model. BERT-AVG, which uses the BERT representations without fine-tuning, achieves a surprisingly excellent performance on this task. After fine-tuning, the performance of BERT-CLS becomes even better. Our model consistently improves over BERT-AVG and BERT-CLS, which indicates that our model can better utilize these semantic representations. The accuracy of our model reaches about 81.52% and 76.96% in terms of the accuracy measure on the restaurant and laptop datasets, respectively.

Table 4. Comparison results on the laptop dataset.

Methods	year	Reported from	Accuracy	Macro F1
LSTM	1997	[37]	66.50	60.10
CNN	2014		66.93	57.75
TC-LSTM	2016		67.08	61.11
AT-LSTM			69.44	63.16
ATAE-LSTM			67.40	58.47
ATAE-Bi-LSTM			70.53	63.43
MemNet			64.42	58.10
IAN	2017		68.50	60.90
RAM			67.55	59.73
AF-LSTM(CONV)	2018	[38]	69.97	63.70
AF-LSTM(CORR)			69.78	63.38
GCAE		[37]	65.83	59.20
DAuM		[38]	70.36	65.06
IARM			68.63	63.30
CEA			70.52	64.52
Base model + BG	2019	[9]	-	54.31
Base model + BG + SC			-	55.81
Base model + BG + OE			-	55.62
AA-LSTM		[3]	66.93	61.45
ATAE-LSTM (AA)			69.28	62.10
MTKFN-Struct		[30]	69.55	62.96
PG-CNN		[31]	69.10	-
HAM-GLOVE			**71.16**	**65.07**
Coattention-MemNet	2019	[1]	72.9	-
Coattention-LSTM			73.5	-
BERT-AVG		[32]	76.50	-
ANTM+$BERT_B$		[17]	75.37	71.89
SPAN-collapsed		[33]	-	48.66
TAG-collapsed			-	65.23
BERT-Soft		[34]	74.92	-
BERT-Hard			74.10	-
BERT-Original			74.57	-
IGCN	2020	[35]	75.24	
BERT-LSTM		[36]	75.31	69.37
BERT-Attention			75.16	68.76
HAM-BERT			**76.96**	**72.23**

4.6. Analysis of proposed model

According to tables 3 and 4, the improvements in the restaurant dataset are less than those on the laptop dataset. It results in more 1-word aspect cases in the restaurant dataset compared to the number of cases with 1-word aspect in the laptop dataset (Table 5). In other words, the laptop dataset has more multi-words aspects than the restaurant category. The main contribution of the proposed method is using the two LSTM networks for modeling of the aspect and context such that the neural architectures are able to learn the continuous features and the complicated relationship between an aspect and its text words.

In this model, the aspects are modeled with an

LSTM network, whose aspects can also contain multiple words. In the proposed model, the aspect information (in the form of vectors) can influence the process of context modeling and also filter useless information for the given aspect. Therefore, it can create more effective context hidden states based on the given aspect and get the different context hidden state vectors by analyzing those comments that contain multiple aspects.

Table 5. Number/percentage of single-word and multi-word aspects in the datasets used.

Properties	Datasets	
	Restaurant	Laptop
Single-word (len = 1)	3521/74.5%	1825/61.5%
Multi-word (len = 2)	819/17.3%	857/28.9%
Multi-word (len > 2)	388/8.2%	284/9.6%

4.7 Model size and model cost

We compared the size (number of parameters), the running time of models on the 1120 samples of the ATSA task's restaurant test set, and the amount of memory used by the proposed model with the other baseline methods. For all the compared models, we used an open-source PyTorch implementation and run them on the same GPU.

The results obtained are shown in table 6. Note that the values for the running time and the memory used by the BERT-CLS method have not been mentioned in the original paper [32,39]. Using the same dimension of the hidden states, the proposed HAM-GLOVE model has a lower model size and memory compared to these LSTM-based methods. The LSTM-based models require more running time due to the time dependence of the LSTM structure. Moreover, for the models RAM and MemNet with multiple attention layers, they need more time to complete the testing process. Considering the three conditions parameter quantity, running time, and model performance, it is obvious that HAM- GLOVE is superior to the other models. Since MemNet only has one shared attention layer and two linear layers, it is the smallest model, which is not able to calculate the hidden states of word embedding. The HAM-GLOVE's lightweight ranks second because, in comparison with MemNet, it takes more parameters as the input to model the hidden states of sequences. The BERT-based models indeed have the larger model sizes, and when we switch from Glove embedding to BERT representations, the size of the model increases. Compared to BERT-CLS, the proposed HAM-BERT model has fewer parameters and model size, and is more accurate on the restaurant and laptop datasets.

Table 6. Model Size and model cost of some models for the restaurant dataset.

Models	Model size		Model cost
	Params $\times 10^6$	Memory (MB)	Time Complexity (s)
TC-LSTM	2.1666	14.30	7.153
ATAE-LSTM	2.52	16.61	12.396
ATAE-BiLSTM	2.25	15.58	7.40
IAN	2.168	15.30	12.803
RAM	5.77	31.18	30.80
MemNet	0.36	7.82	19.64
LCRS	3.25	18.51	18.76
HAM-GLOVE	**2.1663**	**14.17**	**13.68**
HAM-BERT	**112.77**	**452.22**	**207.18**
BERT-CLS	335.14	-	-

4.8. Comparison between Glove and Bert

Tables 7 and 8 and figures 2 and 3 show that when we use the BERT pre-training vectors in the proposed model, the overall performance is much better than that of the Glove vectors. BERT has an advantage over other models like Glove because in Word2Vec and Glove, each word has a fixed representation without being influenced by the context within which the word appears. In contrast, BERT generates word dynamically informed representations, considering the words around them. Also Glove does not take into account word order in training; however, BERT takes into account the word order.

Table 7. Results of Glove and Bert vectors in the proposed model in terms of different evaluation measures using the restaurant dataset.

Measures	Labels	HAM-GLOVE	HAM-BERT
Accuracy	Negative	52.60	59.50
	Neutral	27.30	32.91
	Positive	79.40	82.13
Macro-F1	Negative	68.90	74.64
	Neutral	42.90	49.52
	Positive	88.50	90.19
Precision	Negative	67.00	69.13
	Neutral	66.00	66.54
	Positive	83.70	87.67
Recall	Negative	71.40	81.12
	Neutral	31.60	39.79
	Positive	91.90	92.85

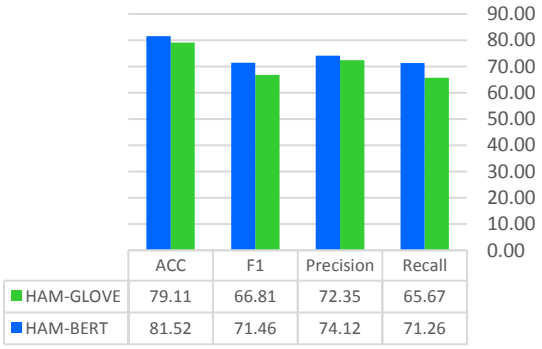

Figure 1. Results of Glove and Bert vectors in the proposed model in terms of different evaluation measures using the restaurant dataset.

Table 8. Results of Glove and Bert vectors in the proposed model in terms of different evaluation measures using the laptop dataset.

Measures	Labels	HAM-GLOVE	HAM-BERT
Accuracy	Negative	45.02	53.20
	Neutral	33.16	42.65
	Positive	71.39	77.55
Macro-F1	Negative	62.90	69.53
	Neutral	49.80	59.80
	Positive	83.30	87.35
Precision	Negative	53.37	64.23
	Neutral	67.00	68.18
	Positive	81.10	85.63
Recall	Negative	74.20	75.78
	Neutral	39.60	53.25
	Positive	85.60	89.14

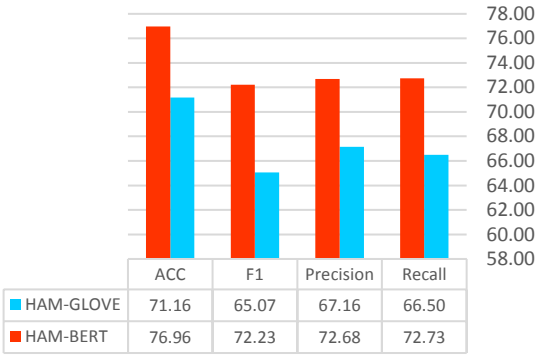

Figure 2. Results of Glove and Bert vectors in the proposed model in terms of different evaluation measures using the laptop dataset.

5. Conclusions

According to the impotence of the aspect-level sentiment classification in the sentiment analysis, in this paper, we proposed HAM, A hierarchical attention model, to resolve the sentiment polarity of a specific aspect mentioned in the text. HAM works in two stages: firstly, it extracts an embedding vector for the aspects; secondly, simultaneously, it employs these aspect vectors with the information content to determine the sentiment of the text. The primary benefit that the proposed model contributes is that the aspect information that is represented as a vector would influence the process of context modeling. It also filters useless information for the given aspect. Therefore, HAM can create more effective context hidden states based on the given aspect and get the different context hidden state vectors by analyzing those comments that contain multiple aspects. The experimental results on the SemEval 2014 datasets reveal that the model we proposed can learn the practical features and obtain a superior performance over the baseline models. ASC is a fine-grained and complex task, and thus many other approaches like handling sentiment negation can be adopted. The expand and improve mathematical relationships in the attention mechanism achieve a higher accuracy. We believe all these can help improve the sentiment analysis and provide more effective solutions in the future that will increase the accuracy in this field.

References

[1] Yang C., et al. (2019). Aspect-based sentiment analysis with alternating coattention networks. Information Processing & Management, vol. 56, no. 3, pp.463-478.

[2] Zhang, L. & Liu, B. (2014). Aspect and entity extraction for opinion mining. in data mining and knowledge discovery for big data, pp. 1-40.

[3] Xing B., et al. (2019). Earlier attention? Aspect-aware LSTM for aspect-based sentiment analysis. In Proceedings of the 28th International Joint Conference on Artificial Intelligence.

[4] Rezaeinia, S.M., Ghodsi, A. & Rahmani, R. (2017). Improving the accuracy of pre-trained word embeddings for sentiment analysis.

[5] Ma, D., Li, S., Zhang, X. & Wang, H. (2017). Interactive attention networks for aspect-level sentiment classification. arXiv preprint arXiv:1709.00893.

[6] Wang, Y., Huang, M., Zhu, X. & Zhao, L. (2016). Attention-based lstm for aspect-level sentiment classification. In proceedings of the 2016 conference on empirical methods in natural language processing, pp. 606-615.

[7] Tay, Y., Luu, A. T. & Hui, S. C. (2017). Learning to attend via word-aspect associative fusion for aspect-based sentiment analysis.

[8] Chen, P., Sun, Z., Bing, L. & Yang, W. (2017). Recurrent Attention Network on Memory for Aspect Sentiment Analysis. In Proceedings of the 2017 conference on empirical methods in natural language processing, pp. 452-461.

[9] Li X., et al. (2019). A unified model for opinion target extraction and target sentiment prediction. In Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, pp. 6714-6721.

[10] Jiang, L., Yu, M., Zhou, M., Liu, X. & Zhao, T. (2011). Target-dependent twitter sentiment classification. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, pp. 151-160.

[11] Kaji, N. & Kitsuregawa, M. (2007). Building lexicon for sentiment analysis from massive collection of HTML documents. In Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL), pp. 1075-1083.

[12] Zhang, Q., & Lu, R. (2019). A Multi-Attention Network for Aspect-Level Sentiment Analysis. Future Internet, vol. 11, no. 7, pp. 157.

[13] Zeng B., et al. (2019). LCF: A Local Context Focus Mechanism for Aspect-Based Sentiment Classification. Applied Sciences, vol. 9, no. 16, pp. 3389.

[14] Cai, N., Ma, C., Wang, W. & Meng, D. (2019). Effective Self Attention Modeling for Aspect Based Sentiment Analysis. In International Conference on Computational Science, pp. 3-14.

[15] Tang, D., Qin, B., Feng, X. & Liu, T. (2015). Effective LSTMs for target-dependent sentiment classification.

[16] Gu, S., Zhang, L., Hou, Y. & Song, Y. (2018). A position-aware bidirectional attention network for aspect-level sentiment analysis. In Proceedings of the 27th International Conference on Computational Linguistics, pp. 774-784.

[17] Mao Q., et al. (2019). Aspect-Based Sentiment Classification with Attentive Neural Turing Machines. In Proceedings of the 28th International Joint Conference on Artificial Intelligence, pp. 5139-5145.

[18] Song Y., et al. (2019). Attentional encoder network for targeted sentiment classification. arXiv preprint arXiv:1902.09314.

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

[20] Devlin, J., Chang, M. W., Lee, K. & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

[21] Szegedy C., et al. (2016). Rethinking the inception architecture for computer vision. In proceedings of the IEEE conference on computer vision and pattern recognition, pp. 2818-2826.

[22] Pontiki M., et al. (2014). SemEval-2014 Task 4: Aspect Based Sentiment Analysis. In Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), vol. 14.

[23] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural computation, vol. 9, no. 8, pp. 1735-1780.

[24] LeCun, Y. & Bengio, Y. (1995). Convolutional networks for images, speech, and time series. The handbook of brain theory and neural networks, vol. 3361, no. 10, pp. 1995.

[25] Tang, D., Qin, B. & Liu, T. (2016). Aspect level sentiment classification with deep memory network
Xue, W. & Li, T. (2018). Aspect based sentiment analysis with gated convolutional networks.

[26] Zhu, P. & Qian, T. (2018). Enhanced aspect level sentiment classification with auxiliary memory. In Proceedings of the 27th International Conference on Computational Linguistics, pp. 1077-1087.

[27] Majumder N., et al. (2018). IARM: Inter-aspect relation modeling with memory networks in aspect-based sentiment analysis. In proceedings of the 2018 conference on empirical methods in natural language processing, pp. 3402-3411.

[28] Yang J., et al. (2019). Multi-entity aspect-based sentiment analysis with context, entity, and aspect memory. In Thirty-Second AAAI Conference on Artificial Intelligence, vol. 18, no. 4, pp. 1-22.

[29] Wu S., et al. (2019). Aspect-based sentiment analysis via fusing multiple sources of textual knowledge. Knowledge-Based Systems, vol. 183, pp. 104868.

[30] Huang, B. & Carley, K. M. (2019). Parameterized convolutional neural networks for aspect level sentiment classification. arXiv preprint arXiv:1909.06276.

[31] Huang, B. & Carley, K. M. (2019). Syntax-Aware Aspect Level Sentiment Classification with Graph Attention Networks. arXiv preprint arXiv:1909.02606.

[32] Hu M., et al. (2019). Open-Domain Targeted Sentiment Analysis via Span-Based Extraction and Classification. arXiv preprint arXiv:1906.03820.

[33] Hu M., et al. (2019). Learning to Detect Opinion Snippet for Aspect-Based Sentiment Analysis. arXiv preprint arXiv:1909.11297.

[34] Kumar A., et al. (2020). Aspect Based Sentiment Classification Using Interactive Gated Convolutional Network. IEEE Access, vol. 8, pp. 22445-22453.

[35] Song Y., et al. (2020). Utilizing BERT Intermediate Layers for Aspect Based Sentiment Analysis and Natural Language Inference. arXiv preprint arXiv:2002.04815.

[36] Zhou J., et al. (2019). Deep Learning for Aspect-Level Sentiment Classification: Survey, Vision, and Challenges. IEEE Access, vol. 7, pp. 78454-78483.

[37] Chen, Z. & Qian, T. (2019). Transfer Capsule Network for Aspect Level Sentiment Classification. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 547-556.

[38] Keshavarz, H. R. & Saniee Abadeh, M. (2018)."MHSubLex: Using metaheuristic methods for subjectivity classification of microblogs." Journal of AI and Data Mining, vol. 6, no. 2, pp. 341-353.