Multimedia Tools and Applications (2022) 81:15333–15348
https://doi.org/10.1007/s11042-022-12487-x

Sentence constituent-aware attention mechanism for end-to-end aspect-based sentiment analysis

Ting Lu[1] · Yan Xiang[1] · Li Zhang[1] · Jiqun Zhang[1]

Received: 23 July 2021 / Revised: 16 December 2021 / Accepted: 25 January 2022 /
Published online: 28 February 2022

Abstract
End-to-end aspect-based sentiment analysis aims to complete aspect terms extraction and aspect sentiment classification simultaneously. Most existing methods ignore the sematic connection between the two subtasks. In this paper, we solve the problem by inducing constituents from input sentences, and propose a novel model based on sentence constituent-aware attention mechanism for end-to-end aspect-based sentiment analysis. Our framework mainly involves three layers. The first layer gets word representations by the pre-trained language model. Followed by the proposed sentence constituent-aware attention layer to induce constituents from the input sentence. With the operation of inducing constituents, the words in the same constituent are constrained to attend to each other, making the aspect term pay more attention to its corresponding opinion. Finally, a simple linear classification layer is adopted to predict the unified tags. Experimental results demonstrate that the proposed model outperforms other baselines on four benchmark datasets.

Keywords Sentence constituent · Attention mechanism · Aspect-based sentiment analysis · Aspect term-polarity co-extraction · Transformer · Deep learning

1 Introduction

Aspect-based sentiment analysis (ABSA) aims to identify people's sentiment polarity of specific aspects in a review. For example, in the sentence "*So what if the laptops look chic and cool the after sales support is terrible.*", the aspect terms are "*look*" and "*after sales support*", and the sentiment polarity of them are *positive* and *negative* respectively. Recently, ABSA has gradually deepened into two subtasks: aspect terms extraction (ATE) and aspect sentiment classification (ASC). Feature selection [19–21] and ATE are both information extraction tasks. However, unlike the dimensionality reduction of feature selection, ATE is a task of extracting the aspect terms mentioned in the

✉ Yan Xiang
50691012@qq.com

[1] Department of Information Engineering and Automation, Kunming University of Science and Technology, Kunming City, Yunnan Province, China

review text, which is essentially a sequence labeling task, and it has been extensively studied [4, 7, 8, 12, 15, 27]. ASC is a task of inferring the sentiment polarity (e.g., positive, neutral, negative) of a given aspect in a review. It is essentially a classification task, and the related researches are numerous [2, 5, 9, 17, 25, 29, 31]. However, most existing works address the two subtasks as two separate tasks, i.e., solve them one by one or only perform one task, which may not fully exploit the joint information from the two subtasks. Thus, the end-to-end aspect-based sentiment analysis (E2E-ABSA) has been proposed to eliminate this deficiency.

E2E-ABSA is a subtask of ABSA, which aims to solve ATE and ASC simultaneously in an end-to-end way. E2E-ABSA explores two research questions. First, how to improve the model's ability to solve the two subtasks of ATE and ASC. The works related to this question include how to capture the boundary information of the aspect terms in the ATE task and how to maintain the sentiment consistency of the aspect terms in the ASC task. Second, how to bridge ATE and ASC. As mentioned before, ATE and ASC are quite different tasks, ATE is a sequence labeling task, while ASC is a classification task. Getting the connection between them is the key to accomplish the aspect term-polarity co-extraction.

In this paper, we focus on solving the above two research questions, and present a model based on sentence constituent-aware attention mechanism ($C-ATT$). Specifically, we design $C-ATT$ to induce different constituents from the input sentence. In the $C-ATT$, we use a sentence constituent-aware coefficient to make the aspect term pay more attention to the opinion word that belong to the same constituent, which is helpful to accomplish the E2E-ABSA task. Moreover, the operation of the $C-ATT$ inducing constituents involves two important components. One is the neighboring attention mechanism, which determines whether two adjacent words in a sentence belong to a constituent. The other is the hierarchical constraint, which enables a sentence constituent to include more words. In the result induced by $C-ATT$, not only each word owns constituent boundary information, but also each aspect term and its corresponding opinion words are grouped into different constituents. In this way, making the aspect term impossible to locate other opinion words, then ensuring the sentiment consistency of individual words within the same aspect term. In addition, following the work of Li et al. [10], we dismiss the boundary of ATE and ASC by utilizing a unified tagging scheme and treat the two subtasks as a sequence labeling task, so as to accomplish the E2E-ABSA task.

The main contributions of this paper are as follows.

- We propose a novel framework $C-ATT$ to group the input sentence into different constituents and give the boundary information of each word in the sentence, which assists the recognition of the aspect terms.
- We also propose to maintain the sentiment consistency of the aspect term and enhance the interactions between aspect term and its corresponding opinion based on keeping the aspect term and its corresponding opinion within the same constituent, thereby achieving the improvement of sentiment classification.
- In the experiment, the proposed model outperforms baseline models on four benchmark SemEval datasets.

2 Related work

At present, the three main settings of ABSA are pipelined, joint, and collapsed manner respectively.

2.1 Pipeline methods

First of all, ATE is a sequence labeling task, while ASC is a classification task. Thus, they are traditionally treated as two separate tasks, and solved one by one in a pipeline manner. Mostly, the works related to it are either concentrated on ATE or ASC. Luo et al. [15] focused on ATE, and proposed a novel bidirectional dependency tree network to extract dependency structure features from the given sentences. Xu et al. [27] proposed a CNN model that employed two types of pre-trained embeddings for ATE, namely general-purpose embeddings and domain-specific embeddings. Later, Li et al. [12] introduced a masked sequence-to-sequence method for conditional augmentation of ATE. Additionally, Wang and Lu [23] focused on ASC, and developed a segmentation attention-based LSTM model, which employed a linear-chain conditional random field (CRF) layer to capture the structural dependencies between the aspect and the opinions. Xu et al. [28] presented a multiple CRFs based attention model that was capable of extracting aspect-specific opinion spans, then applied the extracted opinion features and contextual information to complete ASC.

2.2 Joint methods

Overall, the pipeline methods need to train two complex models separately, limiting their practical applications. Moreover, the pipeline methods may not fully exploit the joint information between the two subtasks. A possible idea to address these problems is to jointly trained a model to solve ATE and ASC simultaneously, i.e., the joint method. Specifically, bridging the difference between the two tasks by changing ASC to a sequence labeling task. Then, ATE and ASC have the same formulation.

Follow the above idea, Luo et al. [16] proposed a cross-shared RNN frame-work (DOER), in which a dual recurrent neural network was adopted to extract the respective representations of each task and a cross-shared unit was adopted to connected the representations. He et al. [6] proposed an interactive multi-task learning network (IMN) which was able to jointly learn multiple related tasks simultaneously at both the token level and the document level. Liang et al. [13] proposed a novel dependency syntactic knowledge augmented interactive architecture with multi-task learning for end-to-end ABSA.

2.3 Collapse methods

For the ABSA task, previous researchers have attempted two approaches to a more integrated solution, one is the mentioned joint method, the other is collapse method. Their differences lie in: On the one hand, the differences in tasks. The former treated ABSA as two sequence labeling tasks, while the later involved only one sequence labeling task. On the other hand, the differences in tags. As shown in Table 1, the former utilized two separate sets of tagging schemes, such as B and NEG, while the latter utilized a unified tagging scheme as B − NEG.

Because of the potential of a more integrated model is promising, the collapse method has received a lot of attention in recent years. Li et al. [10] presented a collapsed model, which involved two stacked RNN: One was adapted to predict sentiment labels, the other performed an auxiliary target boundary prediction. Li et al. [11] investigated the important impact of the contextualized embeddings on the collapsed model, and showed that even with a simple linear classification layer, these BERT-based architectures can still achieve considerable results. Wang et al. [26] proposed a hierarchical multi-task learning framework, which explicitly

Table 1 Tagging schemes used in the three methods

Sentence	the	after	sales	support	is	terrible
Pipeline &Joint	O	B	I	E	O	O
	O	NEG	NEG	NEG	O	O
Collapsed	O	B-NEG	I-NEG	E-NEG	O	O

leveraged task-related knowledge via the supervision of intermediate layers. Bie et al. [1] proposed a multitask multi-view network (MTMVN) architecture, which took the unified end-to-end ABSA as the main task with the two subtasks as auxiliary tasks. Mitchell et al. [18] and Zhang et al. [30] compared the pipelined, joint, and collapsed approaches for ABSA. They found that the joint and collapsed approaches are superior to the pipelined approaches.

2.4 Attention mechanism in aspect-based sentiment analysis

Wang et al. [24] appended the aspect vector into the input word vector, and concatenated the aspect vector with the sentence hidden representation to compute attention weights, and then proposed an ATAE-LSTM network for aspect-level sentiment classification. Tang et al. [22] regarded aspect vector as the input and context word vectors as the external memory to adaptively selected important evidences from memory through attention layers, further presented a deep memory network for ABSA. Fan et al. [3] proposed a multi-grained attention network (MGAN), which can capture the word-level interaction between aspect and context.

In this paper, we also emphasized the effectiveness of the attention mechanism in ABSA, and presented a constituent-aware attention mechanism to enhance the interactions between aspect terms and its corresponding opinions.

3 Model

In order to clearly illustrate our method, the symbols mentioned in the method are listed in Table 2.

The overall architecture of our model includes three components, as shown in Fig. 1. Firstly, we employ BERT component to calculate the contextualized representations $H^L = \{h_1^L, h_2^L, \cdots, h_N^L\}$ of the input sentence, where L is the number of transformer layers and

Table 2 Symbol description table

Symbol	Description	Symbol	Description
X	Input sentence	p	Probability
N	Sentence length	a	Weight sequence
Y	Tag	c	Weight
H	Sentence representation	l	Layer
Q	Query vector matrix	W	Weight matrix
K	Key vector matrix	b	Bias
C	Sentence constituent-aware coefficient matrix	K	The number of training instance
$C_{i,j}$	The element of C	M	The number of label category
q	Query vector	y_{ij}	Predicted label
k	Key vector	g_{ij}	Truth label

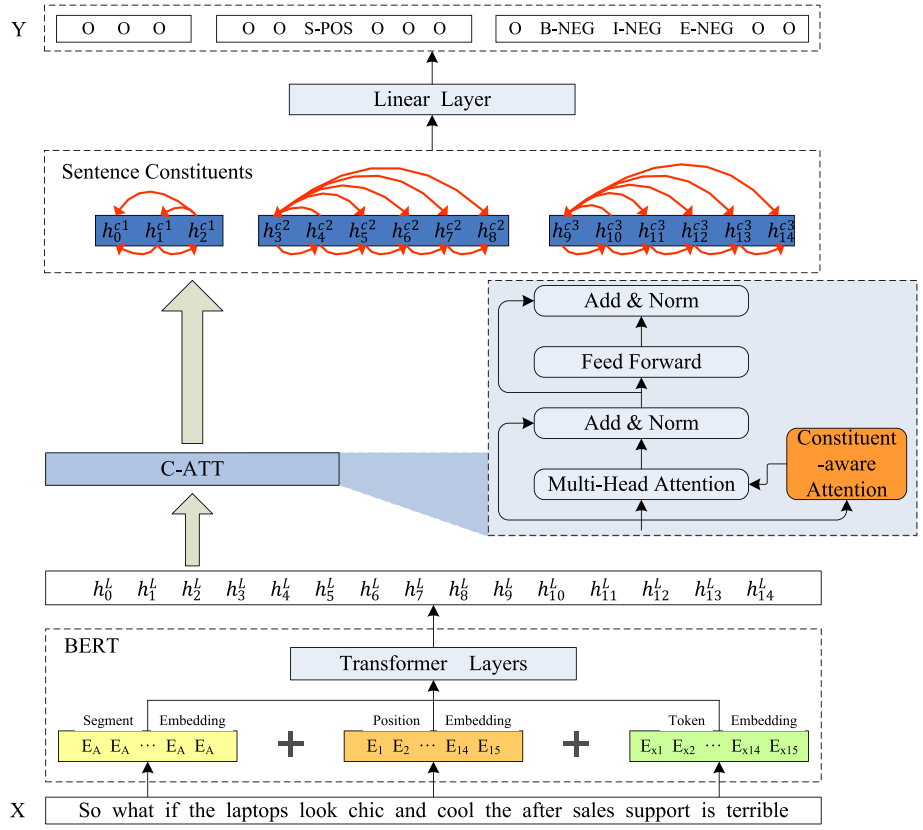

Fig. 1 The overall architecture of our model includes three components: BERT, C-ATT, and Linear layer

N is the sentence length. Secondly, H^L as the input of $C-ATT$ component to induce sentence constituents, and the output of $C-ATT$ is the sentence representations containing boundary and sentiment information. Finally, a simple linear classification layer is adopted to complete aspect term-polarity co-extraction.

3.1 Task formalization

Given an input sentence $X = \{x_1, x_2, \cdots, x_N\}$ of length N, the model aims to predict the out label Y corresponding to X. Following the setting of E2E-ABSA, we formulate it as a sequence labeling problem, and utilize a unified tagging Scheme Y = {B − POS, I − POS, E − POS, S − POS; B − NEG, I − NEG, E − NEG, S − NEG; B − NEU, I − NEU, E − NEU, S − NEU; O }. Where B, I, E denotes the beginning of aspect, the inside of aspect, and the end of aspect term. S denotes the single word aspect, and POS, NEG, NEU denotes the positive, negative, neutral sentiment respectively. Except O, each tag contains two parts of tagging information: the boundary of aspect term, and the sentiment polarity of the aspect term. A labeling example is illustrated in Table 3.

Table 3 A labeling example of E2E-ABSA

Sentence	the	laptop	look	chic	and	cool
Label	O	O	S-POS	O	O	O
Sentence	the	after	sales	support	is	terrible
Label	O	B-NEG	I-NEG	E-NEG	O	O

3.2 Embedding layer

The word embeddings trained by the traditional Word2Vec and Glove are static, which will cause the same words in different context to be mapped into the same embedding space, resulting in the problem of polysemous words. To a certain extent, the introduction of Embeddings from Language Models (ELMO) solved this problem. However, the feature extraction capability of LSTM-based ELMO is far weaker than that of Transformer-based extractor. Therefore, we choose Bidirectional Encoder Representations from Transformers (BERT) as the embedding layer to model the contextualized embeddings, which adopts the deep bidirectional Transformer layers as feature extractors. Thereby alleviating the polysemous word problem and extracting the deep semantic features of the context successfully.

As shown in Fig. 1, Given an input sentence $X = \{x_1, x_2, \cdots, x_N\}$, we pack its token embedding, position embedding and segment embedding as $H^0 = \{e_1, \cdots, e_N\}$, then sent it into BERT to calculate the contextualized representation $H^L = \{h_1^L, h_2^L, \cdots, h_N^L\}$.

3.3 C − ATT

3.3.1 Sentence constituent-aware coefficient

The traditional attention mechanism, scaled dot-product attention, has three matrices Q, K, and V as input, all of them are consist of vectors with dimension d. First, the word embeddings are linearly projected to obtain the initial query vectors q, key vectors k, and value vectors v respectively. Then, the dot-product of q and all k is calculated as the weight of v, and the value of v is constantly updated. Finally, each of the q, k, and v vectors are packed together to form the Q, K, and V matrix. That is, Q is the query vector matrix, K is the key vector matrix, and V is the value vector matrix. It calculates the dot products of the query with all keys, divides each by $\sqrt{d}$, and applies a softmax function to obtain the weights on the values. The specific process is as follows:

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d}}\right)V \tag{1}$$

When large value of d, the dot product QK^T grows large in magnitude, pushing the softmax function into the region where it has extremely small gradients. So, the dot product QK^T is scaled by a scaling factor $\sqrt{d}$ to counteract this effect.

Compared with the traditional attention mechanism, $C - ATT$ introduces a sentence constituent-aware coefficient that gives higher attention weights to words belonging to the same constituent, while lower attention weights to other words. As depicted in Fig. 1, in the Sentence Constituents module, the red arrows represent Self-Attention, and the bule blocks indicate different sentence constituents. Words in the same constituent pay attention to each

other, but words across constituents are constrained to hardly attend to each other. The implementation process is shown in formula (2).

$$C\text{-}ATT(Q,K,C) = C \odot softmax\left(\frac{QK^T}{\sqrt{d}}\right) \tag{2}$$

Where C is a N by N matrix, $\odot$ is the element-wise multiplication, the matrices Q, K are respectively composed of the query vectors and key vectors corresponding to $H^L = \{h_1^L, h_2^L, \cdots, h_N^L\}$ obtained by the embedding layer, and d is the dimension of the query vectors and key vectors. In formula (2), the element of C is $C_{i,j}$, it is the probability that word w_i and w_j belong to a same constituent. When $C_{i,j}$ has small value, it means that the words at position i and j are in different constituents, so hardly attend to each other.

3.3.2 Adjacent attention mechanism and hierarchical constraints

As mentioned before, the difference between the proposed $C-ATT$ and the traditional attention lies in the sentence constituent-aware coefficient $C_{i,j}$. The following section will detail the calculation process of $C_{i,j}$.

First of all, for an input sentence X, we respectively calculate the probability of the word w_i and its adjacent words w_{i-1}, w_{i+1} belonging to a constituent. This process is similar to the traditional attention, so it is denoted the adjacent attention mechanism. In our model, the adjacent attention mechanism determines which words in the sentence belong to the same constituent, while $C_{i,j}$ determines which words in the sentence are closely connected and weights them. The adjacent attention weights are computed by the following formulas.

$$p_{i,i-1} = \frac{q_i k_{i-1}}{\sqrt{d_{model}}} \tag{3}$$

$$p_{i,i+1} = \frac{q_i k_{i+1}}{\sqrt{d_{model}}} \tag{4}$$

Where q_i is a query vector of w_i with d dimension, and k_{i-1}, k_{i+1} are the d-dimensional key vectors of the two adjacent words w_{i-1}, w_{i+1} of w_i. The scaling factors in formulas (3) and (4) are different from formula (2).

In order to prevent our model from linking all words together and assigning them to the same constituent, we deal it with the constraint of the softmax operation, as follows:

$$p_{i,j} = \boldsymbol{softmax}\left(p_{i,i-1}, p_{i,i+1}\right) \tag{5}$$

Where $p_{i,i-1}$, $p_{i,i+1}$ are calculated by formulas (3) and (4) respectively, then $p_{i,i-1} + p_{i,i+1} = 1$ is established. Additionally, $p_{i,j}$ represents the tendency that w_i and w_j belong to the same constituent. When $p_{i,i-1} > p_{i,i+1}$, w_i belongs to the same constituent with its left neighbor w_{i-1}, otherwise it belongs to a constituent with its right neighbor w_{i+1}.

So far, whether two adjacent words in a sentence belong to a constituent or not can be determined by formula (5), and Layer 3 in Fig. 2 has shown this visually. And it is observed

that $p_{i,i+1}$ and $p_{i+1,i}$ may have different values because the different query vectors they have, such as $p_{2,3} \neq p_{3,2}$ in Fig. 2. We average the two attention weights as the following formula.

$$a_i = \sqrt{p_{i,i+1} \times p_{i+1,i}} \tag{6}$$

Thus, the weight sequence $a = \{a_1, \cdots, a_i, \cdots, a_N\}$ is obtained based on the adjacent attention mechanism, where N is the sentence length, a_i is the probability that the word w_i and its right neighbor w_{i+1} are in a constituent.

For each word w_i in the input sentence, it either belongs to the same constituent with its left neighbor w_{i-1} or its right neighbor w_{i+1}. Hence, a sentence constituent can only contain two words for a maximum. In order to group more words into the same constituent, we introduced hierarchical constraints. Specifically, $C-ATT$ of each layer shares a same adjacent attention weight sequence a, and the hierarchical constraints are applied to restrict the constituents in layer $l+1$ to always be larger than the current layer l. Then constituents in the lower layer merge into larger one in the higher layer. For instance, in Fig. 2, a constituent contains up to two words in Layer 1, while more words in Layer 2. The definition of the hierarchical constraints is as follows:

$$c_i^l = c_i^{l-1} + \left(1 - c_i^{l-1}\right) a_i^l \tag{7}$$

Where c_i^{l-1} indicates the adjacent attention weight sequence with hierarchical constraints in the layer $l-1$, and a_i^l is calculated by formula (6) in the current layer l. Formula (7) ensures that once two words belong to the same constituent in the lower layer, they would still belong to the same constituent in the higher layer. Initially, as shown in Fig. 2, different words are regarded as different constituents, so c_i^0 is initialized as 0.

After obtaining the adjacent attention weight sequence $c_i^l = \{c_1^l, \cdots, c_i^l, \cdots, c_N^l\}$ with hierarchical constraints in the layer l, we calculate the sentence constituent-aware coefficient $C_{i,j}(c_{i \le k < j})$ in the layer l by formula (8).

$$C_{i,j} = \prod_{k=i}^{j-1} c_k \tag{8}$$

Intuitively, we choose multiplication operation to calculate $C_{i,j}$. For two small probabilities, the multiplication between them becomes smaller, which can improve the sensitivity of our

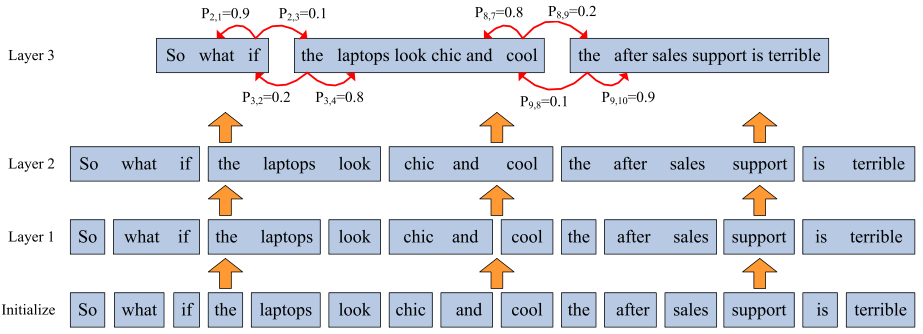

Fig. 2 The adjacent attention mechanism induces constituents from sentence

model to the breakpoints between words. In implementation, to avoid probability vanishing, formula (8) is updated as follows:

$$C_{i,j} = e^{\sum_{k=i}^{j-1} log(c_k)} \tag{9}$$

3.4 E2E-ABSA layer

As depicted in Fig. 1, $C - ATT$ accepts matrices Q and K as input, which are formed by learning linear projections of $H^L = \{h_1^L, h_2^L, \cdots, h_N^L\}$, then induces sentence constituents, finally outputs a sentence representation $H^C = \{h_1^{c1}, h_2^{c1}, \cdots, h_N^{cn}\}$ containing the boundary and sentiment information. We directly put the obtained representation H^C into a linear layer to predict the E2E-ABSA labels. The calculation process of the linear layer as a classifier is as follows:

$$H^C = C\text{–}ATT(Q, K) \tag{10}$$

$$P(y_i|x_i) = \boldsymbol{softmax}(W_o H^C + b_o) \tag{11}$$

Where W_0 and b_0 are the learnable parameters, y_i is the predicted label corresponding to x_i.

3.5 Model training

In this paper, we use the cross-entropy function as loss to train our model.

$$Loss = -\frac{1}{K}\sum_{i}^{K}\sum_{j}^{M} g_{ij}\left(\log\_soft\max\left(p\left(y_{ij}|x_i\right)\right)\right) \tag{12}$$

Where K is the total number of training instances, M is the total number of label categories, and we set $M = 13$. y_{ij} is predicted label by model, g_{ij}denotes the one-hot encoding of the truth labels.

4 Experiments

4.1 Datasets

We run experiments on four benchmark datasets to test the effectiveness of our model. The statistics are summarized in Table 4. Laptop14 is taken from the Laptop dataset published by SemEval challenge 2014 Task4. Rest14, Rest15 and Rest16 are taken from the Restaurant datasets respectively published by SemEval challenge 2014 Task4, SemEval Challenge 2015 task 12, SemEval Challenge 2016 task 5. In addition, Sent is the total number of sentences, while Aspect is the number of aspect terms.

Table 4 Statistics of datasets

Dataset		Train	Dev	Test	Total
Laptop14	Sent	2741	304	800	3845
	Aspect	*2041*	*256*	*634*	*2931*
Rest14	Sent	2736	304	800	3840
	Aspect	*3266*	*329*	*1120*	*4715*
Rest15	Sent	1183	130	685	1998
	Aspect	*1079*	*130*	*547*	*1756*
Rest16	Sent	1799	200	676	2675
	Aspect	*1589*	*167*	*622*	*2378*

4.2 Experimental settings

We use the deep learning framework Pytoch1.2.0 based on Python 3.7.3. in our experiment. Our code is open-source and available at https://github.com/Kmustxz109/E2E-ABSA/tree/master. In our experiments, the "bert-base-uncased" pre-training model is applied to word embedding initialization. The number of layers of its component transformer $L = 12$, and the hidden size $dim_h = 768$. The hidden size of the proposed $C - ATT$ is consistent with transformer, in which $d = 768$. We train our model in batches, where batch_size is set to 16, epoch is dynamically adjusted according to the data size, and the number of steps is 1500. After training 1000 steps, we select model on the development set for every 100 steps with the best micro-averaged F1 score for evaluation. We use Adam optimizer with learning rate $5e-5$. Dropout with $p = 0.1$ is employed to prevent overfitting. Following these settings, we train 5 models with different random seeds and report the average results.

4.3 Baseline models

To validate the performance of the proposed model $C - ATT$ on the task, a comparative experiment with some appropriate baseline models is conducted. Models can be classified to two groups by whether using BERT as embedding layer. One is the four classic models for the E2E-ABSA task, and the other is BERT-based models.

Base model + BG + SC + OE This model was proposed by Li et al. [10], which completed the E2E-ABSA task by a collapsed way. It consists of a base model and three components. The base model, i.e., two stacked recurrent neural networks, where the upper one solved the E2E-ABSA and the lower one solved the ATE. The three components were three methods to connect the two base networks, namely, boundary guidance (BG), sentiment consistency (SC) and opinion-enhanced (OE).

Doer This model was proposed by Luo et al. [16], which completed the E2E-ABSA task by a joint way. It involves a stacked dual RNNs, one stacked RNN for ATE, and the other for ASC. Moreover, a cross-shared unit is used to consider the relationship between them.

Lm-LSTM-CRF This model was proposed by Liu et al. [14]. It is a competitive model in several sequence tagging tasks. Li et al. [10] adopted it to solve the E2E-ABSA and reported the tagging results based on the unified tagging scheme.

IMN This model was proposed by He et al. [6], which treated the E2E-ABSA as a multi-task learning, and introduced a message passing architecture to connect the different tasks.

BERT + E2E-ABSA layer These models are BERT-based models proposed by Li et al. [11], which all completed the E2E-ABSA task by a collapsed way. After obtaining the BERT representations, Li et al. investigated several different designs for the E2E-ABSA layer, namely, linear layer, recurrent neural network (GRU), self-attention network (SAN), transformer layer (TFM), and conditional random field (CRF). Further, they reported the results.

4.4 Results and analysis

4.4.1 Evaluation metric

The evaluation metric in our experiments is the micro-averaged F1 score, which is defined as the following formula.

$$micro\_F1 = \frac{2 \times micro\_P \times micro\_R}{micro\_P + micro\_R} \tag{13}$$

Where $micro\ \_\ P$ is precision, its definition is as formula (14). And $micro\ \_\ R$ is recall, which is defined as formula (15).

$$micro\_P = \frac{\sum_{i=1}^{n} TP_i}{\sum_{i=1}^{n} TP_i + \sum_{i=1}^{n} FP_i} \tag{14}$$

Where n represents the total number of instances, TP represents the number of true positive instances, FP represents the number of false positive instances.

$$micro\_R = \frac{\sum_{i=1}^{n} TP_i}{\sum_{i=1}^{n} TP_i + \sum_{i=1}^{n} FN_i} \tag{15}$$

Where FN represents the number of false negative instances.

4.4.2 Main results

In order to validate the effectiveness of the proposed $C - ATT$ on the E2E-ABSA task, a comparative experiment was conducted with the baseline models described in the previous section. The results are shown in Table 5. In Table 5, it should be noted that the Restaurant dataset used in previous works was created by concatenating the officially released datasets, i.e., Rest14, Rest15, and Rest16 respectively. Since the concatenating Restaurant dataset will exist overlap between training data and test data, we keep the official datasets division of Rest14, Rest15, and Rest16 in our experiments.

From Table 5, we can observe that, the BERT-based models outperform the existing models on all datasets. Even the most powerful existing model DOER is beaten by the

Table 5 Micro-F1 score (%) comparison of all models for the task. Best performances are in bold

Model		Dataset			
		Laptop14	Rest14	Rest15	Rest16
Existing models	Base model + BG+SC+OE	57.90	69.80		
	DOER	60.35	72.78		
	LM-LSTM-CRF	56.19	66.38		
	IMN	58.37	69.54	59.18	–
BERT models	BERT + Linear	60.43	73.22		
	BERT + GRU	61.12	73.24		
	BERT + SAN	60.49	74.72		
	BERT + TFM	60.80	74.41		
	BERT + CRF	60.78	74.06		
	BERT+ Linear	*60.43*	*72.61*	*60.29*	*69.67*
	BERT+ GRU	***61.12***	*73.17*	*59.60*	*70.21*
	BERT+ SAN	*60.49*	*73.68*	*59.90*	***70.51***
	BERT+ TFM	*60.80*	*73.98*	*60.24*	*70.25*
	BERT+ CRF	*60.78*	*73.17*	*60.70*	*70.37*
	BERT+C-ATT	*60.62*	***74.80***	***61.70***	*70.35*

BERT+Linear. BERT+Linear leads by 0.08% on Laptop14 and 0.44% on Restaurant, showing the necessity of employing BERT as embedding layer. This can be explained by adopting the traditional Word2Vec or Glove as embedding layer can only generate a context-independent representation for each word, while BRET can generate a contextualized representation. This also verifies that using the pre-trained language model BERT to obtain contextualized embeddings can greatly improve the performance of our model on the E2E-ABSA task.

The italicized results in Table 5 report that BERT+Linear method gets the worst performance of all BERT-based baseline methods, which performs poorly with a micro-F1 score of 60.43% on Laptop14, 72.61% on Rest14, 60.29% on Rest15, and 69.67% on Rest16. The possible reason is that other BERT-based models adopted GRU, SAN, TFM, and CRF respectively to build the task-specific token representations, while BERT+Linear did not.

From Table 5, firstly, we can see that the proposed model BERT+ $C-ATT$ achieves the best performance on Rest14 and Rest15 among all baselines. And we can observe that BERT+ $C-ATT$ all significantly exceed BERT+Linear. Specifically, BERT+ $C-ATT$ improves the performance by 0.19%, 2.19%, 1.41%, and 0.68% on the four benchmark datasets respectively, which fully demonstrates the effectiveness of our model. That is, BERT+Linear did not contain any auxiliary information to complete E2E-ABSA, while BERT+$C-ATT$ employed $C-ATT$ to aware constituents from the input sentence, giving the boundary information of each word in the sentence and making the aspect term pay more attention to the opinion that belong to the same constituent. It is reasonable for the proposed model achieve overwhelming performance. Secondly, compared with BERT+GRU, BERT+SAN, and BERT+TFM, BERT+ $C-ATT$ always outperforms them in three sets of experiments. Specifically, BERT+$C-ATT$ outperforms BERT+SAN by 0.13%, 1.12%, 1.80% on Laptop14, Rest14, Rest15 respectively, only marginally worse by 0.16% on Rest16. The possible explanation is that $C-ATT$ maintains the sentiment consistency of aspect terms by keeping the aspect term and its corresponding opinion within the same constituent, effectively alleviating the misjudgment of sentiment polarity of different aspect terms in a sentence. Additionally, compared with BERT+CRF, BERT+ $C-ATT$ exceeds it in half of experiments, exceeding 1.63% on Rest14 and 1% on Rest15. These comparisons suggest that the operation of inducing constituents from

input sentences is very helpful for the E2E-ABSA task. What's more, some syntactic information based methods depends on the syntactic parser tool, while our work directly uses the $C-ATT$ to induce constituents from input sentence. The performance of the proposed model does not depend on any syntax parser tool.

4.4.3 Experiment of hierarchical constraints

As mentioned in Section 3.3.2, we adopt hierarchical constraints to ensure that $C-ATT$ includes appropriate words when inducing the sentence constituents. Too lower layers of $C-ATT$, different words are regarded as different sentence constituents, leading to the aspect term and its corresponding opinion divided into different constituents. On the contrary, when layers of $C-ATT$ exceed a certain layer, all words are grouped into the same constituent, making $C-ATT$ behave the same as traditional attention mechanism. Therefore, we discuss the sensitivity of $C-ATT$ layers to sentence constituents in this section. The experiment is conducted on four benchmark datasets by varying the number of layers from 2 to 10 with the step of 2, and the experimental results are shown in Fig. 3.

As depicted in Fig. 3, L refers to the number of layers of $C-ATT$. Overall, we can first observe that $C-ATT$ consistently achieves the best performance on Rest14 and performs with 74.38% on average. Firstly, the achievement should be related to the amount of data. As the statistics in Table 4, Rest14 has a total of 4715 data, far exceeding the rest of datasets, which shows that increasing the amount of data can improve the performance of the model. Secondly, the achievement can be explained by the semantic comprehension of our model. As displayed in Table 6, we counted the proportions of the four parts-of-speech in our datasets, namely, noun, verb, adjective, and adverb. From Table 6, we can see that more than half of the words in Rest14 and Rest16 are nouns and adjectives. This implies that our model pays more attention to nouns and adjectives during the training process, helping our model gains better results on Rest14 and Rest16. In addition, this also naturally follows the rule that most aspect terms are nouns and most opinions are adjectives in the E2E-ABSA, demonstrating the superiority and rationality of employing the proposed model to deal with the E2E-ABSA task.

According to the data trend in Fig. 3, the micro-F1 of $C-ATT$ has a peak at state $L=4$ on almost all the datasets except for the Rest16. Empirically, $L=4$ is the best choice for our work. Besides, the peak of the micro-F1 on four datasets all appears after $L>2$. When the number of $C-ATT$ layers is too small, different words are regarded as different sentence

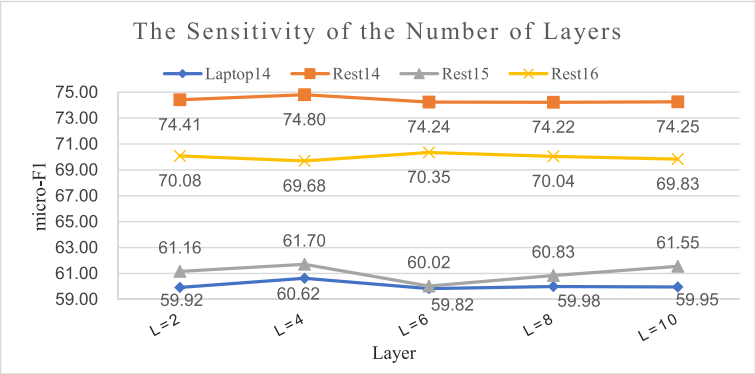

Fig. 3 Micro-F1 score of different $C-ATT$ layers on four benchmark datasets

Table 6 Part-of-speech analysis

dataset	noun Num. (Pro.)	adv Num. (Pro.)	adj Num. (Pro.)	verb Num. (Pro.)	total Num. (Pro.)
Laptop14	2329 (38%)	870 (14%)	992 (16%)	1989 (32%)	6180 (100%)
Rest14	2658 (**39%**)	930 (14%)	1359 (**20%**)	1937 (28%)	6884 (100%)
Rest15	2036 (37%)	769 (14%)	920 (17%)	1713 (32%)	5438 (100%)
Rest16	2116 (**40%**)	733 (14%)	952 (**18%**)	1499 (27%)	5330 (100%)

constituents, weakening the connections between aspect terms and opinions. It demonstrates that increasing the number of layers is beneficial. Instead, the performance of $C - ATT$ stops growing or even decreases when $L \geq 8$. It proves that the words are all grouped into the same constituent with too many layers, and increasing the layer number will no contribute to our model. At this time, $C - ATT$ distracts the attention of aspect term to opinion, then decreasing the performance, which is similar to the traditional attention mechanism.

4.4.4 Visualization of sentence constituent-aware coefficient

For more intuitive understanding of the behavior of the $C - ATT$, we further perform a visualization of the sentence constituent-aware coefficient matrix. We take a sentence from Reat16 as an example, and the number of layers is set to 4. The example sentence involved is "The duck confit is always amazing and the foie gras terrine with figs was out of this world". The visualization result of the corresponding matrix C is shown in Fig. 4, where each color cell denotes C_{ij}. The result demonstrates that: Firstly, $C - ATT$ can effectively perceive sentence breakpoints. From Fig. 4, we can observe that the attention weight of the word "and" with the word "amazing" are extremely small, so $C - ATT$ captured a breakpoint between the two words. Secondly, $C - ATT$ allows the words in the same constituent pay more attention to each other. As presented in Fig. 4, the words "is", "always", and "amazing" belong to a same constituent, naturally all of them with high weights. Finally, $C - ATT$ does contribute to the

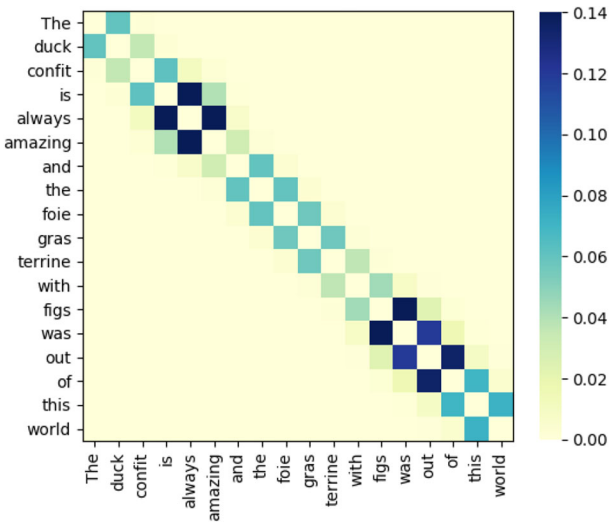

Fig. 4 Visualization of sentence constituent-aware coefficient matrix

E2E-ABSA task. The very high weight values of "amazing" and "out of this world" prove $C - ATT$ captured the opinions correctly, which benefits the E2E-ABSA.

5 Conclusion

This paper focused on E2E-ABSA task and treated the two subtasks of ABSA as a sequence labeling task, then completed the aspect term-polarity co-extraction. For this task, we proposed a novel model based on $C - ATT$. First, we adopted the pre-trained model BERT to gain the contextualized word embedding. Then, $C - ATT$ was employed to induce constituents from sentence. Finally, a simple linear layer was taken as a classifier to predict the unified tagging scheme. Comparative experiment results showed that the proposed model $C - ATT$ outperforms the baseline models, which proves the effectiveness of inducing constituents from input sentence. In the future work, we will adjust the proposed model to fit sentiment triples extraction, including aspect, opinion and sentiment.

Acknowledgments This work was supported by National Natural Science Foundation of China (62162037) and General Projects of Basic Research in Yunnan Province (202001AT070047).

References

1. Bie Y, Yang Y (2021) A multitask multi-view neural network for end-to-end aspect-based sentiment analysis. Big Data Mining and Analytics 4(3):195–207
2. Chen P, Sun Z, Bing L, Yang W (2017) Recurrent attention network on memory for aspect sentiment analysis. In: EMNLP, pp 452–461
3. Fan F, Feng Y, Zhao D (2018) Multi-grained attention network for aspect-level sentiment classification. In: EMNLP, pp 3433–3442
4. He R, Lee WS, Ng HT, Dahlmeier D (2017) An Unsupervised Neural Attention Model for Aspect Extraction. In: ACL, pp 388–397
5. He R, Lee WS, Ng HT, Dahlmeier D (2018) Exploiting document knowledge for aspect-level sentiment classification. In: ACL, pp 579–585
6. He R, Lee WS, Ng HT, Dahlmeier D (2019) An interactive multi-task learning network for end-to-end aspect-based sentiment analysis. In: ACL, pp 504–515
7. Li X, Lam W (2017) Deep multi-task learning for aspect term extraction with memory interaction. In: EMNLP, pp 2886–2892
8. Li X, Bing L, Li P, Lam W, Yang Z (2018a) Aspect term extraction with history attention and selective transformation. In: IJCAI, pp 4194–4200
9. Li X, Bing L, Lam W, Shi B (2018b) Transformation networks for target-oriented sentiment classification. In: ACL, pp 946–956
10. Li X, Bing L, Li P, Lam W (2019a) A unified model for opinion target extraction and target sentiment prediction. In: AAAI, pp 6714–6721
11. Li X, Bing L, Zhang W, Lam W (2019b) Exploiting BERT for End-to-End Aspect-based Sentiment Analysis. In: EMNLP, pp 34–41
12. Li K, Chen C, Quan X, Ling Q, Song Y (2020) Conditional Augmentation for Aspect Term Extraction via Masked Sequence-to-Sequence Generation. In: ACL, pp 7056–7066
13. Liang Y, Meng F, Zhang J, Xu J, Chen Y, Zhou J (2021) A dependency syntactic knowledge augmented interactive architecture for end-to-end aspect-based sentiment analysis. Neurocomputing 454:291–302
14. Liu L, Shang J, Ren X, Frank FX, Gui H, Peng J, Han J (2018) Empower sequence labeling with task-aware neural language model. In: AAAI, pp 5253–5260
15. Luo H, Li T, Liu B, Wang B, Unger H (2018) Improving aspect term extraction with bidirectional dependency tree representation. arXiv:1805.07889

16. Luo H, Li T, Liu B, Zhang J (2019) DOER: Dual cross-shared RNN for aspect term-polarity co-extraction. In: ACL, pp 591–601
17. Ma D, Li S, Zhang X, Wang H (2017) Interactive attention networks for aspect-level sentiment classification. In: IJCAI, pp 4068–4074
18. Mitchell M, Aguilar J, Wilson T, Van Durme B (2013) Open domain targeted sentiment. In: EMNLP, pp 1643–1654
19. Rostami M, Berahmand K, Forouzandeh S (2020) A novel method of constrained feature selection by the measurement of pairwise constraints uncertainty. J Big Data 7:83
20. Rostami M, Berahmand K, Forouzandeh S (2020) A novel community detection based genetic algorithm for feature selection. CoRR abs/2008.03543
21. Rostami M, Berahmand K, Nasiri E, Forouzandeh S (2021) Review of swarm intelligence-based feature selection methods. Eng Appl Artif Intell 100:104210
22. Tang D, Qin B, Liu T (2016b) Aspect level sentiment classification with deep memory network. In: EMNLP, pp 214–224
23. Wang B, Lu W (2018) Learning latent opinions for aspect-level sentiment classification. In: AAAI, pp 5537–5544
24. Wang Y, Huang M, Zhao L et al (2016) Attention-based lstm for aspect-level sentiment classification. In: EMNLP, pp 606–615
25. Wang S, Mazumder S, Liu B, Zhou M, Chang Y (2018) Target-sensitive memory networks for aspect sentiment classification. In: ACL, pp 957–967
26. Wang X, Xu G, Zhang Z, Jin L, Sun X (2021) End-to-end aspect-based sentiment analysis with hierarchical multi-task learning. Neurocomputing 455:178–188
27. Xu H, Liu B, Shu L, Yu PS (2018) Double embeddings and cnn-based sequence labeling for aspect extraction. In: ACL, pp 592–598
28. Xu L, Bing L, Lu W, Huang F (2020) Aspect sentiment classification with aspect-specific opinion spans. In: EMNLP, pp 3561–3567
29. Xue W, Li T (2018) Aspect Based Sentiment Analysis with Gated Convolutional Networks. In: ACL, pp 2514–2523
30. Zhang M, Zhang Y, Vo DT (2015) Neural networks for open domain targeted sentiment. In: EMNLP, pp 612–621
31. Zheng L, Wei Y, Yu Z, Zhang X, Li X (2019) Exploiting coarse-to-fine task transfer for aspect-level sentiment classification. In: AAAI, vol 33, pp 4253–4260