Complex & Intelligent Systems (2023) 9:537–563
https://doi.org/10.1007/s40747-022-00818-2

ORIGINAL ARTICLE

Aspect term extraction via information-augmented neural network

Ning Liu[1,2] · Bo Shen[1,2]

Received: 12 November 2021 / Accepted: 24 June 2022 / Published online: 22 July 2022

Abstract

Aspect term extraction (ATE) aims at identifying the aspect terms that are expressed in a sentence. Recently, Seq2Seq learning has been employed in ATE and significantly improved performance. However, it suffers from some weaknesses, such as lacking the ability to encode the more informative information and integrate information of surrounding words in the encoder. The static word embeddings employed in ATE fall short of modeling the dynamic meaning of words. To alleviate the problems mentioned above, this paper proposes the information-augmented neural network (IANN) which is a novel Seq2Seq learning framework. In IANN, a specialized neural network is developed as the key module of the encoder, named multiple convolution with recurrence network (MCRN), to encode the more informative information and integrate information of surrounding words in the encoder. The contextualized embedding layer is designed to capture the dynamic word sense. Besides, the novel AO ({Aspect, Outside}) tags are proposed as the less challenging tagging scheme. A lot of experiments have been performed on three widely used datasets. These experiments demonstrate that the proposed IANN acquires state-of-the-art results and validate that the proposed IANN is a powerful method for the ATE task.

Keywords Aspect term extraction · Aspect-based sentiment analysis · GRU · CNN · Deep learning

Introduction

In natural language processing (NLP), one of the hot research fields is aspect-based sentiment analysis (ABSA) which is a fine-grained subtask of sentiment analysis [1]. One of the important tasks in ABSA is the ATE task, which aims at identifying aspect terms (or targets) that are expressed in a sentence [2–4]. Aspect terms can be explicit or implicit in a sentence. Explicit aspect terms are explicitly expressed in a sentence, implicit aspect terms are implicitly expressed in a sentence. To give an example, in Fig. 1, the ATE task aims at detecting two explicit aspect terms "food" and "service" in sentence 1. In another sentence "The camera is too expensive", the target of the ATE task is the implicit aspect term "price" because "expensive" is used to describe the aspect term "price" of a camera. This paper focuses on the explicit ATE task. Not only in the academic community but also in the business community, the ATE task has been paid much attention. For example, in e-commerce platform, people are more interested in knowing different attributes toward the specific product. Besides, a company can automatically detect these attributes of a product or service from a customer's review by using the ATE technique. Also, business enterprises can further enhance their products by analyzing customer reviews using the ATE technique.

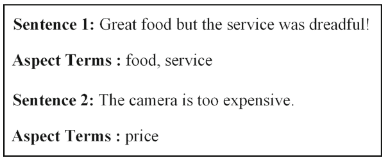
In conventional statistic machine learning, previous works mainly employ hidden Markov model (HMM) [5] and conditional random fields (CRF) [6] in the ATE task. The hand-designed features have a great influence on the performance of these models. The process of feature design is also called feature engineering, which not only consumes a lot of manpower and material resources but also requires extra expert knowledge. Neural networks have achieved good performance in many areas, the ATE task is no exception. Deep neural network-based methods have dominated the ATE task and achieved good performance in the ATE task [7]. Deep learning can automatically extract high-level and the more appropriate features which are the more suitable representations for the end task. These neural network-based methods also treat the ATE task as the sequence labeling task. However, the overall semantics of a sentence cannot be well

✉ Bo Shen
bshen@bjtu.edu.cn

Ning Liu
17111016@bjtu.edu.cn

[1] School of Electronic and Information Engineering, Beijing Jiaotong University, Beijing 100044, China

[2] Key Laboratory of Communication and Information Systems, Beijing Municipal Commission of Education, Beijing, China



Sentence 1: Great food but the service was dreadful!

Aspect Terms : food, service

Sentence 2: The camera is too expensive.

Aspect Terms : price

Fig. 1 The ATE task

captured by these neural network-based sequence labeling methods in the ATE task. Besides, they have a weak capacity in modeling dependencies between labels.

Recently, some works regard the ATE task as a sequence to sequence (Seq2Seq) learning task, the Seq2Seq learning can naturally tackle the problems of the sequence labeling-based models [8]. However, Seq2Seq learning employed in the ATE task still suffers from some weaknesses. Firstly, the encoder in the Seq2Seq-based methods mostly employs long-short term memory (LSTM) [9] or gated recurrent unit (GRU) [10] in the ATE task. Only employing LSTM or other methods is powerless in encoding the more informative information in the encoder. Besides, since LSTM processes the inputs in sequence order, it can’t focus on the surrounding words toward a specific word. Hence, the encoder is also weak in integrating information of the surrounding words toward a specific word and detecting the local features. Although the attention mechanism [11] can be used in the encoder to focus on the surrounding words, it may introduce some noises which can damage model performance [12]. Moreover, the word embeddings employed in the ATE task are static word embeddings, which means the word embeddings are fixed in different contexts. However, the meaning of a word in human language is dynamic and depends on its context. The static word embeddings fall short of modeling the dynamic word sense.

This paper proposes an IANN which is a novel Seq2Seq learning framework and can alleviate the above-mentioned inadequacies. In IANN, a novel and specialized neural network is developed as the key module of the encoder, named MCRN, which combines multiple convolution with recurrence operations. Combining multiple convolution with recurrence operations in an appropriate way can improve the ability of the encoder. The multiple convolution operations of MCRN can integrate information of surrounding words toward a specific word and detect the local features. Instead of using vanilla RNN, the bidirectional GRU is employed in MCRN which can not only model the sequence information but also capture the bidirectional long-distance dependencies. Therefore, MCRN can encode more informative information than the encoder that consists of a single type neural network and integrate information of the surrounding words toward a specific word. The MCRN model can be formed into a single or multi-layer architecture encoder. The multi-layer encoder learns the more informative higher-order features, which are more appropriate representations for the ATE task to some extent. The decoder of IANN consists of the unidirectional GRU, it is used to decode the encoding representations to predict the labels of each word. The contextualized embedding layer is developed to model the dynamic word sense and generate contextualized word embeddings by employing bidirectional encoder representation from transformers (BERT) [13]. Besides, novel dual AO tags are proposed as the less challenging tagging scheme in the ATE task. A lot of experiments demonstrate that the proposed IANN gets better performance than other state-of-the-art (SOTA) baselines for the ATE task. Moreover, these experiments also verify that the proposed IANN is independent of dataset type. In general, this work makes the following contributions.

1. This paper proposes an IANN which is a novel Seq2Seq learning framework and can alleviate the weaknesses of the previous Seq2Seq-based model and static word embeddings in the ATE task.
2. This paper designs a specialized hierarchical neural network as a key module of the encoder, named MCRN which combines multiple convolution with recurrence operations, to encode the more informative information. The MCRN model can not only integrate information of surrounding words toward a specific word and detect the local features but also model the sequence information and capture the long-distance dependencies.
3. This paper proposes the multi-layer MCRN as the multi-layer encoder, which can learn the more informative higher-order features, which are more appropriate representations for the ATE task.
4. This paper proposes the novel dual tags, named AO tags, as the less challenging tagging scheme.
5. A lot of experiments have been performed on three datasets which involve many sources and domains. These experimental results demonstrate that IANN obtains SOTA performance and verify that the proposed IANN is a powerful method for the ATE task. Besides, these experiments also validate that IANN is independent of dataset type.

Related work

ABSA is a promising technique in artificial intelligence and it can be used in many realistic scenarios [14, 15], ABSA is first developed by Hu and Liu [16]. In the last few years, ABSA has been widely researched [17]. Three subtasks which are aspect level sentiment classification (ALSC), ATE, and opinion term extraction (OTE) constitute ABSA [18]. The ATE task aims at identifying aspect terms (or targets) that are expressed in a sentence, and the ALSC task tries to detect sentiment polarities toward aspect terms that are given in the specific sentence. This paper pays more attention to the ATE task. The comprehensive surveys referring to the ATE task can be found in Rana and Cheah's survey [19]. Not only different aspect extraction techniques and approaches are discussed, but they also cover detailed explanations and precise comparisons. Tubishat et al. [20] focus on implicit aspect extraction. They provided comprehensive comparison analysis and taxonomy in implicit aspect extraction. The related and complicated problems are also reported in their research.

The ATE task is first studied by Hu and Liu [16]. They developed several rules to deal with ATE. They employed an association rule miner [21] to detect candidate aspect terms and calculated the frequency of these aspect terms, they filter out some aspect terms based on the given threshold. Opinion terms are used to find the non-frequent aspect terms by the dependency relations. To improve the performance of the rule-based method, Popescu and Etzioni [22] employed pointwise mutual information (PMI) to identify aspect terms. To mitigate the vulnerability of syntactic rule-based approaches, the TF-RBM model is developed by Rana and Cheah [23]. The explicit aspect terms are discovered by the sequential patterns-based rules. Then the concept extraction is performed in TF-RBM. However, these rule-based methods suffer from some shortcomings, such as needing elaborate rules, and the form of the sentence affects the performance of the rules.

Many conventional statistical machine learning methods have been explored in the ATE task. Compared to the rule-based methods, they acquire better performance [24–26]. In the traditional machine learning algorithms, both supervised and unsupervised learning, such as HMM and latent Dirichlet allocation (LDA) [27], are explored in the ATE task. LDA is a document-level probabilistic model and it is powerless in capturing co-occurrence relationships in a sentence. To tackle the drawback, the Enriched LDA (ELDA) is designed by Shams and Baraani-Dastjerdi [28]. The ELDA method can integrate domain knowledge and LDA. The domain knowledge can be obtained by calculating the co-occurrence frequency and similarity between aspects and relevant topics. Wang et al. [29] proposed an association constrained LDA (AC-LDA) for capturing the co-occurrence relationships in LDA. The features of the syntactic structure are used to formalize word association relationships. Ozyurt et al. [30] proposed a sentence segment LDA (SS-LDA) to alleviate the data sparsity problem in short texts for the ATE task. However, in practice, the supervised learning methods perform better than the unsupervised learning methods. The sequence labeling learning is adopted in the ATE task in the supervised learning methods. In order to combine the linguistic features with HMM, Jin et al. [5] proposed the lexicalized HMM. Shu et al. [31] proposed lifelong CRF (L-CRF) in supervised ATE. L-CRF first learns prior knowledge from past domains, and then it utilizes the knowledge by lifelong learning to identify aspect terms in a new domain. However, these traditional statistical machine learning-based supervised learning approaches highly depend on the designed features, which can consume a lot of energy and time.

Recently, the deep neural network has made significant progress and dominated in many areas, including ATE. There is no need for carefully designing features by hand, these methods can automatically produce the more suitable and abstract features for the end task. The field of computer coding has also made some progress [32]. In the ATE task, the convolutional neural network (CNN) [33], LSTM [34, 35], GRU [10], and recursive neural network [36] are often employed in extracting aspect terms. Aspect extraction in a deep learning manner is first studied by Poria et.al [37]. They proposed a 7-layer CNN architecture to tag each word in a sentence. To further improve the performance, they combined the CNN classifier with a series of linguistic rules. Luo et al. [38] proposed an incremental deep learning method that combines bidirectional LSTM (BiLSTM) and CRF. The tree-structured knowledge is encoded from the given dependency tree by either of two propagations. Xu et al. [7] proposed a dual embeddings CNN (DE-CNN) which explored the different embeddings in the ATE task. In addition to the general domain embedding, the specific domain embedding is also considered in DE-CNN. Zhang et al. [39] proposed a topic-aware dynamic CNN (TADC) to extract aspect terms. In the TADC model, a neural topic model is used to improve the ability to identify aspect terms.

Phan et al. [40] combined syntactical information with the contextualized word embeddings such as BERT [13] to further enhance the ability of the model toward identifying aspect terms (or targets) that are expressed in a sentence. Venugopalan et al. [41] proposed a guided LDA model with BERT for each aspect category. Oh et al. [42] proposed a deep contextualized relation-aware network (DCRAN) for aspect extraction. They designed two modules to capture the association relationship between subtasks of ABAE with the contextualized information. Lekhtman et al. [43] studied the domain adaptation of aspect extraction and proposed a customized pre-training method for BERT with the unlabeled data. Nguyen et al. [44] proposed a novel weakly supervised method for the ATE task. An uncertainty-aware objective function is developed to utilize seed words in the weakly-supervised method. Zhang et al. [45] treated the ATE task as a question-answering task. Hu et al. [46] viewed the aspect extraction task as a multi-label learning problem. They employed the prototypical network to develop a few-shot learning model and designed two attention mechanisms to alleviate the noise.

Different attention mechanisms are often employed in the ATE task [47–49]. The attention mechanism can pay more attention to the important words that are relevant to the specific task and ignore the inessential words that are not relevant to the specific task in a sentence. Some works also simultaneously perform the ATE and ALSC tasks or the ATE and OTE tasks. For the aspect term-polarity co-extraction, a dual recurrent neural network with other units, named Dual crOss-sharEd RNN (DOER), is proposed by Luo et al. [8]. A span-based model is developed by Zhao et al. [50]. The model is trained with other tasks and aspect-opinion pairs are detected by the span boundaries. Chen et al. [51] proposed two transformer-based decoders for the ATE, OTE, and ALSC tasks. Mukherjee et al. [52] developed a pointer network-based model for the aspect sentiment triplet extraction. Besides, sufficient annotated data is essential in ATE. To obtain more data, the conditional generation method is employed to generate data in Li et al. [53], the process of generating was controllable and can generate more diversified sentences. They employed Transformer [11] to implement the conditional strategy. To alleviate the problem of insufficient annotated data, Wang et al. [54] proposed a self-training framework for the ATE task, and they utilized the discriminator to deal with the problem of the pseudo-labels.

These deep learning-based approaches treat ATE as the sequence labeling task and use BIO labels to tag each word. However, the overall semantic of a sentence cannot be well captured by the sequence labeling learning. The dependency relations between the output tags are also important in ATE, whereas the sequence learning is weak at capturing the dependencies between tags. To alleviate the weaknesses mentioned above, Ma et al. [55] developed a Seq2Seq learning-based model, named Seq2Seq4ATE. The encoder of the model consists of a bidirectional GRU. The decoder of the model is composed of another unidirectional GRU. Instead of using BIO tags, Hu et al. [56] developed the span boundaries-based framework to detect aspect terms in the ATE task. They employed BERT as the default backbone network. Our proposed model is different from the Seq2Seq4ATE and the span boundaries-based method. The encoder of the proposed IANN can integrate information of surrounding words toward a word in a sentence by MCRN which can capture the more informative information for the ATE task. The dynamic meaning of words can be modeled by the contextualized embedding layer in our proposed model. Besides, the novel AO tags are used as the tagging scheme in the proposed IANN.

Methodology

The ATE task is defined and the proposed IANN model is discussed in detail in this part. Table 1 exhibits the notations that are used in the proposed model.

Table 1 The notations used in the proposed IANN

Notation	Category	The relevant explanation
S	Set	A text sequence of the specific sentence
w_i	Vector	The one-hot vector form of the ith word
n	Scalar	The length of a specific sentence
S^{input}	Set	A specific input token sequence
$\lvert S^{input} \rvert$, p	Scalar	The length of the specific input token sequence except for [CLS] and [SEQ]
BERT()	Function	The BERT function
d	Scalar	The size of the hidden dimension of the BERT model
h^{bert}	Matrix	The output of the BERT model in the embedding layer
h_i^{bert}	Vector	The ith term in h^{bert} excluding [CLS] and [SEQ]
$\widehat{s}_h^i$, $\widehat{s}_w^i$	Scalar	The size of the convolution kernel in the ith convolution operation
h^{c_i}	Matrix	$i \in [1, 2, 3, 4, 5]$, the output representations of the ith convolution operation
d_i	Scalar	$i \in [1, 2, 3, 4, 5]$, the number of output channels produced by the ith convolution
W_i, U_i	Matrix	$i \in \{m, z, r, h, g, u\}$, the weight parameter matrix in IANN
b_i	Vector	$i \in \{m, z, r, h, g, u\}$, the bias in IANN
d_m	Scalar	The dimension of the hidden representation of MLP in MCRN
d_g	Scalar	The size of the output in GRU
d_n	Scalar	The output size of MLP in the multi-layer encoder
r_t, z_t	Vector	The value toward reset gate and update gate
N	Scalar	The number of hops in the multi-layer encoder
$\overrightarrow{h}_t$, $\overleftarrow{h}_t$, h_t	Vector	The forward, backward, and bidirectional outputs of the tth time step in GRU
r	Scalar	The size of each label embedding
MCRN()	Function	The MCRN function
$h_t^{l_i}$	Vector	The output of the tth token in MCRN of a multi-layer encoder in hop i
$h_t^{g_i}$	Vector	The output of the tth token in MLP of a multi-layer encoder in hop i
h_t^e	Vector	The output of the tth token in a multi-layer encoder
h_t^u	Vector	The output of the tth token in the unidirectional GRU unit in decoding layer

Table 1 (continued)

Notation	Category	The relevant explanation
d_u	Vector	The output size of the unidirectional GRU unit in the decoding layer
$\widehat{e}$	Vector	The specific label embedding
$\widetilde{y}_t$	Vector	The one-hot vector of a specific predicted label at time step *t*
E	Matrix	The label embedding matrix
lookup()	Function	The lookup table operation
h_t^v	Vector	The hidden representation of MLP in the decoding layer at time step *t*
d_v	Vector	The output size of MLP of the decoding layer
h_t^s	Vector	The *t*th time step output in the softmax
softmax()	Function	The softmax function
c	Scalar	The number of the predicted labels
$\widehat{y}_t$	Scalar	The predicted label toward the word at time step *t*

Task definition

Given a sentence $S = \{w_1, w_2, ..., w_{n-1}, w_n\}$, the length of the word sequence is n, the ATE task aims at detecting explicit aspect terms $A = \{a_1, a_2..., a_m\}$ toward an entity, the total number of the explicit aspect terms is *m* in the specific sentence *S*. Each aspect term may consist of one or more words in the specific sentence *S*.

An overview of the IANN model

For the purpose of modeling the more informative information and integrating surrounding words' information toward the specific word in the encoder, as well as to capture the dynamic word sense in the specific sentence, this paper designs a novel Seq2Seq learning framework, named IANN. In IANN, the contextualized embedding layer is used to capture the dynamic meaning of words. In the contextualized embedding layer of IANN, BERT is employed to generate contextualized word embeddings. At the encoder layer of IANN, a novel hierarchical neural network, named MCRN, is designed to not only encode the more informative information and integrate information of surrounding words toward a specific word but also to model the sequence information and capture the long-distance dependencies by combining multiple convolution and recurrence operations. MCRN is the key component of the encoder in IANN. The encoder in IANN is either a single-layer architecture or a multi-layer architecture. The decoder layer of IANN is designed to decode the encoding information and convert the final word representations into appropriate features that are suitable for predicting labels. Besides, AO dual tags are proposed as the less challenging tagging labels. Figure 2 illustrates the entire framework of the designed IANN with novel AO tags.

Contextualized embedding layer

The word embedding can map words from the discrete, sparse one-hot vector into the continuous, dense representation. One advantage of word embedding is that the word sense and syntax can be captured in the embedding space to some extent. The static word embedding such as Glove is the fixed word embedding. However, in different contexts, some aspect terms are domain-dependent and one term may or may not be an aspect term in the ATE task. For instance, in a laptop review "the memory is enough for use", memory is an aspect term, while in the sentence "memory is sad for me", memory is not an aspect term. The embedding of memory should be different in different contexts. One solution is to use contextualized word embedding such as BERT, which can capture the specific meaning of a word and obtain different embedding of a word in different contexts. Hence, this paper designs the contextualized embedding layer for modeling dynamic word meaning.

In the contextualized embedding layer, the BERT model is employed to generate contextualized word embeddings that are able to capture dynamic word meaning to some extent. BERT is a new language representation model. Unlike ELMo [57] which employs LSTMs as its backbones, BERT adopts transformers as its backbones. Compared to GPT [58] which is an autoregressive language model and employs unidirectional transformers to model the unidirectional information toward a word in a sentence, BERT is an autoencoder language model, it masks a few words in a sentence. One of the training objectives of BERT is that the randomly masked word can be accurately recovered by the bidirectional information. BERT can simultaneously model the bidirectional information toward a word in a sentence.

BERT has two-parameter models: $\text{BERT}_{\text{BASE}}$ and $\text{BERT}_{\text{Large}}$. In the $\text{BERT}_{\text{BASE}}$ model, it has 12 layers, the head number is 12 in the multi-head attention. The size of output representations of the transformer encoder is 768. There are about 110 M parameters in $\text{BERT}_{\text{BASE}}$. In the $\text{BERT}_{\text{Large}}$ model, it has 24 layers, the head number is 16 in the multi-head attention. The size of output representations of the transformer encoder is 1024. There are about 340 M parameters in $\text{BERT}_{\text{Large}}$. As our computing ability is limited, hence the $\text{BERT}_{\text{BASE}}$ model is used as the backbone network in the contextualized embedding Layer.

In the ATE task, the inputs that are fed into the BERT are explicitly transformed into a series of tokens in a sequence. Given a sentence $S = \{w_1, w_2, ..., w_{n-1}, w_n\}$, a specific symbol [CLS] is inserted first at the front of a specific sentence *S*. Also, another specific symbol [SEP] is inserted

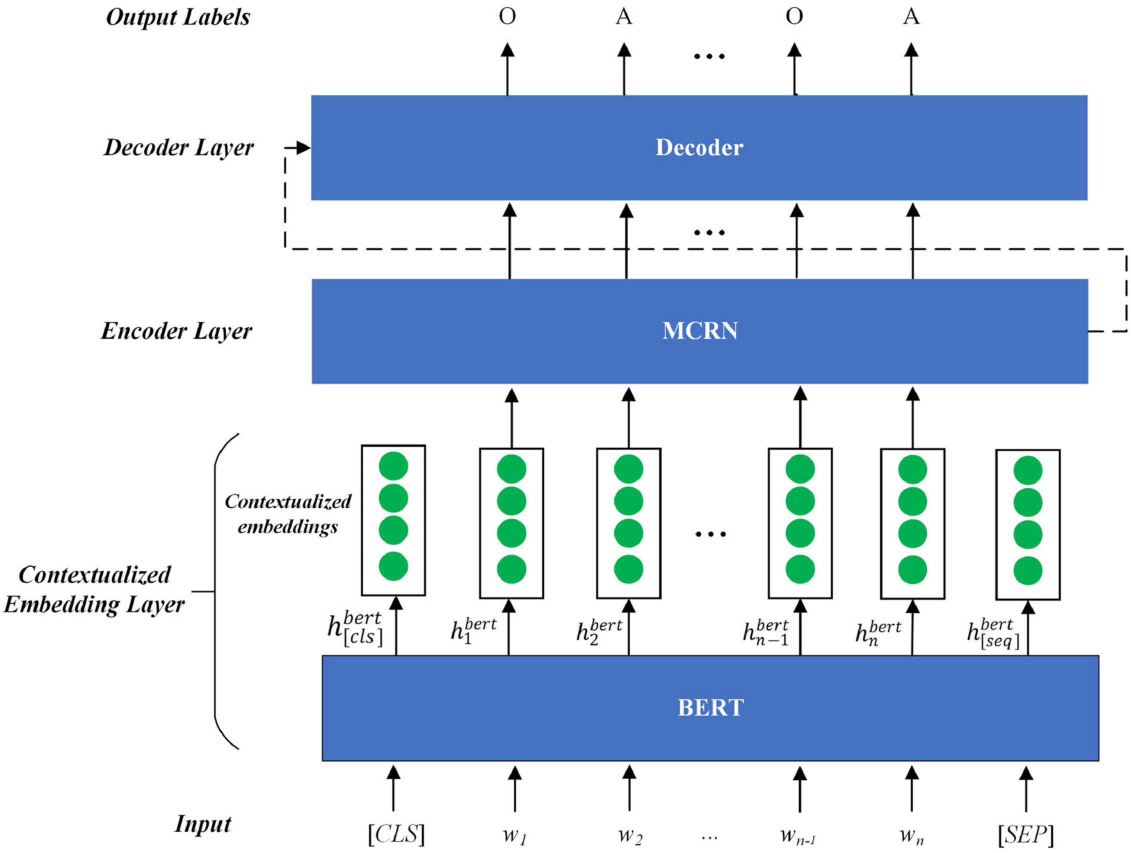


Fig. 2 The overall framework of IANN. The encoder layer is composed of the single-layer encoder in the current diagram. The output labels are the dual AO tags

at the end of the specific sentence S. A token sequence $S^{input} = \{[CLS], w_1, w_2, ..., w_{n-1}, w_n, [SEP]\}$ can be obtained, it is then fed into the developed IANN, and the total length of the word sequence S is n. Figure 2 shows the specific input form. The output representation in contextualized embedding layer is defined by formula (1):

$$h^{bert} = BERT(S^{input}) \tag{1}$$

where $h^{bert} \in R^{|S^{input}| \times d}$,$|S^{input}| = n + 2$, n is the total length of the raw sentence S, the output dimension of BERT is d, $|S^{input}|$ is the overall length of the transformed input of BERT, BERT() denotes all operations in BERT.

In the outputs from the BERT model, except for output representations of special tokens [CLS] and [SEQ] which are separately denoted as $h^{bert}_{[cls]}$ and $h^{bert}_{[seq]}$, the other output representations in BERT are sent to the next layer, which is denoted as $h^{bert}_i \in R^d$, $i \in [1, n]$.

MCRN

In the proposed model, the encoder is either a single-layer architecture or a multi-layer architecture. In the single-layer architecture, the encoder consists of an MCRN model. In other words, an MCRN model is employed as the encoder in the single-layer encoder. In the multi-layer architecture, the encoder mainly consists of multiple MCRN models.

The MCRN model is a hierarchical architecture. As the key component of the encoder, the multiple convolution layer, fully connected layer, and bidirectional GRU make up MCRN. The MCRN model can integrate information of surrounding words toward a specific word by using multiple convolution operations as well as capture the bidirectional long-distance dependencies and model the sequence information by employing the bidirectional GRU. Therefore, MCRN can model more informative information than the single-type neural network. In the single-layer encoder, MCRN is the encoder. The overall architecture of MCRN can be seen in Fig. 3.

Multiple convolution layer

The operations of the convolution are the key techniques for integrating information of surrounding words toward a specific word and are sensitive to local patterns in a specific sentence. Convolution can model the current word and

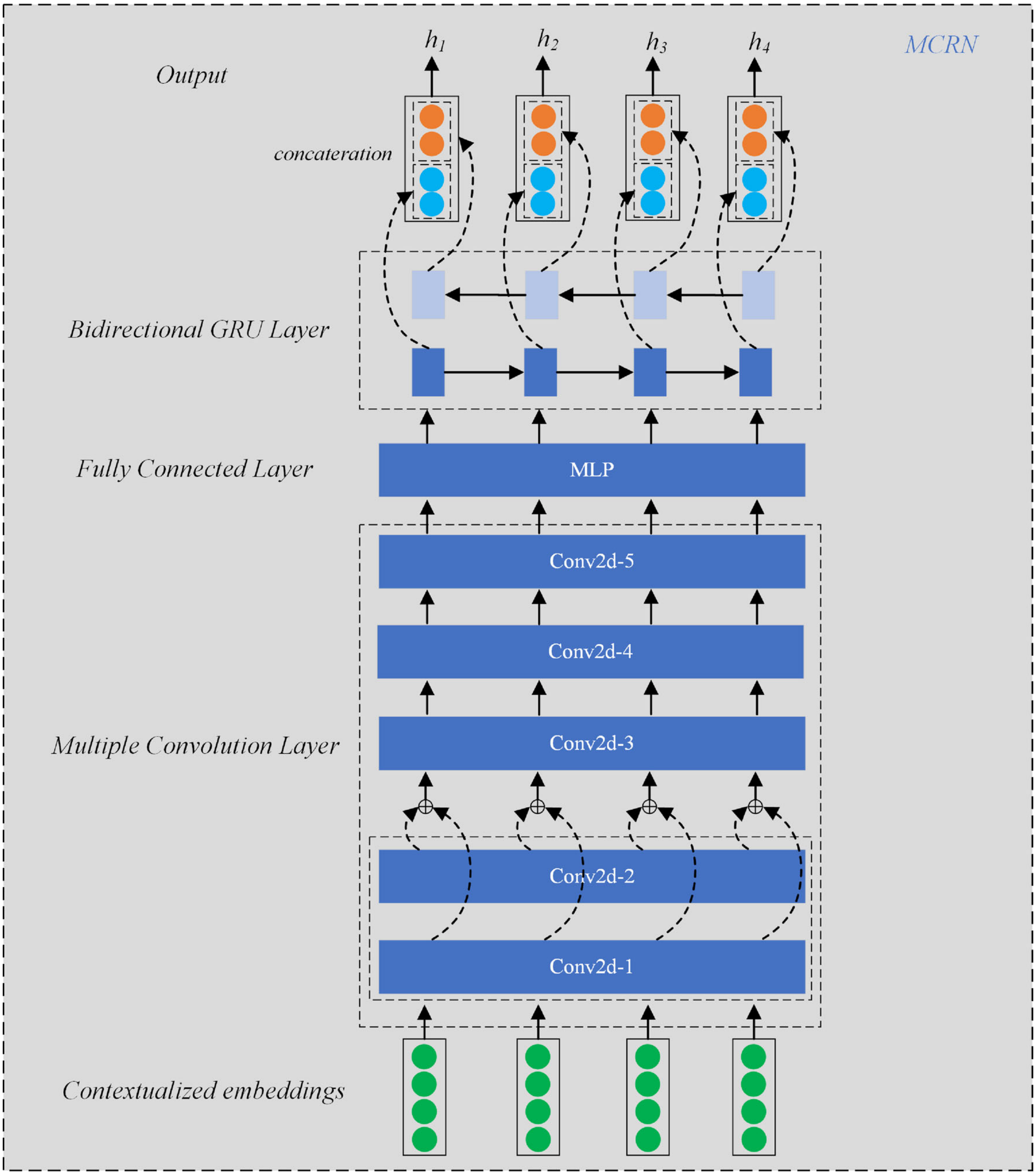


Fig. 3 The overall architecture of MCRN

its surrounding words within a specified window. The size of the specified window decides the scope of the neighbor words. Only within the specified scope, the current word and its surrounding words can be modeled and integrated. Hence convolution is sensitive to local patterns and features in a specific sentence. The larger the window size, the more neighbor words are modeled. The specific architecture of the multiple convolution layer can be found in Fig. 3. In the multiple convolution layer, there are five convolution operations. In Fig. 3, these five convolution operations are separately denoted as Conv2d-1, 2, 3, 4, and 5. The symbol "⊕" denotes the concatenation of the vectors in Fig. 3. The output representations of the convolution operations are defined by formula (2) to formula (6):

$$\mathrm{h}^{\mathrm{c}_1} = \mathrm{g}\left(\mathrm{Conv2d}\left(\mathrm{h}^{\mathrm{bert}} \middle| \widehat{\mathrm{s}}_{\mathrm{h}}^{1}, \widehat{\mathrm{s}}_{\mathrm{w}}^{1}\right)\right) \tag{2}$$

$$\mathrm{h}^{\mathrm{c}_2} = \mathrm{g}\left(\mathrm{Conv2d}\left(\mathrm{h}^{\mathrm{bert}} \middle| \widehat{\mathrm{s}}_{\mathrm{h}}^{2}, \widehat{\mathrm{s}}_{\mathrm{w}}^{2}\right)\right) \tag{3}$$

$$\mathrm{h}^{\mathrm{c}_3} = \mathrm{g}\left(\mathrm{Conv2d}\left([\mathrm{h}^{\mathrm{c}_1}; \mathrm{h}^{\mathrm{c}_2}] \middle| \widehat{\mathrm{s}}_{\mathrm{h}}^{3}, \widehat{\mathrm{s}}_{\mathrm{w}}^{3}\right)\right) \tag{4}$$

$$h^{c_4} = g\left(\mathrm{Conv2d}\left(h^{c_3} \middle| \widehat{s}_h^4, \widehat{s}_w^4\right)\right) \tag{5}$$

$$h^{c_5} = g\left(\mathrm{Conv2d}\left(h^{c_4} \middle| \widehat{s}_h^5, \widehat{s}_w^5\right)\right) \tag{6}$$

where Conv2d represents the specific convolution operation, the notation "|" implies that the convolution operation dependents on the parameter $\widehat{s}_h^i$ and $\widehat{s}_w^i$, which are hyperparameters. The symbols $\widehat{s}_h^i$ and $\widehat{s}_w^i$ separately represent the height and the width of the filter in the convolution operation, and the notation i represents the ith convolution operation. The concatenation operation is denoted by the symbol ";", it means concatenating in the column of the matrix in formula 4. for the sake of simplicity, $|S^{\mathrm{input}}|$ is represented by p, $h^{\mathrm{bert}} \in R^{p\times d}$ is the output representation of the BERT model, $h^{c_1} \in R^{p\times d_1}$, $h^{c_2} \in R^{p\times d_2}$, $h^{c_3} \in R^{p\times d_3}$, $h^{c_4} \in R^{p\times d_4}$, and $h^{c_5} \in R^{p\times d_5}$ are the output representations of the five convolution operations, respectively. The dimension of the output channels of each convolution operation is d_1, d_2, d_3, d_4, and d_5, respectively. The parameters d_1, d_2, d_3, d_4, and d_5 are all hyper-parameters and customized in the multiple convolution layer.

In each convolution operation of the proposed MCRN model, for keeping the number of input sequences the same, the padding is added to both sides in the first dimension of the input representations in all convolution operations. The stride of each convolution operation is set to 1, and the value of padding of each convolution operation is $(\widehat{s}_h^i - 1)/2$.

Fully connected layer

For the purpose of transforming output representations of the multiple convolution layer into suitable features for the next layer or module, the fully connected layer is designed in the MCRN model. The multi-layer perception (MLP) makes up the fully connected layer. The computational formula is given as formula (7):

$$h^m = \mathrm{relu}(h^{c_5} \cdot W_m + b_m) \tag{7}$$

where $W_m \in R^{d_5\times d_m}$ is the learned parameter, the bias term is $b_m \in R^{d_m}$ in the fully connected layer, both parameters are learned in the training process, the operation "·" denotes the dot product operation, $h^m \in R^{p\times d_m}$, d_m represents the output dimension in the MLP.

Bidirectional GRU layer

In the ATE task, the sequence information and long-distance dependencies are important. The vanilla RNN is weak in capturing long-distance dependencies because the problem of vanishing gradient damages the model performance, especially when dealing with long sentences. To overcome the problem in the vanilla RNN, the GRU model has been developed, it is a variant of the vanilla RNN. GRU is good at modeling the sequence information and capturing the long-term dependencies. Instead of unidirectional GRU, bidirectional GRU is employed in MCRN. The advantage of the bidirectional GRU is that not only the forward but also backward information of the sentence can be modeled. Forward information is historical information, future information is backward information toward a token in the sentence. The architecture of the bidirectional GRU can be found at the top of Fig. 3.

The forward and backward GRU forms the bidirectional GRU. As shown in Fig. 3, the sequence input order of the forward GRU is the same as the word order, the sequence input order of the backward GRU is from right to left. In the unidirectional GRU, it is composed of two gates and one hidden. Let's take the forward GRU as an example, the outputs can be obtained by formula (8) to formula (11):

$$z_t = \sigma_g(W_z h_t^m + U_z h_{t-1} + b_z) \tag{8}$$

$$r_t = \sigma_g(W_r h_t^m + U_r h_{t-1} + b_r) \tag{9}$$

$$\widetilde{h}_t = g(W_h h_t^m + U_h(r_t \odot h_{t-1}) + b_h) \tag{10}$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \widetilde{h} \tag{11}$$

where σ_g represents the logistic function, g represents the hyperbolic tangent function in the current GRU of MCRN, the output of tth token in GRU is $h_t \in R^{d_g\times 1}$, d_g is the output size of the GRU unit, $h_t^m \in R^{d_m\times 1}$ is the hidden representation of the fully-connected layer toward the tth token, in other words, h_t^m is the input of GRU toward the tth token in a sentence. The update and reset gates are z_t and r_t in GRU, respectively. The element-wise multiplication is denoted by the symbol $\odot$. For simplicity, the symbol $\overrightarrow{h}_t$ is used to denote the final output of the forward GRU and the symbol $\overleftarrow{h}_t$ is used to denote the output representation of the backward GRU toward the tth token in a sentence. The final output representations of bidirectional GRU can be obtained as formula (12):

$$h_t = [\overrightarrow{h}_t; \overleftarrow{h}_t] \tag{12}$$

where $\overrightarrow{h}_t \in R^{d_g\times 1}$, $\overleftarrow{h}_t \in R^{d_g\times 1}$, $h_t \in R^{2d_g\times 1}$, the concatenating vector operation is represented by the symbol ";" in the formula 12.

Multi-layer encoder

The single-layer encoder can be extended to a multi-layer encoder. The overall architecture of the multi-layer encoder

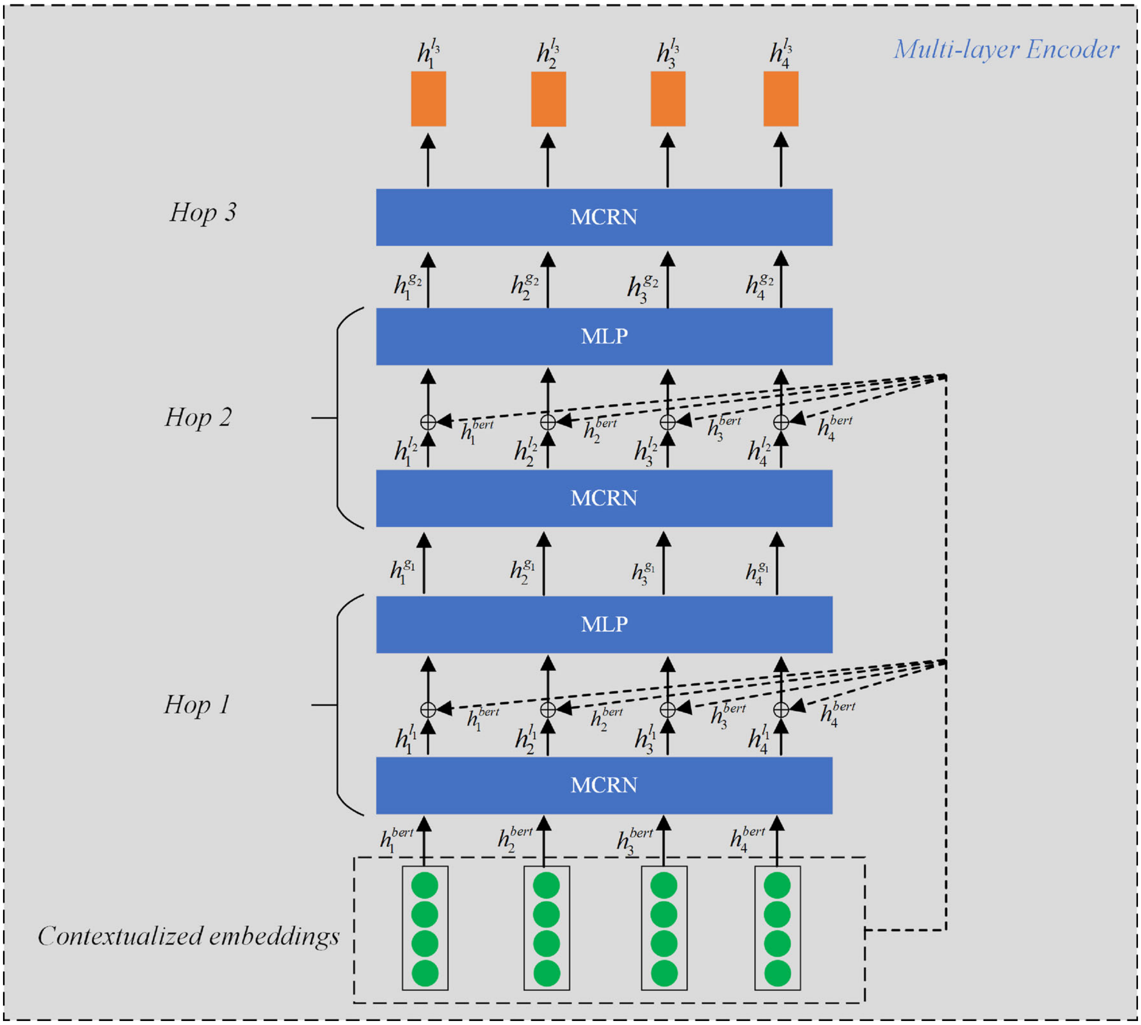

Multi-layer Encoder
$h_1^{l_3}$ $h_2^{l_3}$ $h_3^{l_3}$ $h_4^{l_3}$
Hop 3
MCRN
$h_1^{g_2}$ $h_2^{g_2}$ $h_3^{g_2}$ $h_4^{g_2}$
MLP
Hop 2
h_1^{bert} h_2^{bert} h_3^{bert} h_4^{bert}
$h_1^{l_2}$ $h_2^{l_2}$ $h_3^{l_2}$ $h_4^{l_2}$
MCRN
$h_1^{g_1}$ $h_2^{g_1}$ $h_3^{g_1}$ $h_4^{g_1}$
MLP
Hop 1
h_1^{bert} h_2^{bert} h_3^{bert} h_4^{bert}
$h_1^{l_1}$ $h_2^{l_1}$ $h_3^{l_1}$ $h_4^{l_1}$
MCRN
h_1^{bert} h_2^{bert} h_3^{bert} h_4^{bert}
Contextualized embeddings

Fig. 4 The overall framework of the multi-layer encoder with three computational layers (hops) in IANN

with three computational layers (hops) is exhibited in Fig. 4. The multi-layer architecture can capture more useful information. For the specific end task, the multi-layer encoder can learn the more appropriate and informative representations with the increment of the layers.

Each computational layer (hop) except the last computational layer (hop) is composed of an MCRN model with an MLP. The last computational hop only consists of an MCRN. Compared to previous hops, the last hop does not involve an MLP, because the last hop does not need to transform the output features to the input representations that are appropriate for the modeling in the next hop. The final outputs of the multi-layer encoder are the output features generated by the last hop, the final output representations are directly fed into the decoder.

The parameter sharing strategy is used to accelerate the training process and avoid having too many parameters in the model. The parameters of MCRNs and MLPs in each hop are shared. Take the multi-layer encoder with the three computational hops as an example, the outputs of the multi-layer encoder can be obtained by formula (13) to formula (17):

$$h_t^{l_1} = \mathrm{MCRN}\left(h_t^{bert}\right) \tag{13}$$

$$h_t^{g_1} = \mathrm{Relu}\left(W_g\left[h_t^{l_1}; h_t^{bert}\right] + b_g\right) \tag{14}$$

$$h_t^{l_2} = \mathrm{MCRN}\left(h_t^{g_1}\right) \tag{15}$$

$$h_t^{g_2} = \mathrm{Relu}\left(W_g\left[h_t^{l_2}; h_t^{bert}\right] + b_g\right) \tag{16}$$

$$h_t^{e} = h_t^{l_3} = \mathrm{MCRN}\left(h_t^{g_2}\right) \tag{17}$$

where $h_t^e \in R^{2d_g \times 1}$ denotes the final output representations of the multi-layer encoder, $h_t^{l_1} \in R^{2d_g \times 1}$, $h_t^{l_2} \in R^{2d_g \times 1}$ and $h_t^{l_3} \in R^{2d_g \times 1}$ are respectively the outputs of MCRN in hop1,

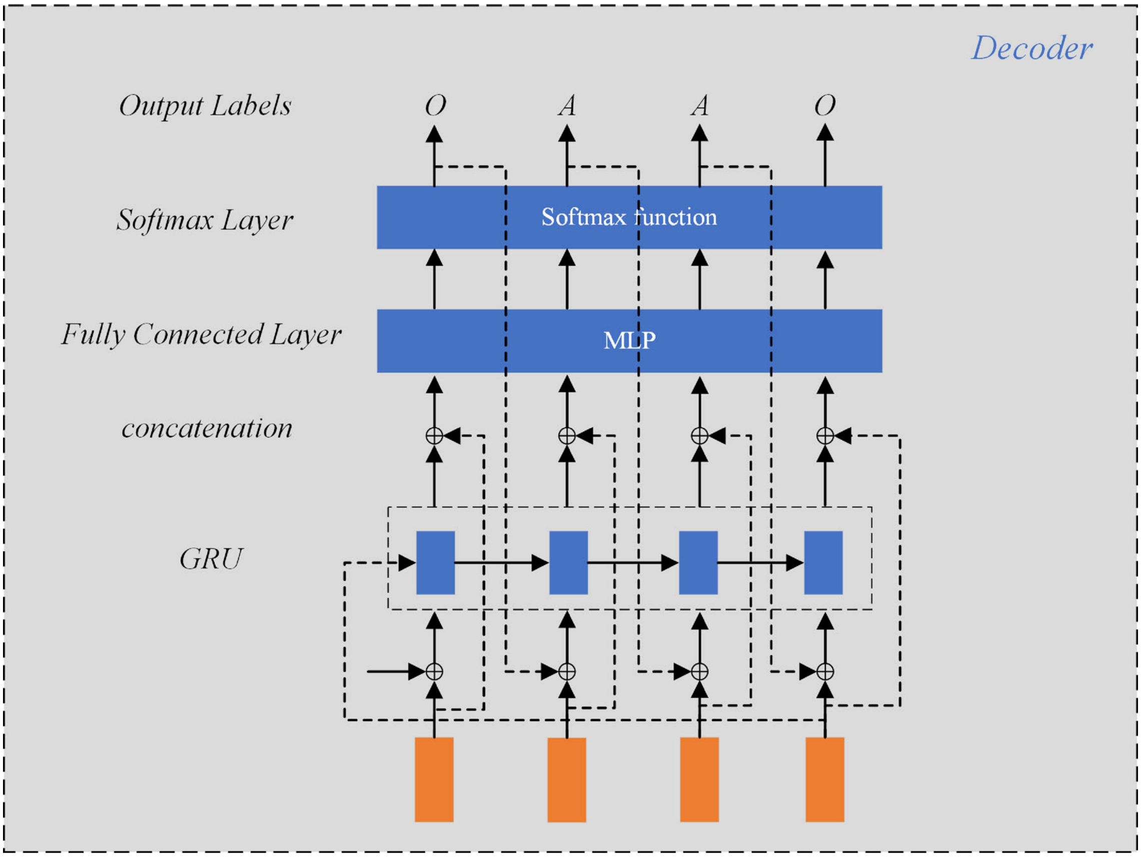


Fig. 5 The overall framework of the decoder

hop 2, and hop 3 toward tth token, the dimension of the hidden representation in the GRU unit of MCRN is $\mathrm{d_g}$, $\mathrm{h_t^{g1}} \in \mathrm{R^{d_n \times 1}}$ and $\mathrm{h_t^{g2}} \in \mathrm{R^{d_n \times 1}}$ are respectively the outputs of MLP in hop 1 and hop 2 toward tth token, the dimension of hidden representation in MLP of the multi-layer encoder is $\mathrm{d_n}$, MCRN() denotes the operations of MCRN, $\mathrm{W_g} \in \mathrm{R^{d_n \times (2d_g + d)}}$ and $\mathrm{b_g} \in \mathrm{R^{d_n}}$ are separately the weight and bias in the MLP, they are learned in the training process.

Decoder layer

The decoding layer receives the output representation from the encoder layer as the input. The initial state of the unidirectional GRU in the decoding layer is initialized by the last forward hidden output of the bidirectional GRU of the last hop. The decoder layer is used to decode the encoding representations to generate the labels of words. The fully connected layer, softmax layer, and GRU make up the decoding layer. Besides, this paper proposes the novel AO tags which are consisted of two labels: aspect and outside. The overall architecture of the decoder can be seen in Fig. 5. The following subsections discuss the details of each layer of the decoder.

GRU layer

In the decoding layer, it is not possible to utilize future information (backward information) to predict the current label in a real scenario. Hence, the unidirectional GRU is employed in the decoder. The details of the GRU have been discussed in the previous section, these details are not described in this section. It is worth pointing out that the initial state of the unidirectional GRU in the decoder is assigned by the last forward hidden representation in the bidirectional GRU. The overall meaning of a whole sentence is important for predicting the labels in the ATE task. For example, “The memory is nearly full” in a laptop review, the explicit aspect term is the word “memory”, however, in another sentence “The memory is a little fuzzy”, the word “memory” is not an aspect term. The last forward output representation of the bidirectional GRU of MCRN can capture and model the overall semantics of the sentence. In other words, the representation of the specified sentence is obtained by the last forward output representation of the bidirectional GRU of MCRN. Hence, the model can improve performance by considering the overall meaning of a whole sentence in the ATE task.

For the specific token, the label information of its previous token is also important to accurately obtain its label. The

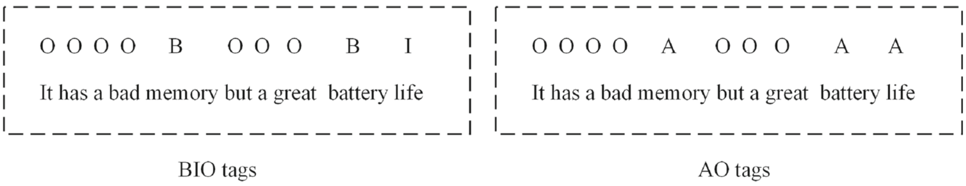

O O O O B O O O B I
It has a bad memory but a great battery life
BIO tags
O O O O A O O O A A
It has a bad memory but a great battery life
AO tags

Fig. 6 The difference between BIO tags and AO tags

decoder can directly make use of the previous word's label information to improve the model performance by feedback on previous label information. The outputs of the GRU layer in the decoder can be obtained by formula (18) to formula (19):

$$\mathrm{h}_t^u = \mathrm{GRU}([\mathrm{h}_t^e; \widehat{\mathrm{e}}]) \tag{18}$$

$$\widehat{\mathrm{e}} = \mathrm{lookup}(\widetilde{\mathrm{y}}_t) = \mathrm{E} \cdot \widetilde{\mathrm{y}}_t \tag{19}$$

where $\mathrm{h}_0^u = \overrightarrow{\mathrm{h}}_p^e$, $\mathrm{h}_t^u \in \mathrm{R}^{d_u \times 1}$ denotes the outputs of the unidirectional GRU unit belonging to the decoding layer, d_u is the size of the outputs, GRU() denotes all operations of the GRU unit in formulas 8–11, the parameters of the GRU unit in the decoding layer are different from the parameters of the bidirectional GRU unit in the encoding layer, $\widehat{\mathrm{e}} \in \mathrm{R}^{r \times 1}$ is the label embedding, the size of the label embedding is r, $\mathrm{E} \in \mathrm{R}^{r \times c}$ is the label embedding matrix, $\widetilde{\mathrm{y}}_t \in \mathrm{R}^{c \times 1}$ is the predicted label's one-hot vector toward the tth token in a sentence, the number of output labels is c for each token, lookup() denotes the lookup table operation.

Fully connected layer

In order to transform the hidden states of GRU into the appropriate features for the label prediction, the fully connected layer is employed. In the decoder, the fully connected layer receives outputs of the unidirectional GUR as the inputs. An MLP constitutes the fully connected layer in the decoder. The MLP in the decoder is different from the MLP in the encoder, the former does not involve a nonlinear operation. The outputs of the fully connected layer are denoted as formula (20):

$$\mathrm{h}_t^v = \mathrm{W}_v \mathrm{h}_t^u + \mathrm{b}_v \tag{20}$$

where $\mathrm{h}_t^v \in \mathrm{R}^{d_v \times 1}$, $\mathrm{W}_v \in \mathrm{R}^{d_v \times d_u}$, $\mathrm{b}_v \in \mathrm{R}^{d_v}$ are the weight and bias in the MLP of the decoding layer, they are also learned in the training process, and the output size of MLP is d_v in the decoder layer.

Softmax layer

For the purpose of getting the probabilities of each label toward a specific word, the softmax layer is developed in the decoder which consists of the softmax function. The label toward a certain word is obtained by finding the label with the maximum value after the softmax function. The computational process can be seen from formula (21) to formula (22):

$$\mathrm{h}_t^s = \mathrm{softmax}\,(\mathrm{h}_t^v) \tag{21}$$

$$\widehat{\mathrm{y}}_t = \underset{c}{argmax}\,(\mathrm{h}_t^s) \tag{22}$$

where $\mathrm{h}_t^s \in \mathrm{R}^{c \times 1}$ is the probability distribution value of the tth token in the softmax function, c is the total number of all label classes.

AO labels

This paper develops the novel dual tags, named AO tags, as the less challenging tagging scheme. The AO tags contain two labels: A (Aspect) and O (Outside). The Aspect tag denotes that the current word is the aspect term, and the Outside tag denotes that the current term is a non-aspect term. For instance, It has a bad memory but a great battery life, memory and battery life are two aspect terms, the output label sequence of the specific sentence using AO tags can be seen in Fig. 6. The AO tags are the less challenging tagging scheme. Because modeling the probability distribution of two types of tags is easier than modeling the probability distribution of three types of tags.

Algorithm 1 The process of extracting different aspect terms under AO tags

```
Input: the batch output sequence array of AO tags (T), the length of the specific input
    sequence (n), the batch raw input text array (R), the output “0” in T denotes Outside tag,
    the output “1” in T denotes Aspect tag
Output: the aspect terms list (AT)
A = [] // initialization of the aspect term list
B = T // initially array for BIO tags
repeat s in Output (B) // selecting each sentence
    a = n - 1
    repeat a
        i = 0
        if s[i] == 1 or s[i] == 2
            j = i + 1
            if s[j] == 1
                s[j] = 2
        i = i +1
    end repeat
end repeat
a = 0
repeat sentence (s) in Output (B)
    start_index = 0, end_index = 0
    predict_terms = [] // initially empty array of predicting for the current sentence
    i = 0
    repeat n
        if s[i] == 1
            start_index = i
            flag = False
            j = 0
            repeat sentence (k) in Output (B[i+1: ])
                if k[j] != 2
                    if flag is False
                        end_index = start_index
                    else
                        end_index = start_index + 1
                    break
                if k[j] == 2
                    end_index = i + 1 + j
                    flag = True
                j = j + 1
            end repeat
            aspect = R[a][start_index : end_index]
            add aspect in predict_terms
        i = i + 1
    end repeat
    a = a + 1
    add predict_terms to AT
end repeat
Output: the aspect term list (AT)
```

The most important thing about AO tags is how to get the different aspects under the AO tags in the ATE task. For BIO tags, it is easy to identify the different aspect terms by B tag. In a general way, the B tag denotes the beginning of an aspect term. For AO tags, the different aspects are separated by O tag, the difference between AO tags and BIO tags is illustrated in Fig. 6. The rules of detecting these different aspect terms under the AO tags are designed in Algorithm 1. Algorithm 1 explains in detail how to transform AO tags into BIO tags and then detect the different aspect terms.

Experiments

All datasets are described and abundant experiments are performed to show the excellent performance of IANN. The corresponding results are reported and the reasons why the proposed model can outperform other SOTA baselines are discussed in detail.

Datasets

To maintain consistency with previous studies, the laptop, restaurant, and twitter datasets are used to perform abundant experiments. These datasets are from different sources. The laptop dataset is from SemEval 2014 [59] which contains product reviews in a laptop domain. The restaurant dataset consists of SemEval 2014, 2015, and 2016 [60, 61] from the restaurant domain. The twitter dataset is built by Michell et al. [62] and is composed of twitter posts. The training and testing sets already exist in the laptop and restaurant datasets, IANN is trained on the training set and tested on the testing set. Following Li et al. [63] and Hu et al. [56], in the experiments, ten-fold cross-validation is used on the twitter dataset, as there is no training and testing set in the twitter dataset. In all datasets, the gold labels of all aspect terms are available, all aspect terms are labeled with the corresponding tags. Table 2 shows the statistics of three datasets, such as the number of all aspect terms and sentences, the max length of all aspect terms. Besides, the maximum number of all aspect terms within a sentence for the laptop, restaurant, and twitter dataset is 9, 25 and 9, respectively. For each fold of the twitter dataset, the training and testing sets contain 2115 and 235 sentences, respectively.

Experiment and hyperparameter setting

The publicly available $\mathrm{BERT_{BASE}}$ is employed in the contextualized embedding layer of the proposed model. In the single-layer MCRN encoder, the size of the convolution kernel is 3 in Conv2d-1, and the size of the filter is 5 in Conv2d-2, 3, 4, and 5. The dimension of the outputs in Conv2d-1 and Conv2d-2 is 128, and the dimension of the outputs in Conv2d-3, 4, and 5 is 256. The dimension of hidden representations of the unidirectional GRU unit in the encoder and decoder is 300. The dimension of label embedding is 100. The Adam optimizer is adopted to optimize parameters in IANN, and the learning rate of Adam is set to 0.00002. In the training stage, the warmup strategy is adopted and the value of the warmup is 0.1. The training epoch is 100 and the batch size is 32 in our experiments. The dropout is used in the proposed model and it is set to 0.55.

In this work, the evaluating metrics are the precision, recall, and F1 in the proposed model. Their computational formulas can be seen from formula (23) to formula (25):

$$\mathrm{P} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{23}$$

$$\mathrm{R} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{24}$$

$$\mathrm{F1} = \frac{2 \times \mathrm{P} \times \mathrm{R}}{\mathrm{P} + \mathrm{R}} \tag{25}$$

where P means precision, R means recall, TP is the true positives, FP is the false positives and FN is the false negatives.

Baselines

Some baselines related to the ATE task are described in this section. For validating the efficacy of the important layers or modules in IANN, the corresponding variants toward IANN are designed. To show the superior performance of IANN, this paper describes the SOTA models which are used to be compared to the proposed model in the ATE task.

Variants of IANN

In the proposed model, the contextualized embedding layer and the MCRN model are fundamental in the proposed IANN. For verifying the validity of the layers or components of IANN, some variants of IANN are developed. IANN-v1 is designed to verify the contextualized embedding layer. IANN-v2, IANN-v3, and IANN-v4 are proposed to validate the effectiveness of the MCRN model. IANN-v5 is designed to verify the method of capturing the overall semantics of a sentence which are used in the decoder in IANN. IANN-v6 is designed to verify the AO tags.

1. IANN-v1: This variant of IANN discards the contextualized embedding layer and employs fixed word embeddings in IANN. In other words, the BERT model is not used in this model, the 300d-Glove [64] word embeddings are employed to map the words into the fixed word embeddings. The difference between IANN-v1 and IANN is that IANN-v1 lacks the ability to model dynamic word meaning, in other words, IANN-v1 employs static pre-trained word embedding.
2. IANN-v2: This variant of IANN discards the multiple convolution operations and only remains the recurrence operations in MCRN. Hence, there is only the bidirectional GRU in MCRN, not including the convolution operations. The inputs of the bidirectional GRU in MCRN are the outputs of the contextualized embedding layer. The major difference between IANN and IANN-v2 is that this variant removes the multiple convolutional operations in MCRN. In IANN-v2, it cannot model the

Table 2 The information for all datasets

Dataset	Total			Training			Testing		
	#Sent	#Aspect	#Max	#Sent	#Aspect	#Max	#Sent	#Aspect	#Max
Laptop	1869	2936	11	1458	2302	9	411	634	11
Restaurant	3900	6603	25	2481	4314	25	1419	2289	16
Twitter	2350	3243	22	–	–	–	–	–	–

#Max denotes the maximum length of all aspect terms, #Aspect and #Sent denote the number of all aspect terms and sentences separately

information of the surrounding words toward a specific word and detect local features.

3. IANN-v3: This variant of IANN removes the recurrence operations in MCRN, there are only the multiple convolutional operations in MCRN. The major difference between IANN and this variant is that the latter model discards the bidirectional GRU in MCRM. In other words, IANN-v3 lacks the ability to model the word order information and capture the long-term dependencies.
4. IANN-v4: This variant of IANN employs the multiple transformers in the encoder, not using the multiple convolution and recurrence operations of MCRN. The difference between IANN and this variant is the latter model employs multiple transformers rather than MCRN as task-specific layers on top of BERT.
5. IANN-v5: This variant of IANN employs the average pooling of the forward outputs of the bidirectional GRU in MCRN to capture the overall semantics of a sentence, instead of using the last forward outputs of the bidirectional GRU in MCRN. In the decoder, the difference between IANN and IANN-v5 is that IANN-v5 uses the average pooling of the forward outputs of MCRN and concatenates average pooling representations, label embedding, and outputs of the encoder as inputs of the decoder.
6. IANN-v6: This is sixth the variant of IANN. Instead of AO tags, BIO tags are used in IANN-v6. The difference between IANN-v6 and IANN is that IANN-v6 uses BIO tags in the decoding stage, however, IANN uses AO tags in the decoding stage.

State-of-the-art methods

Some state-of-the-art methods in ATE are used to compare with IANN, they are CRF, WDEmb, Bi-LSTM, BiLSTM-CNN-CRF, DE-CNN, HAST, Seq2seq4ATE, DOER, TAG, SPAN, and BERT-BiGRU-CRF. The CRF and WDEmb fall into the category of utilizing CRF. The Bi-LSTM, BiLSTM-CNN-CRF, DE-CNN, and BERT-BiGRU-CRF fall into the category of employing neural networks in the ATE task. The HAST is the joint method for extracting opinion and aspect terms simultaneously. The Seq2seq4ATE belongs to the seq2seq model for the ATE task. The DOER, TAG, and SPAN are the BERT-based methods for the ATE task and they are also the joint methods for identifying aspect terms and the corresponding sentiment polarity. As most of these baselines employ the BIO tags as the tagging labels, IANN with the BIO tags, named IANN-BIO, is also used to demonstrate the superior performance of the developed framework.

1. CRF: CRF makes use of the Glove [64] word embedding as well as some basic feature templates in the CRF model for the ATE task.
2. WDEmb: WDEmb is the first to learn embeddings of the words and dependency paths. The optimization objective can be formalized as $\mathrm{w}_1 + \mathrm{r} \approx \mathrm{w}_2$, where $\mathrm{w}_1, \mathrm{w}_2$ are words, r is the corresponding path. The model employs CRF as the decoder, and the learned word embeddings and dependency path embeddings are used to predict the label in the ATE task [65].
3. Bi-LSTM: Bi-LSTM applies different kinds of Bi-RNN in the ATE task, such as Elman/Jordan-type RNN. Besides, different kinds of embeddings are also employed in the method, such as google embedding or amazon embedding [66].
4. BiLSTM-CNN-CRF: BiLSTM-CNN-CRF is the outstanding method for identifying all named entities in a sentence. It employs CNN and Bi-LSTM to model character and word-level representations, separately. The CRF is used to generate the appropriate transitions between labels [67].
5. DE-CNN: DE-CNN applies both Glove embedding and domain embedding in the multilayer convolutional neural network for extracting aspect terms [7].
6. HAST: HAST deals with the ATE task using two useful additional information with LSTMs which are opinion summary and aspect detection history [68].
7. Seq2seq4ATE: Seq2seq4ATE is the first to view the ATE task as the seq2seq labeling task. Besides, the decoder can leverage the word representations by a gated unit network. For considering the information of adjacent words in the decoding stage, position-aware attention is designed in the encoder [55].

Table 3 The experiments of different variants on each dataset

Method	Laptop			Restaurant			Twitter		
	P	R	F1	P	R	F1	P	R	F1
IANN-v1	81.47	80.44	80.95	78.04	88.82	83.08	68.12	64.94	66.48
IANN-v2	84.84	87.38	86.09	85.39	90.65	87.94	74.19	77.43	75.76
IANN-v3	84.87	88.49*	86.64	85.43*	90.35	87.82	73.22	78.29	75.66
IANN-v4	82.32	87.38	84.77	84.70	90.00	87.27	73.89	77.81	75.78
IANN-v5	84.00	86.12	85.05	84.38	90.87	87.51	72.34	78.17	75.12
IANN-V6	**86.40**	88.15	87.27*	85.25	91.32*	88.18*	74.50*	78.34*	76.37*
IANN	86.22*	**88.80**	**87.49**	**85.48**	**91.52**	**88.40**	**74.54**	**78.68**	**76.54**

The precision is denoted by "P", the recall is denoted by "R", and the F1 value is denoted by "F1". The best result is expressed in bold, and the second-best result is indicated by an asterisk

8. DOER: DOER is a dual architecture method. The model can learn the corresponding different features for different tasks. In order to model the correlation of two tasks, a cross-shared unit is proposed in DOER [8].
9. TAG: TAG performs the ATE task by the paradigm of the sequence labeling method, it employs BERT as an encoder and uses a CRF as the decoder [56].
10. SPAN: SPAN is the SOTA model in the ATE task. It employs aspect term span boundaries to detect aspect terms, not viewing ATE as the sequence tagging problem. There are two pointers, namely the start position and the end position, in a span boundary. The BERT model is employed as the backbone network in SPAN, and it can identify multiple aspect terms in a sentence at the same time [56].
11. BERT-BiGRU-CRF: BERT-BiGRU-CRF employs the BERT, the single-layer bidirectional GRU, and CRF for ATE.

The architectures of CRF, WDEmb, Bi-LSTM, BiLSTM-CNN-CRF, HAST, and Seq2seq4ATE are the same as the architectures that are presented in Ma et al. [55]. The architectures of DE-CNN and DOER are in line with the architectures that are described in Luo et al. [8]. The architectures of TAG and SPAN are as with the architectures that are described in Hu et al. [56].

Overall results

The experimental results of the developed method and other baselines are reported in this section. Besides, this paper also demonstrates the effect of the proposed method and provides a comprehensive analysis to explain why the proposed method can perform better than other baselines.

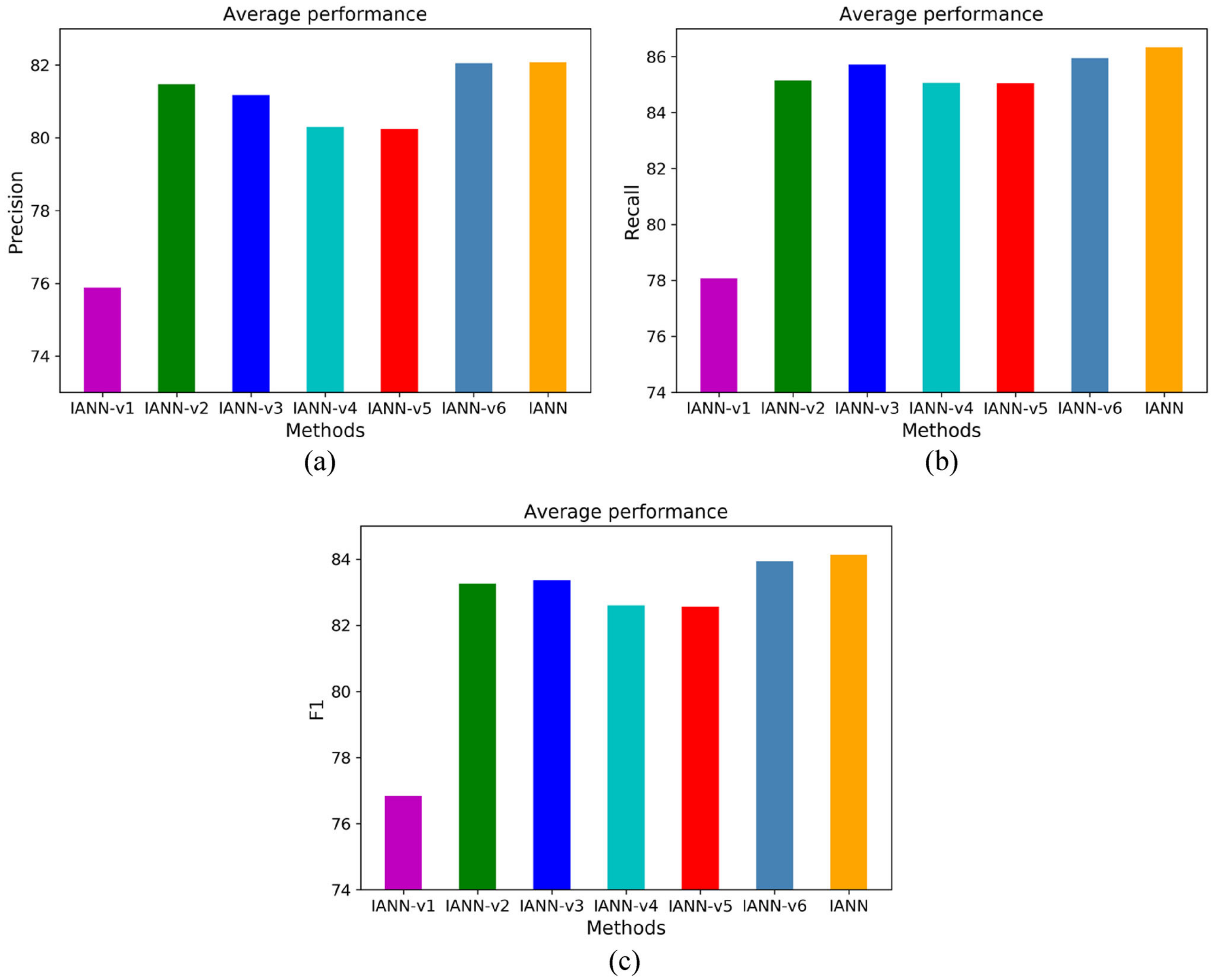
Results of IANN variants

The ablation experiments are used to verify the efficacy of the important modules or layers of IANN. Specifically, six variants are used to compare with IANN. The corresponding results of these variants toward each dataset are reported in Table 3, and the average performance of each variant toward all datasets is shown in Fig. 7.

As shown in Table 3, IANN acquires the best precision, the best recall, and the best F1 scores in most of the datasets. The paired *t* test has been performed in Table 3, and the results are $p < 0.05$. In Fig. 7, the experimental results show that IANN acquires the highest average precision, recall, and F1 values on all datasets. All these experiments validate the effectiveness of the important modules and layers that are designed in the proposed IANN model.

The difference between IANN and IANN-v1 is that IANN-v1 does not consider the dynamic meaning of words in different contexts. The significant performance degradation from IANN in precision, recall, and F1 indicates that capturing the dynamic meaning of the words is very important in ATE. Besides, it proves that the contextualized embedding layer in IANN successfully models the dynamic meaning of the words and improves the model performance.

IANN-v2, IANN-v3, and IANN-v4 verify the efficacy of MCRN. IANN-v2 throws away the multiple convolution operations, it only employs the bidirectional GRU to capture the long-term dependency and encode sequence information in the encoder. When comparing IANN-v2 and IANN, the performance of IANN-v2 drops on all datasets. IANN-v2 cannot integrate information of surrounding words toward a specific word and detect the local features in a sentence. Even if the information of surrounding words toward a specific word is captured and the local features are detected, the performance of the model may be unsatisfactory. The performance degradation of IANN-v3 validates this view,



(a)

(b)

(c)

Fig. 7 The average performance of variants of IANN on all datasets. **a** Shows the average precision, **b** shows the average recall, **c** shows the average F1 score

the information of surrounding words and local features are considered in IANN-v3, but it does not consider modeling the sequence information and capturing long-distance dependencies. There are many alternatives such as transformers on top of BERT. However, their performance does not perform as well as MCRN. This is illustrated by IANN-v4. IANN employs MCRN to model the more informative information. MCRN can not only integrate the information of surrounding words toward a specific word and detect the local features but also model the sequence information and capture long-distance dependencies.

IANN-v5 demonstrates that using the forward last hidden state of the bidirectional GRU in MCRN as the initializing the decoder is more effective than concatenating the average pooling of the forward outputs of the encoder, label embedding, and outputs of the encoder as the inputs of the decoder. That is to say, the method which is employed in IANN is more appropriate than the average method which is employed in IANN-v5 in terms of capturing the overall semantics of a sentence for the decoder.

IANN-v6 validates the validity of the proposed AO tags. IANN-v6 employs the BIO tags in the decoding stage. As shown in Table 3, IANN-v6 achieves the second-best performance on all datasets. Besides, the experimental results in Fig. 7 also show that IANN-v6 acquires the second-best results on the average performance of all datasets. On the one hand, the experimental results fully prove that the AO tags are less challenging than BIO tags in the ATE task, the model performance can be improved by the AO tags to some extent. On the other hand, compared to other variants of IANN, the experiments further imply the significance of the designed contextualized embedding layer and the MCRN model.

Table 3 shows that IANN-v2 outperforms IANN-v3 on the restaurant dataset and twitter dataset. The main reason is that sequence information and long-distance dependencies are more important in the ATE task. Capturing the word order information and long-distance dependencies can provide more useful information to determine whether a word is an aspect term in the ATE task. However, the performance of IANN-v3 is more than the performance of IANN-v2 in the laptop dataset which is shown in Table 3, and IANN-v3 also achieves better average performance than IANN-v2 which is presented in Fig. 7. As shown in Table 2, the scale of the laptop dataset is relatively small, in this scenario, modeling the information of surrounding words toward a word can provide more useful information for ATE than the information provided by modeling sequence information and capturing long-distance dependencies. Therefore, integrating the information of surrounding words is a powerful method in the ATE task. Besides, the maximum word number of a specific aspect term within the laptop dataset is less than the maximum word number of a specific aspect term within the restaurant dataset and twitter dataset. Hence, the importance degree of long-term memory in the laptop dataset is less important than the importance degree in the restaurant and twitter dataset.

As can be seen from Table 3, IANN-v1 obtains the worst performance in the twitter dataset than the performance of IANN-v1 in the laptop and restaurant datasets. The main reason why it performs worst on the twitter dataset is that the sentence in twitter is casual and the meaning of words in a sentence highly depends on the context. The pre-trained word embeddings in IANN-v1 are fixed and cannot capture contextualized information. In other words, modeling the dynamic word sense by employing contextualized word embedding can effectively alleviate the problem in IANN-v1. This is demonstrated by the performance of other variants and the proposed IANN.

Results of state-of-the-art methods

To be consistent with the previous works which generally employ F1 as the metric, this paper presents F1 values of the SOTA baselines and our model with the BIO tags, named IANN-BIO. The results of TAG and SPAN using their codes with the base type of BERT are shown in Table 4. The F1 values of DE-CNN and DOER are reported from Luo et al. [8]. The results of the other state-art-the-art methods are reported from Ma et al. [55] where they do not report twitter and restaurant collection from 2014, 2015, and 2016 datasets. The results of these baselines and the novel IANN with BIO tags can be seen in Table 4. IANN-BIO means that the proposed IANN employs the BIO tagging scheme in the decoding state. As can be seen from Table 4, the IANN-BIO model acquires the best F1 score on all datasets when the IANN-BIO model is compared with other SOTA methods. IANN-BIO outperforms other baselines on all datasets by around 0.22–1.85% than other methods. For the reason that some methods only report F1 values on the laptop dataset in Table 4, hence, this paper performs the paired t test about DE-CNN, DOER, TAG, BERT-BiGRU-CRF, SPAN, and IANN-BIO. The values of paired t test between these baselines and the proposed IANN are all $\text{p} < 0.05$ in Table 4. The <precision, recall> pairs of TAG respectively are <84.76, 85.96>, <85.04, 90.17> and <75.69, 76.53> on laptop, restaurant, and twitter dataset. The <precision, recall> pairs of BERT-BiGRU-CRF respectively are < 84.73, 86.12>, <85.16, 90.21>, and <75.72, 76.58> on laptop, restaurant, and twitter dataset. The <precision, recall> pairs of SPAN respectively are < 84.64, 85.17>, <83.76, 86.76>, and <78.24, 73.84> on laptop, restaurant, and twitter dataset. The <precision, recall> pairs of IANN-BIO respectively are <86.40, 88.15>, <85.25, 91.32>, and <74.50, 78.34> on laptop, restaurant, and twitter dataset.

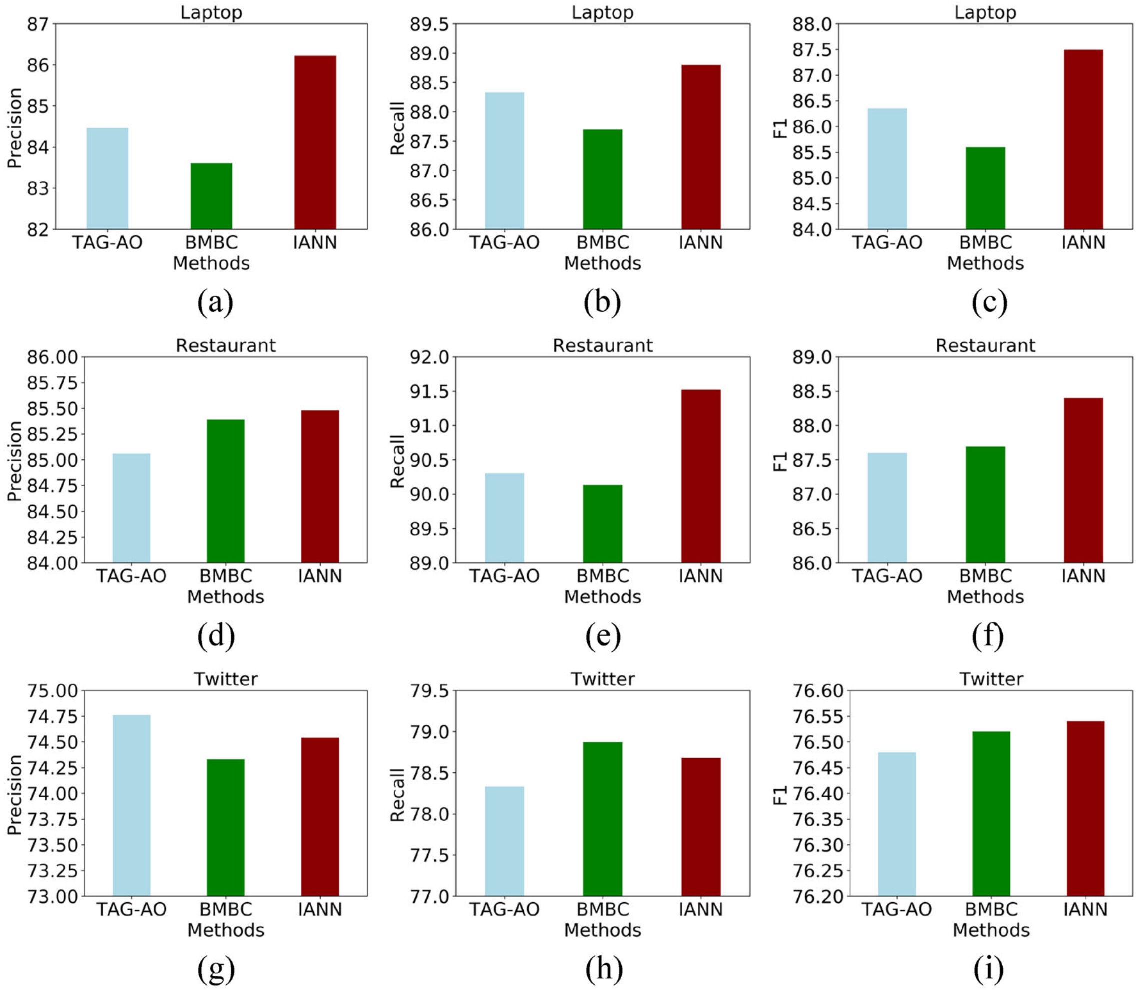
Table 4 The F1 values of the baselines and the proposed method

Method	Laptop	Restaurant	Twitter
CRF	74.01	–	–
Bi-LSTM	75.25	–	–
WDEmb	75.16	–	–
BiLSTM-CNN-CRF	77.80	–	–
DE-CNN	81.26	78.98	63.23
HAST	79.52	–	–
Seq2seq4ATE	80.31	–	–
DOER	82.61	81.06	71.35
TAG	85.36	87.53	76.07
BERT-BiGRU-CRF	85.42*	87.61*	76.15*
SPAN	84.91	85.24	75.78
IANN-BIO	**87.27**	**88.18**	**76.37**

The best result is expressed in bold, and the second-best result is indicated by an asterisk. The symbol '–' denotes that the corresponding experiment is not performed in the original paper

The additional experiments are also performed on the proposed AO tags, the methods used for comparison are TAG-AO, BERT-Multilayer BiGRU-CRF (BMBC), and IANN. The TAG model with AO tags is abbreviated as TAG-AO. The BMBC model consists of BERT, multilayer bidirectional GRU, and CRF. The number of layers of BMBC and IANN is identical. The experimental results of these methods in all datasets are illustrated in Fig. 8. As shown in Fig. 8, IANN still acquires the best results in all datasets with AO tags.

The first reason that our method can achieve superior performance is that this paper explicitly models the dynamic meaning of words in a sentence by the contextualized embedding layer. The contextualized word embeddings generated by the contextualized embedding layer can provide more precise and beneficial information to the next layers or modules, the beneficial information can enhance the model performance. The experimental results of IANN-v1 validate



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Fig. 8 The performance of different methods on three datasets with AO tags. **a**–**c** Show the performance in the laptop dataset, **d**–**f** show the performance in the restaurant dataset, **g**–**i** show the performance in the twitter dataset

the efficacy of the contextualized embedding layer and our assumption.

The second reason that our model can achieve superior performance is that MCRN can learn more informative information. The MCRN model can not only integrate the information of surrounding words toward a word and detect the local features but also model the sequence information and capture long-distance dependencies. In the encoder, these abilities are important for the model. Therefore, the MCRN model can capture more informative information. The baseline model IANN-v2 eliminates the multiple convolution operations, which leads to a model ignoring integrating the information of surrounding words and detecting the local features, hence, resulting in poor performance. Also, the baseline model IANN-v3 removes the bidirectional GRU, which leads to the model lacking the ability to model sequence information and capture long-distance dependencies.

To further validate the effectiveness of MCRN toward capturing long-distance dependencies and integrating the information of surrounding information, the BMBC model and IANN on the AO tags are compared in detail. The BMBC model consists of BERT, multilayer bidirectional GRU, and CRF. The number of layers in the multilayer bidirectional GRU of BMBC is the same as the number of layers in the multilayer MCRN of IANN. The performance of different methods can be seen in Fig. 8. The experimental results show that IANN outperforms BMBC in Fig. 8, and these results also provide sounder evidence for verifying the efficacy of MCRN in IANN.

The third reason that our model can achieve superior performance is because of the multi-layer structure in the encoder. The multi-layer MCRN can learn the more informative higher-order features, which are more appropriate representations for the ATE task. To further validate the validity of the multilayer architecture in the proposed model. The IANN model is compared with other multilayer architecture

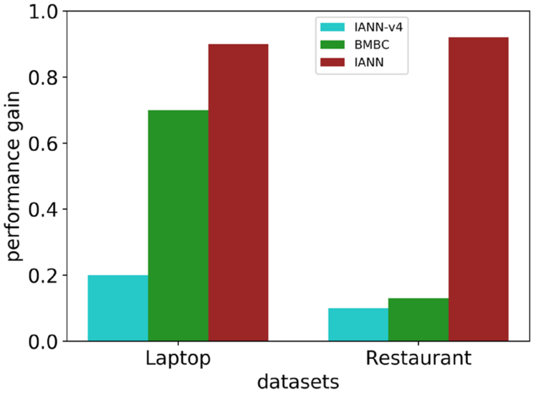


Fig. 9 The performance gain of different models with multilayer architecture in the laptop dataset and restaurant dataset

models, such as BERT-Multilayer BiGRU-CRF (BMBC) and IANN-v4, in terms of performance gain in different layers on the laptop and restaurant datasets under the AO tagging scheme. In IANN-v4, it replaces the multiple convolution and recurrence operations of MCRN with multilayer transformers. The BMBC model is an alternative sequence labeling method. The performance gain of the model is calculated from 1 to 2 layers and 6 layers on the laptop and restaurant datasets, respectively. Because the performance gain of IANN-v4 is negative, the absolute value of the performance gain of IANN-v4 is used. Figure 9 reports the experimental results of performance gain. The IANN model acquires the best performance gain when compared with other models. It also demonstrates the influence of higher-order features captured by IANN. The main reason that the multilayer architecture designed in the proposed model can acquire the best performance gain is that MCRN can encode the more informative information and integrate information of surrounding words in a sentence, as well as the multi-layer MCRN can further improve the model's ability to encode information. Hence, the proposed model's performance with multilayer architecture can acquire better performance than other models.

This paper also reports the proposed model's performance on both AO tags and BIO tags. Table 5 reports the corresponding results. It can be seen from Table 5 that IANN with AO tags outperforms IANN with the BIO tags (IANN-BIO) in all datasets. The important reason is that AO tags are a less challenging task. To further validate the efficacy of AO tags, additional experiments toward TAG on both AO tags and BIO tags are performed. The compared models are TAG and TAG-AO. TAG consists of a BERT model and a CRF. The labeling scheme of TAG is the BIO tags. The structure of TAG-AO is the same as the TAG, but the labeling scheme of TAG-AO is the AO tags. The precise, recall and F1 of both models on three datasets are shown in Fig. 10. The AO tags roundly outperform the BIO tags on three datasets. The TAG-AO model respectively exceeds the TAG by 0.99 F1, 0.15 F1, and 0.41 F1 in the laptop dataset, restaurant dataset, and twitter dataset. The experimental results again prove the validity of the AO tags.

To make the architecture of the model clearer, the hyperparameters and settings of some baselines mentioned above are presented in Table 6. The notation "n_layers" represents the number of layers of the multilayer architecture in the model. The notation "Lap" denotes the laptop dataset, "Res" denotes the restaurant dataset, and "Twi" denotes the twitter dataset.

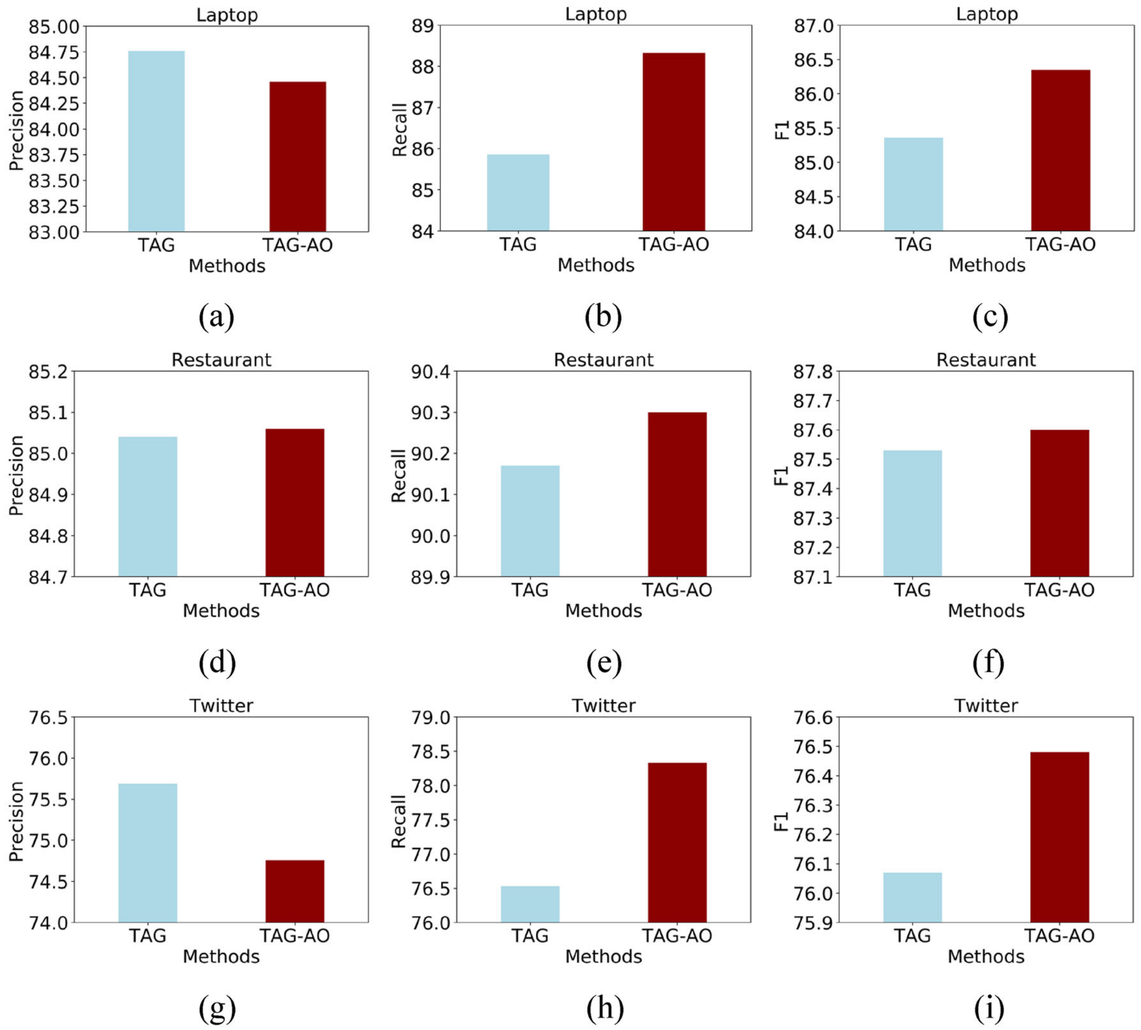
Analysis of the effect of sentence lengths

To further study the proposed model's performance and further reveal the potency of the proposed model, additional experiments concerning different sentence lengths on laptop and restaurant datasets are performed. Figure 11 reports the corresponding results, the models to be compared with the proposed model are the TAG and SPAN models. The experimental results of TAG and SPAN concerning different sentence lengths are reported from the original paper. From Fig. 11a, the experimental results show that the proposed IANN with the BIO tagging scheme (IANN-BIO) outperforms other baselines by 1.07 F1 and 3.81F1 when the sentence length is less than 40 and surpasses TAG by 5.69 F1 when the length exceeds 40 on laptop dataset. As can be seen from Fig. 11b, the IANN-BIO model surpasses other models by 1.98 F1, 4.61 F1, and 14.14 F1 about all sentence lengths on the restaurant dataset. The performance of the TAG and SPAN models dramatically decreases as the sentence length increases, while the IANN-BIO model can achieve the more robust performance than other models for

Table 5 The experiments of the proposed model on both AO tags and BIO tags

Method	Laptop			Restaurant			Twitter		
	P	R	F1	P	R	F1	P	R	F1
IANN-BIO	**86.40**	88.15	87.27	85.25	91.32	88.18	74.50	78.34	76.37
IANN	86.22	**88.80**	**87.49**	**85.48**	**91.52**	**88.40**	**74.54**	**78.68**	**76.54**

The precision is denoted by "P", the recall is denoted by "R", and the F1 value is denoted by "F1". The best result is expressed in bold



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Fig. 10 The performance of models on the laptop, restaurant, and twitter datasets

Table 6 The hyper-parameters and settings of the methods

Methods	Learning rate	Regularization	n_layers			Tagging scheme
			Lap	Res	Twi	
IANN-v1	2e−5	Dropout	2	6	1	AO tags
IANN-v2	2e−5	Dropout	2	6	1	AO tags
IANN-v3	2e−5	Dropout	2	6	1	AO tags
IANN-v4	2e−5	Dropout	2	6	1	AO tags
IANN-v5	2e−5	Dropout	2	6	1	AO tags
IANN-v6	2e−5	Dropout	2	6	1	BIO tags
TAG	2e−5	Dropout	–	–	–	BIO tags
BERT-BiGRU-CRF	2e−5	Dropout	–	–	–	BIO tags
SPAN	2e−5	Dropout	–	–	–	span boundaries
IANN-BIO	2e−5	Dropout	2	6	1	BIO tags
TAG-AO	2e−5	Dropout	–	–	–	AO tags
BMBC	2e−5	Dropout	2	6	1	AO tags
IANN	2e−5	Dropout	2	6	1	AO tags

The symbol '–' denotes that the model is the signal layer architecture

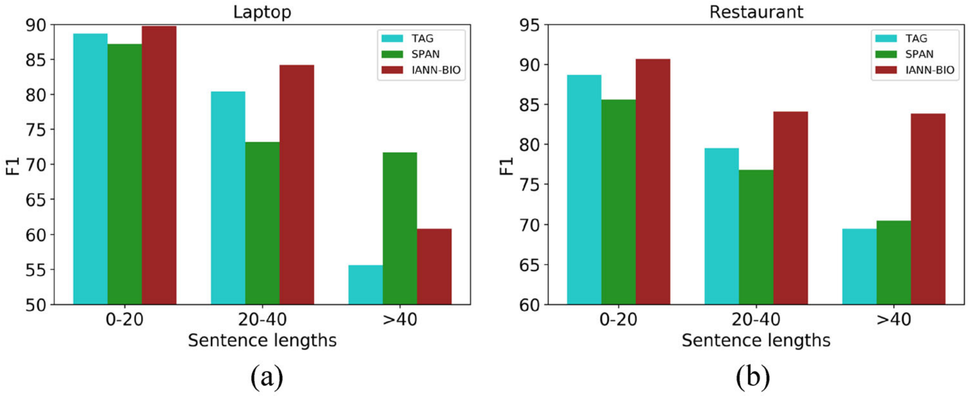


Fig. 11 F1 on laptop and restaurant w.r.t different sentence lengths. **a** Reports the results in the laptop dataset, **b** reports the results on the restaurant dataset

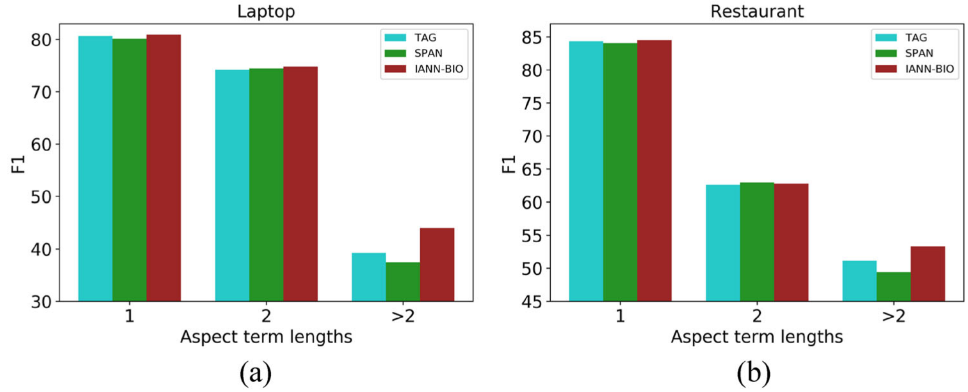


Fig. 12 F1 on laptop and restaurant w.r.t different aspect term lengths. **a** Reports the results in the laptop dataset, **b** reports the results in the restaurant dataset

short and long sentences. The experimental results verify that the developed method is suitable for short and long sentences due to the advantages that it can consider the dynamic word meaning, model the more informative information, and integrate the information of surrounding words.

Besides, from Fig. 11a, the experimental results show that SPAN outperforms IANN-BIO when the sentence length exceeds 40 on the laptop dataset. The reason is that the scale of the laptop dataset limits the proposed model's performance in the scenario where the sentence length is long, and the size of the laptop dataset is small. Compared to SPAN, the proposed model is complicated and has more parameters. The small dataset results in the proposed model underfitting and causes the proposed model to fail to learn more useful information when the sentence length exceeds 40. The large dataset can alleviate the problem and it is validated by Fig. 11b.

Analysis of the effect of aspect term lengths

To gain more insights on performance improvements of the proposed model, additional experiments concerning different aspect term lengths on the restaurant dataset and the laptop dataset are performed. Figure 12 reports the experimental results. The compared state-of-the-art methods are TAG and SPAN. From Fig. 12, the experimental results show that the performance of TAG and SPAN significantly decreases as the aspect term length becomes longer, while our proposed model is more robust for long aspect terms. The main reason is that other baselines are weak in encoding the more informative information and integrating surrounding information toward a specific word when they identify long aspect terms. Our proposed model, on the contrary, can naturally alleviate such problems because MCRN can effectively encode the more informative information, which can not only integrate the surrounding information toward a specific word and detect the local features but also model the sequence information and capture the long-distance dependencies.

Effect of the number of layers in multi-layer encoder

To analyze the proposed model's performance concerning different numbers of layers of the multi-layer framework of the proposed IANN model and how to decide the optimal number of layers of the multi-layer encoder for the ATE task. Additional experiments toward different numbers of layers in the multi-layer encoder of IANN are performed to validate the influence of different numbers of layers of IANN on the

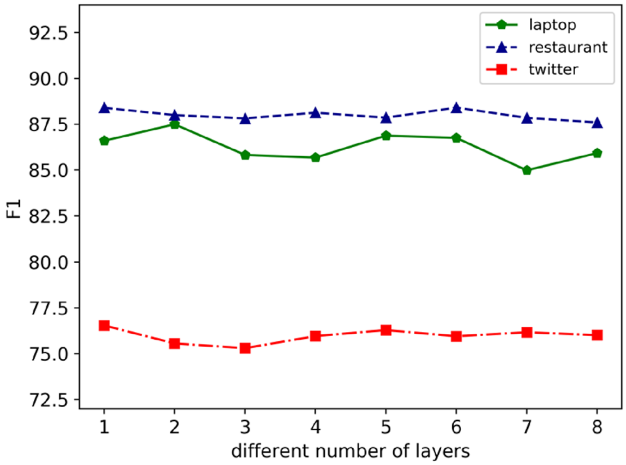


Fig. 13 F1 of IANN toward different numbers of layers on three datasets

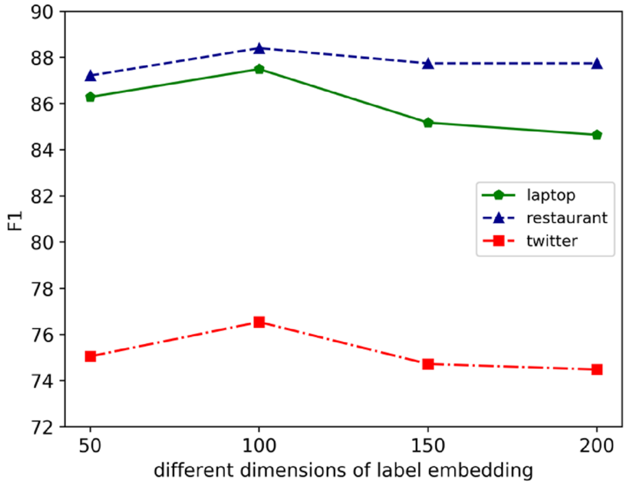


Fig. 14 F1 of IANN with respect to the different dimensions of label embedding on three datasets

twitter dataset, the restaurant dataset, and the laptop dataset. The corresponding results are shown in Fig. 13.

Figure 13 shows that different numbers of layers indeed have different effects on the proposed model's performance. The optimal numbers of layers are different in different types of datasets. In the laptop dataset and the restaurant dataset, the optimal number of layers is 2 and 6, respectively. When the model has fewer layers, dependency information and surrounding information will not be effectively learned in the model. Moreover, the model will suffer from over-fitting and generating redundant information when the number of layers is too large. In the twitter dataset, the number of layers has little effect on model performance. The optimal number of layers is 1 in the twitter dataset. The model is easy to overfit and learn noise information in the twitter dataset, which decreases model performance.

Effect of the dimension of the label embedding

For the purpose of researching the effect of the dimension of the label embedding, experiments with different dimensions of label embeddings are performed on the laptop dataset, restaurant dataset, and twitter dataset. The dimensions to be compared are 50, 100, 150, and 200 respectively. Figure 14 reports the corresponding results.

As shown in Fig. 14, the dimension of label embedding has a certain degree of influence on the proposed IANN. The optimal dimension of label embedding is 100 in all datasets. From Fig. 14, the experimental results show that there are two obvious trends. The performance increases as the dimension increases when the dimension is less than 100, and the performance decreases as the dimension increases when the dimension is greater than 100. The main reason is that the smaller dimension does not learn enough information and the larger dimension makes the model overfit.

Effect of different sizes of the filter

In order to show the influence of the size of the convolution kernel (i.e., filter), this paper performs experiments concerning the size of the filter in the multiple convolution operations. Different sizes of the multiple convolutions are divided into two groups, Group 1 involves Conv2d-1, Group 2 involves Conv2d-2, 3, 4, and 5. The size of each of Group 2 is the same. The sizes to be compared respectively are 3, 5, 7, and 9. The experiments are performed on the laptop and restaurant datasets. The corresponding results are reported in Fig. 15.

As shown in Fig. 15a, the proposed IANN acquires the best results when the size of the convolution kernel in Group 1 is 3 on both datasets. It can be seen from Fig. 15b, the proposed IANN acquires the best result when the size of the convolution kernel in Group 2 is 5 on both datasets. Hence, the default sizes of the convolution kernel of Group 1 and Group 2 respectively are 3 and 5 in the proposed model.

Time complexity

Accurate analysis of the time complexity of the model is very challenging. For the purpose of incarnating the time complexity of different methods, this paper performs additional experiments concerning the running time of some methods, such as Seq2Seq4ATE, TAG, SPAN, and IANN-BIO with a single layer encoder, on the laptop dataset. The training environment of all models is the same, Nvidia RTX 2080Ti GPU is used in the experiments.

The experiments of the runtime of some state-of-the-art methods in the training stage on BIO tags are reported in Fig. 16. As shown in Fig. 16, the runtime of the proposed method is 7 s less than Seq2Seq4ATE because the latter has extra complex attention and gated unit computing operations.

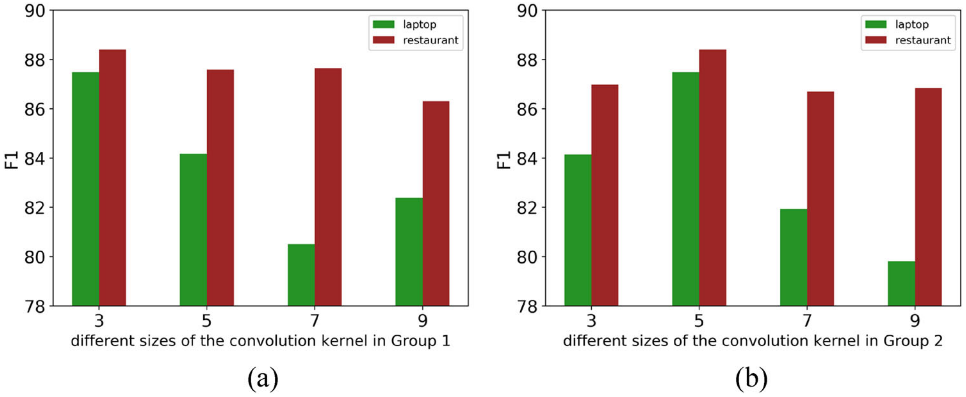


Fig. 15 F1 w.r.t different sizes of the convolution kernel in Group 1 and Group 2. **a** Reports the results in Group 1, **b** reports the results in Group 2

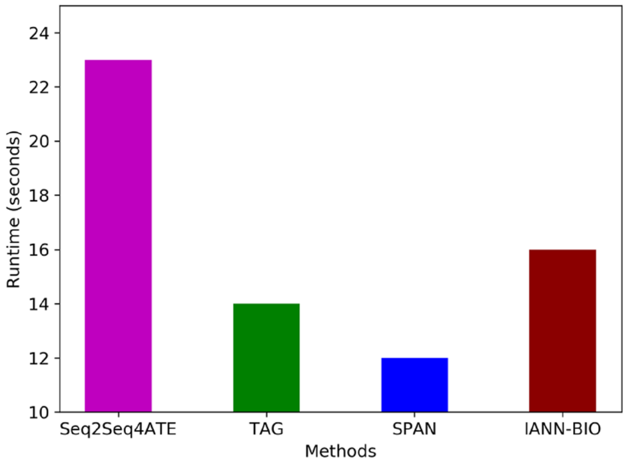


Fig. 16 Running time of one epoch of different methods in the training stage on the laptop dataset

Although the proposed model has complex operations in the GRU unit along the sequential sequence, it still achieves comparable results compared with other methods that are not based on GRU, such as TAG and SPAN.

Case study

Table 7 presents some cases sampled from the state-of-the-art methods such as TAG, SPAN, and IANN-BIO. As observed in the first two examples, TAG incorrectly predicts the aspect terms by missing the phrase "-year computer accidental protection warranty", and the predictions of SPAN are all wrong. The main reason is that models such as TAG and SPAN are less effective when dealing with the long aspect terms. However, the proposed model is more robust about the long aspect terms. In the scenario of long sentences, such as sentence 2 and sentence 3, the TAG and SPAN methods also perform worse. They are less able to deal with long sentences. Moreover, as observed in the last examples, TAG and SPAN sometimes predict redundant results (e.g., "space" in sentence 4). The main reason is that TAG and SPAN consider the noise information, encode the redundant information, and lack the ability to encode the more informative information. However, the proposed model can effectively alleviate these problems by employing the multi-layer MCRN model in the encoder.

Discussion

To give deeper insights into our obtained results, how to improve the model performance is discussed in detail and the reasons why the proposed model can achieve better results than other models for identifying aspect terms are explained in this section. The important ideas to alleviate the shortcomings of the current sequence-to-sequence learning and fixed word representations are that the sequence-to-sequence model should consider the more informative information and can model dynamic word meaning in a sentence for identifying aspect terms.

The first reason that the proposed model acquires better results than the previous seq2seq-based models (e.g., Seq2Seq4ATE) is that a contextualized embedding layer to model dynamic word meaning is designed in a sentence. A word has different meanings or parts of speech in different contexts, the meaning or part of speech of the word plays an important role in determining if the word is an aspect term. Hence modeling the dynamic word meaning can provide supplementary information for the learning of the model and improve model performance. The word embeddings can map a word into an embedding space where the representation of the word is a continuous, dense, real-valued vector. The distributional semantic of a word can be captured in the embedding space, that is to say, the meaning and syntax characters of a word can be reflected through vectors in the embedding space. For example, in the specific embedding space, the cosine similarity between two word vectors can reflect their semantic similarity to a certain extent. However, in the static word embedding space, the representation of a specific word within a sentence is fixed regardless

Table 7 The predictions of different methods

Sentence	TAG	SPAN	IANN-BIO
1. But the **mountain lion** is just too slow	Mountain lion	None (✘)	Mountain lion
2. I opted for the **SquareTrade 3-Year Computer Accidental Protection Warranty** ($1500–2000) which also support "accidents" like drops and spills that are NOT covered by **AppleCare**	Squaretrade 3 (✘), applecare	Accidental protection warranty (✘), applecare	Squaretrade 3-year computer accidental protection warranty, applecare
3. I work as a designer and coder and I needed a new buddy to work with, not **gaming**	Coder (✘), gaming	Coder (✘), gaming	Gaming
4. The smaller **size** was a bonus because of space restrictions	Size, space (✘)	Size, space (✘)	Size

The gold results are marked in bold. Incorrect predictions are marked with ✘

of the context. The fixed word embedding is weak at capturing the different meanings of a word. The contextualized word embedding can alleviate the problem, it can capture the dynamic meaning of a word. Because contextualized word embedding can map a word within a sentence into different representations by the context.

Besides, contextualized word embedding can alleviate the word sense ambiguity in some degree. The base type of BERT is employed in the contextualized embedding layer and is pre-trained on a large open-domain dataset where a word may appear in a different context. In the distributional hypothesis, the word meaning can be decided by the context. Hence the pre-trained BERT can capture the generally different meaning of a word by the context to some extent. Besides, BERT is fine-tuned in a specific domain during the training of IANN. For the meaning of a word in the specific domain, it can be modeled by fine-tuning on the specific domain to a certain extent. Hence, the contextualized embedding layer of IANN can capture the word sense by the context to some extent.

The second reason that the proposed model acquires better results than the previous seq2seq-based models is that this paper proposes MCRN to model the more informative information. The MCRN model can not only integrate surrounding information toward a word and detect the local features but also model the sequence information and capture the long-term dependency. The multiple convolutional operations are used to integrate surrounding information toward a specific word and detect the local features. Different sizes of the filter determine different scopes of the surrounding words toward a specific word. Different convolution kernels can capture the different scopes of the surrounding words toward a specific word. Hence, multiple convolutional operations can model more informative surrounding information. The bidirectional GRU in MCRN can capture the long-term dependency and model the sequence information. All these characters enable MCRN to better learn the more informative information in the encoder and they are not available in vanilla CNN and GRU. Besides, the experimental results in Fig. 8 also provide sounder evidence and validate the validity of MCRN.

The third reason that the proposed model acquires better results than the previous seq2seq-based models is that the multi-layer MCRN is employed as the multi-layer encoder in our model. The multi-layer MCRN can learn the more abstract and higher-order features as the number of layers increases, which may generate more appropriate representations for the ATE task. The experimental results in Fig. 9 also demonstrate the validity of multi-layer MCRN in the proposed method.

The reason why the proposed AO tagging scheme performs better than the BIO tagging scheme is that the two-label-based AO tags are less challenging than the three-label-based BIO tags in the decoding stage. When compared with the BIO tags, the number of predicted labels of the AO tags in each state is reduced from three to two. The performance of the model can be effectively enhanced by reducing the number of predicted labels of a specific word. Besides, the AO tags can also be used in other seq2seq-based or sequence labeling-based models.

The first reason that the proposed model acquires better results than the BERT-based models is that the proposed model can take full advantage of the features generated from

BERT by the seq2seq learning in the ATE task. The contextualized word embeddings are not directly sent into performing the labeling task, but rather they are viewed as the basic representations, and then the labeling task is performed through the seq2seq learning framework. In the encoder of the proposed model, MCRN can not only integrate the surrounding information toward a word and detect local features but also capture the long-term dependencies and model the word order information in a sentence. These functions are not available in the previous BERT-based methods.

The second reason that the proposed model acquires better results than the BERT-based models is that seq2seq learning can fully utilize the overall semantic of a sentence and is more conducive to dealing with dependencies between labels. The overall meaning of a whole sentence is crucial for determining whether a word is an aspect term. Hence, the performance of the model can be enhanced by capturing dependencies between labels and taking advantage of the overall semantic of a sentence.

Conclusion

This paper proposes an IANN, which is a novel sequence-to-sequence learning framework, for the ATE task. The problems existing in the previous sequence-to-sequence learning and fixed word embedding in the ATE task can be effectively alleviated by the IANN model. The IANN model has two important layers or modules, they are the contextualized embedding layer which consists of BERT, and the MCRN model which consists of multiple convolution and recurrence operations. Besides, this paper also proposes a less challenging tagging scheme, named AO tags, which consists of "Aspect" and "Outside" labels. The contextualized embedding layer is designed to model the dynamic word meaning. BERT is employed to generate the contextualized word embeddings in the contextualized embedding layer. The MCRN model is designed to model the more informative information. It can not only integrate surrounding information toward a word and detect the local features but also capture the long-distance dependencies and model the sequence information. Besides, the multi-layer MCRN can learn the more informative higher-order features. Three widely used datasets are employed to perform abundant experiments to validate the performance and generalization of the proposed IANN. The experimental results show that our model acquires state-of-the-art results. Moreover, the number of layers in the multi-layer encoder is given according to different types of datasets. In the next work, the proposed model can be tried to apply to other sequence labeling tasks. Besides, common knowledge can be incorporated into the proposed model as an external learning source.

Funding This work was supported by Beijing Nova Program (Grant no. Z211100002121120) from Beijing Municipal Science & Technology Commission.

Declarations

Conflict of interest The authors declare no competing interests.

References

1. Zhou J, Chen Q, Huang JX et al (2020) Position-aware hierarchical transfer model for aspect-level sentiment classification. Inf Sci 513:1–16
2. Chen H, Zhai Z, Feng F et al (2022) Enhanced multi-channel graph convolutional network for aspect sentiment triplet extraction. In: Proceedings of the 60th annual meeting of the association for computational linguistics (volume 1: long papers), pp 2974–2985
3. Yadav RK, Jiao L, Goodwin M et al (2021) Positionless aspect based sentiment analysis using attention mechanism. Knowl Based Syst 226:107136
4. Cai H, Xia R, Yu J (2021) Aspect-category-opinion-sentiment quadruple extraction with implicit aspects and opinions. In: Proceedings of the 59th annual meeting of the association for computational linguistics and the 11th international joint conference on natural language processing (volume 1: long papers), pp 340–350
5. Jin W, Ho HH, Srihari RK (2009) A novel lexicalized HMM-based learning framework for web opinion mining. In: Proceedings of the 26th annual international conference on machine learning
6. Li F, Han C, Huang M et al (2010) Structure-aware review mining and summarization. In: Proceedings of the 23rd international conference on computational linguistics (Coling 2010), pp 653–661
7. Xu H, Liu B, Shu L et al (2018) Double embeddings and CNN-based sequence labeling for aspect extraction. In: Proceedings of the 56th annual meeting of the association for computational linguistics (volume 2: short papers), pp 592–598
8. Luo H, Li T, Liu B et al (2019) DOER: dual cross-shared RNN for aspect term-polarity co-extraction. In: Proceedings of the 57th annual meeting of the association for computational linguistics, pp 591–601
9. Gers FA, Schmidhuber J, Cummins F (1999) Learning to forget: continual prediction with LSTM
10. Cho K, van Merriënboer B, Gulcehre C et al (2014) Learning phrase representations using RNN encoder–decoder for statistical machine translation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pp 1724–1734

11. Vaswani A, Shazeer N, Parmar N et al (2017) Attention is all you need. In: Advances in neural information processing systems, pp 5998–6008
12. Liu N, Shen B (2020) Aspect-based sentiment analysis with gated alternate neural network. Knowl Based Syst 188:105010
13. Devlin J, Chang M-W, Lee K et al (2018) Bert: pre-training of deep bidirectional transformers for language understanding. http://arxiv.org/abs/1810.04805
14. Liu N, Shen B, Zhang Z et al (2019) Attention-based sentiment reasoner for aspect-based sentiment analysis. HCIS 9(1):35
15. Liu N, Shen B (2020) ReMemNN: a novel memory neural network for powerful interaction in aspect-based sentiment analysis. Neurocomputing 395:66–77
16. Hu M, Liu B (2004) Mining and summarizing customer reviews. In: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, pp 168–177
17. Do HH, Prasad P, Maag A et al (2019) Deep learning for aspect-based sentiment analysis: a comparative review. Expert Syst Appl 118:272–299
18. Chen Z, Qian T (2020) Relation-aware collaborative learning for unified aspect-based sentiment analysis. In: Proceedings of the 58th annual meeting of the association for computational linguistics, pp 3685–3694
19. Rana TA, Cheah Y-N (2016) Aspect extraction in sentiment analysis: comparative analysis and survey. Artif Intell Rev 46(4):459–483
20. Tubishat M, Idris N, Abushariah MA (2018) Implicit aspect extraction in sentiment analysis: review, taxonomy, oppportunities, and open challenges. Inf Process Manage 54(4):545–563
21. Liu B, Hsu W, Ma Y (1998) Integrating classification and association rule mining. In: KDD, pp 80–86
22. Popescu A-M, Etzioni O (2007) Extracting product features and opinions from reviews. In: Natural language processing and text mining. Springer, pp. 9–28
23. Rana TA, Cheah Y-N (2017) A two-fold rule-based model for aspect extraction. Expert Syst Appl 89:273–285
24. Wu C, Wu F, Wu S et al (2018) A hybrid unsupervised method for aspect term and opinion target extraction. Knowl Based Syst 148:66–73
25. Luo Z, Huang S, Zhu KQ (2019) Knowledge empowered prominent aspect extraction from product reviews. Inf Process Manage 56(3):408–423
26. Giannakopoulos A, Musat C, Hossmann A et al (2017) Unsupervised aspect term extraction with b-lstm & crf using automatically labelled datasets. http://arxiv.org/abs/1709.05094
27. Seymore K, McCallum A, Rosenfeld R (1999) Learning hidden Markov model structure for information extraction. In: AAAI-99 workshop on machine learning for information extraction, pp 37–42
28. Shams M, Baraani-Dastjerdi A (2017) Enriched LDA (ELDA): combination of latent Dirichlet allocation with word co-occurrence analysis for aspect extraction. Expert Syst Appl 80:136–146
29. Wan C, Peng Y, Xiao K et al (2020) An association-constrained LDA model for joint extraction of product aspects and opinions. Inf Sci 519:243–259
30. Ozyurt B, Akcayol MA (2021) A new topic modeling based approach for aspect extraction in aspect based sentiment analysis: SS-LDA. Expert Syst Appl 168:114231
31. Shu L, Xu H, Liu B (2017) Lifelong learning CRF for supervised aspect extraction. In: Proceedings of the 55th annual meeting of the association for computational linguistics (volume 2: short papers), pp 148–154
32. Kumar PS (2020) Algorithms for solving the optimization problems using fuzzy and intuitionistic fuzzy set. Int J Syst Assur Eng Manag 11(1):189–222
33. Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems, pp 1097–1105
34. Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780
35. Greff K, Srivastava RK, Koutník J et al (2016) LSTM: a search space odyssey. IEEE Trans Neural Netw Learn Syst 28(10):2222–2232
36. Socher R, Lin CC, Manning C et al (2011) Parsing natural scenes and natural language with recursive neural networks. In: Proceedings of the 28th international conference on machine learning (ICML-11), pp 129–136
37. Poria S, Cambria E, Gelbukh A (2016) Aspect extraction for opinion mining with a deep convolutional neural network. Knowl Based Syst 108:42–49
38. Luo H, Li T, Liu B et al (2019) Improving aspect term extraction with bidirectional dependency tree representation. IEEE/ACM Trans Audio Speech Lang Process 27(7):1201–1212
39. Zhang Z, Rao Y, Lai H et al (2021) TADC: a topic-aware dynamic convolutional neural network for aspect extraction. IEEE Trans Neural Netw Learn Syst
40. Phan MH, Ogunbona PO (2020) Modelling context and syntactical features for aspect-based sentiment analysis. In: Proceedings of the 58th annual meeting of the association for computational linguistics, pp 3211–3220
41. Venugopalan M, Gupta D (2022) An enhanced guided LDA model augmented with BERT based semantic strength for aspect term extraction in sentiment analysis. Knowl Based Syst 246:108668
42. Oh S, Lee D, Whang T et al (2021) Deep context- and relation-aware learning for aspect-based sentiment analysis. In: Proceedings of the 59th annual meeting of the association for computational linguistics and the 11th international joint conference on natural language processing (volume 2: short papers), pp 495–503
43. Lekhtman E, Ziser Y, Reichart R (2021) DILBERT: customized pre-training for domain adaptation with category shift, with an application to aspect extraction. In: Proceedings of the 2021 conference on empirical methods in natural language processing, pp 219–230
44. Nguyen T-N, Nguyen K-H, Song Y-I et al (2021) An uncertainty-aware encoder for aspect detection. In: Findings of the association for computational linguistics: EMNLP 2021, pp 797–806
45. Zhang W, Deng Y, Li X et al (2021) Aspect-based sentiment analysis in question answering forums. In: Findings of the association for computational linguistics: EMNLP 2021, pp 4582–4591
46. Hu M, Zhao S, Guo H et al (2021) Multi-label few-shot learning for aspect category detection. In: Proceedings of the 59th annual meeting of the association for computational linguistics and the 11th international joint conference on natural language processing (volume 1: long papers), pp 6330–6340
47. Tulkens S, van Cranenburgh A (2020) Embarrassingly simple unsupervised aspect extraction. In: Proceedings of the 58th annual meeting of the association for computational linguistics, pp 3182–3187
48. Xu H, Liu B, Shu L et al (2019) BERT post-training for review reading comprehension and aspect-based sentiment analysis. In: Proceedings of the 2019 conference of the north american chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers), pp 2324–2335
49. Chauhan GS, Meena YK, Gopalani D et al (2020) A two-step hybrid unsupervised model with attention mechanism for aspect extraction. Expert Syst Appl 161:113673
50. Zhao H, Huang L, Zhang R et al (2020) SpanMlt: a span-based multi-task learning framework for pair-wise aspect and opinion terms extraction. In: Proceedings of the 58th annual meeting of the association for computational linguistics, pp 3239–3248

51. Chen Y, Zhang Z, Zhou G et al (2022) Span-based dual-decoder framework for aspect sentiment triplet extraction. Neurocomputing 492:211–221
52. Mukherjee R, Nayak T, Butala Y et al (2021) PASTE: a tagging-free decoding framework using pointer networks for aspect sentiment triplet extraction. In: Proceedings of the 2021 conference on empirical methods in natural language processing, pp 9279–9291
53. Li K, Chen C, Quan X et al (2020) Conditional augmentation for aspect term extraction via masked sequence-to-sequence generation. In: Proceedings of the 58th annual meeting of the association for computational linguistics, pp 7056–7066
54. Wang Q, Wen Z, Zhao Q et al (2021) Progressive self-training with discriminator for aspect term extraction. In: Proceedings of the 2021 conference on empirical methods in natural language processing, pp 257–268
55. Ma D, Li S, Wu F et al (2019) Exploring sequence-to-sequence learning in aspect term extraction. In: Proceedings of the 57th annual meeting of the association for computational linguistics, pp 3538–3547
56. Hu M, Peng Y, Huang Z et al (2019) Open-domain targeted sentiment analysis via span-based extraction and classification. In: Proceedings of the 57th annual meeting of the association for computational linguistics, pp 537–546
57. Peters M, Neumann M, Iyyer M et al (2018) Deep contextualized word representations. In: Proceedings of the 2018 conference of the north american chapter of the association for computational linguistics: human language technologies, volume 1 (long papers), pp 2227–2237
58. Radford A, Narasimhan K, Salimans T et al (2018) Improving language understanding by generative pre-training
59. Pontiki M, Galanis D, Pavlopoulos J et al (2014) SemEval-2014 task 4: aspect based sentiment analysis. In: Proceedings of the 8th international workshop on semantic evaluation (SemEval 2014), pp 27–35
60. Pontiki M, Galanis D, Papageorgiou H et al (2015) Semeval-2015 task 12: aspect based sentiment analysis. In: Proceedings of the 9th international workshop on semantic evaluation (SemEval 2015), pp 486–495
61. Pontiki M, Galanis D, Papageorgiou H et al (2016) SemEval-2016 task 5: aspect based sentiment analysis. In: Proceedings of the 10th international workshop on semantic evaluation (SemEval-2016), pp 19–30
62. Mitchell M, Aguilar J, Wilson T et al (2013) Open domain targeted sentiment. In: Proceedings of the 2013 conference on empirical methods in natural language processing, pp 1643–1654
63. Li X, Bing L, Li P et al (2019) A unified model for opinion target extraction and target sentiment prediction. In: Proceedings of the AAAI conference on artificial intelligence, pp 6714–6721
64. Pennington J, Socher R, Manning CD (2014) Glove: global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pp 1532–1543
65. Yin Y, Wei F, Dong L et al (2016) Unsupervised word and dependency path embeddings for aspect term extraction. In: Proceedings of the twenty-fifth international joint conference on artificial intelligence, pp 2979–2985
66. Liu P, Joty S, Meng H (2015) Fine-grained opinion mining with recurrent neural networks and word embeddings. In: Proceedings of the 2015 conference on empirical methods in natural language processing, pp 1433–1443
67. Reimers N, Gurevych I (2017) Reporting score distributions makes a difference: performance study of LSTM-networks for sequence tagging. In: Proceedings of the 2017 conference on empirical methods in natural language processing, pp 338–348
68. Li X, Bing L, Li P et al (2018) Aspect term extraction with history attention and selective transformation. In: Proceedings of the 27th international joint conference on artificial intelligence, pp 4194–4200