QoMRC: Query-oriented Machine Reading Comprehension Framework for Aspect Sentiment Triplet Extraction

Kehang Wang[1], Ye Liu[1], Kai Zhang[1 (✉)], Qi Liu[1,2], Yankun Ren[3], Xinxing Yang[3], Longfei Li[3], and Jun Zhou[3]

[1] State Key Laboratory of Cognitive Intelligence, University of Science and Technology of China
[2] Institute of Artificial Intelligence, Hefei Comprehensive National Science Center
[3] Ant Group
{wangkehang,liuyer,kkzhang0808,qiliuql}@mail.ustc.edu.cn
{yankun.ryk,xinxing.yangxx,longyao.llf, jun.zhoujun}@antgroup.com

Abstract. Aspect Sentiment Triplet Extraction (ASTE) is an essential task in aspect-based sentiment analysis. It aims to extract sentiment triplets from the context, which generally consists of three subtasks: aspect term extraction, opinion term extraction and sentiment classification. Existing methods mainly adopt a bidirectional machine reading comprehension framework to capture corresponding relations among subtasks. However, they input queries for different subtasks into the same encoder simultaneously, which leads the model to confuse the subtask associated with the query. To address this issue, we propose a novel Query-oriented Machine Reading Comprehension (QoMRC) framework which is a two-stage approach. In the first stage, QoMRC utilizes predefined queries and adapter tuning to efficiently generate three different query-oriented adapters for three subtasks that capture task-specific features. In the second stage, we fuse the query-oriented adapter and the shared encoder representation to obtain task-specific representation, which also reserves the correlation among subtasks. In addition, to reduce the semantic gap between the initialized adapters and the pre-trained BERT, we employ a layer-wise distillation approach in the first stage. Extensive experiment results on benchmark datasets show the efficacy of our proposed method, and indicate the necessity of capturing task-specific features.

Keywords: Aspect sentiment triplet extraction · Adapter tuning · Knowledge distillation.

1 Introduction

Aspect Sentiment Triplet Extraction (ASTE) [16], as a variant of the fine-grained Aspect-based Sentiment Analysis (ABSA) [7], has been extensively studied recently [23, 21, 15, 2]. Specifically, ASTE aims to extract the aspect terms (AT) with their corresponding opinion terms (OT) and sentimental polarity (SP) simultaneously. As shown in Figure 1a, ASTE extracts sentiment triplets *(AT, OT, SP)*, such as *(service, slow, negative)* and *(people, friendly, positive)*.

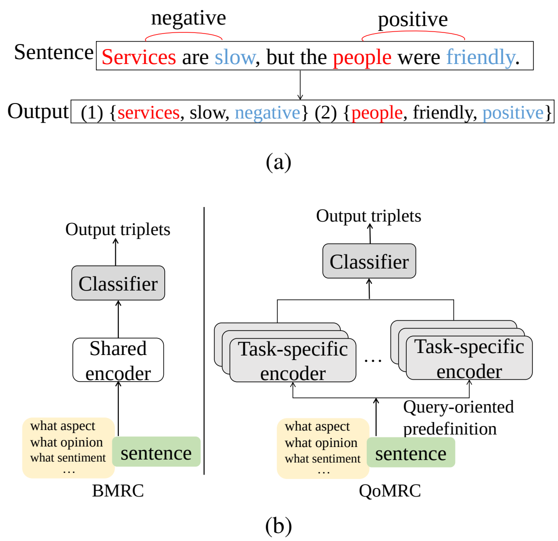

Fig. 1: (a) An illustration of ASTE. (b) An overview of BMRC and QoMRC. The BMRC-based methods (left) input queries for different subtasks into the same encoder simultaneously, which leads the model to confuse the subtasks. Our method (right) utilize predefined queries to guide the model to generate the task-specific representation.

Early methods adopt a pipeline framework [16, 15, 22] extracts all terms (i.e., aspect terms, opinion terms and sentimental polarity) from the given text step by step, and then pairs them. However, these pipeline-based methods ignore corresponding implicit relations between multiple subtasks and could result in the error propagation. In response to this, some researchers formalize ASTE task as a bidirectional machine reading comprehension (BMRC) task [2, 11]. Specifically, BMRC-based methods manually construct different categories of queries for each text, and then utilize these queries to obtain answers. For the example in Figure 1b left, when they ask the model query *what aspect*, the model will give the answers *services* and *people*. When they ask the model query *what opinion*, the model will give the answers *slow* and *friendly*. When they ask the model query *what sentiment about the aspect-opinion pairs*, the model will give the answers *negative* and *positive*.

Although the BMRC-based methods can capture implicit relations between multiple subtasks and could reduce propagation errors, they input queries of different subtasks into the same shared encoder simultaneously which may lead to a lack of task-specific features in the model. This problem makes the model confused about the subtask corresponding to the query. For the example in Figure 1a, when asking the model query: *What aspect given the opinion slow?*, the model may misidentify aspect term extraction as opinion term extraction [25] and give the wrong answer *friendly* instead of *service*. However, if the model is provided with the specific features of each task, it can more accurately determine the corresponding subtask for the given query. Unfortunately, there are two main technical challenges in designing effective solutions to inject task-specific features into the pre-trained model (e.g., BERT). First, it can be challenging to teach abstract task-specific features explicitly to a model. Second, injecting different task features into a pre-trained model can lead to catastrophic forgetting[13], where the model forgets previously learned features while adapting to new tasks.

With the above analysis, in this paper, we propose a novel framework called Query-oriented Machine Reading Comprehension framework (QoMRC) to obtain task-specific representations, as shown in Figure 1b right. Specifically, QoMRC can be regarded a two-stage approach. In the first stage, we predefine five categories of queries for three subtasks: forward aspect query (FAQ) and backward aspect query (BAQ) for aspect term extract (ATE), forward opinion query (FOQ) and backward opinion query (BOQ) for opinion term extract (OTE), and sentiment query (SQ) for sentiment classification (SC). The purpose of these predefined queries is to explicitly represent the task-specific features, thereby enabling the model to learn more effectively. Subsequently, we efficiently train three query-oriented adapters: aspect adapter, opinion adapter and sentiment adapter by using predefined queries. It is worth noting that we utilize adapter tuning [6] which is an parameter-efficient fine-tuning approach to prevent catastrophic forgetting and reduce training time. However, there is a significant semantic gap between pre-trained models (e.g., BERT) and initialized adapters. To address this issue, we employ a technique called layer-wise task-specific knowledge distillation [17, 5, 32]. This technique involves transferring the foundational knowledge learned by the pre-trained model to the adapter. In the second stage, the model selects the corresponding adapter based on the input query and fuses the representations from the adapter's output and the shared encoder's output. During this stage, the parameters of these adapters are frozen, and only the shared encoder is trained. In summary, the main contributions of our work could be summarized as follows.

- We propose a novel QoMRC framework for ASTE task. Our framework is a two-stage method which can inject task-specific features into the pre-trained model (e.g., BERT).
- To bridge the significant semantic gap between the pre-trained model and query-oriented adapters, we employ a technique called layer-wise distillation.
- we conduct extensive experiment results on benchmark datasets, where the experimental results demonstrate the effectiveness of our proposed method.

2 Related Work

2.1 Aspect Sentiment Triplet Extraction

As a fine-grained task of aspect-based sentiment analysis [7, 28–30], aspect sentiment triplet extraction has attracted lots of researchers' interest [23, 21, 15, 2]. It aims to extract aspect terms with their corresponding opinion terms and sentiment polarity. Generally, ASTE can be regarded as a composite task consisting of three subtasks: aspect term extraction, opinion term extraction, and sentiment classification. Existing work can be mainly divided into two types: pipeline-based methods and joint extraction methods. Pipeline-based methods solve subtasks individually and ignore the interdependencies between them. For example, Peng et al. [16] first proposed ASTE task. They solved three subtasks separately, and then paired them for forming the triplets. Another approach to addressing ASTE task is through joint extraction, which trains the model in an end-to-end manner [23, 21, 27, 31]. In this approach, the model is trained to simultaneously perform aspect term extraction, opinion term extraction, and sentiment classification. By jointly optimizing the model for all subtasks, it can capture the correlations and interactions between these subtasks, leading to more coherent and accurate results. For example, Xu et al. [23] regarded ASTE as a sequence tagging task which can capture relations among sentiment triplets. Chen et al. [3] designed a span-level bidirectional to extract triplets in both aspect-to-opinion and opinion-to-aspect directions. In recent years, some researchers studied to utilized machine reading comprehension framework [2, 11, 26] to jointly extract sentiment triplets. These methods manually constructed different queries to obtain sentiment triplets simultaneously.

2.2 Machine Reading Comprehension

Given a query, the machine reading comprehension framework aims to generate the corresponding answer. By using MRC framework, the model can learn the interaction between the query and context comprehensively. Recently, there have been many methods in the field of natural language processing (NLP) that utilized machine reading comprehension to solve various tasks [10], including ASTE task. Machine reading comprehension framework typically requires manually constructed queries, which provide a 0 convenient way for researchers to capture implicit relationships between subtasks. Chen et al. [2] first proposed a bidirectional machine reading comprehension (BMRC) framework to solve ASTE. By devising multi-turn queries, they effectively built the associations among subtasks. Based on MRC framework, Zhai et al. [26] proposed a context augmentation strategy and a discriminative model to address the issue of interference between multiple aspect terms.

Despite the success of MRC framework in ASTE task, the BMRC-based methods ignore task-specific features, leading to the problem of task confusion in the model. How to efficiently inject task-specific features to address subtask confusion in the BMRC framework is a challenging problem.

2.3 Knowledge Distillation

Knowledge distillation [17, 5, 32] mitigates the performance degradation resulting from model compression. Specifically, knowledge distillation is a recently emerging approach that seeks to obtain a small student model by distilling knowledge from a larger teacher model, while still achieving comparable performance. For example, Sanh et al. [18] utilize soft target to train the student model. Jiao et al. [8] and Sun et al. [19] aligned the hidden representation between teacher and student models. In this paper, we use layer-wise distillation to align the hidden layer outputs of adapters and pre-trained models, thereby addressing the semantic gap between them.

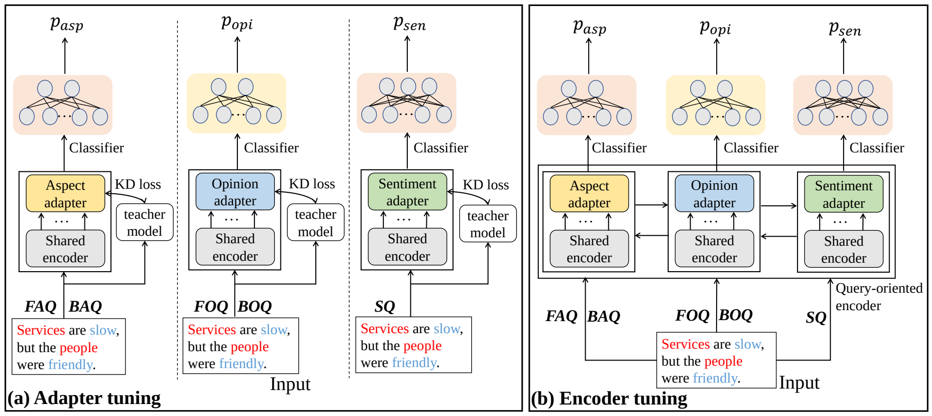

Fig. 2: The overview of our proposed QoMRC framework. (a) The query-oriented adapter tuning stage: we freeze the shared encoder and train three query-oriented adapters based on different subtasks, respectively. (b) The query-oriented encoder tuning: we freeze trained adapters and fuse the representation of the adapter and the shared encoder based on the category of query to conduct joint training.

3 QoMRC Framework

In this section, we first present the problem statement of aspect sentiment triplet extraction (ASTE), and then give an overview of our proposed QoMRC framework. After that, we explain the technical details of QoMRC framework.

3.1 Problem Statement

Given a sentence S=$\{w_1, w_2, ..., w_l\}$ consisting of l tokens, the aim of ASTE task is to extract a set of sentiment triplets $\mathit{\Gamma}$ = $\{(a_i, o_i, s_i)\}_{i=1}^{|T|}$, where a_i, o_i, s_i, and $|T|$ denote AT, OT, SP, and the number of sentiment triplets.

The BMRC-based methods typically require manual query design and adopt a multi-turn interaction with the model. Specifially, they formalized ASTE task as a three-turn MRC task. In the first turn, they constructed non-restrictive queries to extract aspect terms or opinion terms. In the second turn, given the aspect terms or opinion terms extracted in the first turn, they constructed restrictive queries to extract the corresponding opinion terms or the corresponding aspect terms. In the final turn, given the aspect-opinion pairs, they constructed sentiment classification query to predict the sentiment polarity for each pair.

3.2 An Overview of QoMRC

In this paper, we propose the QoMRC framework, which is shown in Figure 2. Our framework is based on BMRC [2] but incorporates significant improvements. Specifically, QoMRC is a two-stage method. In the first stage, we train three query-oriented

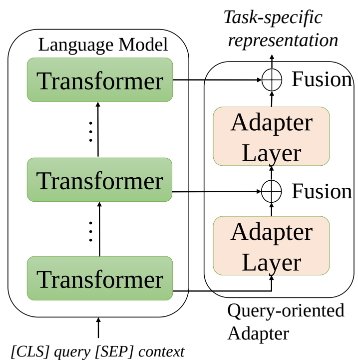

Fig. 3: The structure of query-oriented adapter.

adapters which can capture task-specific features by utilizing predefined queries. Otherwise, we employ a trained teacher model to guide adapters in generating hidden representations. In the second stage, we freeze trained adapters and fuse their representations and the representations from the shared encoder to obtain task-specific representations while also preserving task-shared features.

3.3 Predefined Query

To guide the model in capturing task-specific features, we divide queries in the BMRC framework into five fine-grained categories: forward aspect query (**FAQ**) (i.e., *what aspect?*) aims to query all aspects from the context; backward opinion query (**BOQ**) (i.e., *what opinion?*) aims to query all opinions from the contexts; forward opinion query (**FOQ**) (e.g., *what opinion given the aspect service?*) aims to query opinion based on predicted aspect; backward aspect query (**BAQ**) (e.g., *what aspect given the opinion slow?*) aims to query aspect based on predicted opinion; sentiment query (**SQ**) (e.g., *what sentiment given the aspect service and the opinion slow?*) aims to query sentimental polarity based on predicted query-opinion pairs. Note that **FAQ** and **BOQ** are non-restrictive queries, **FOQ**, **BAQ** and **SQ** are restrictive queries. Specifically, for each sentence, we have a **FAQ** and a **BOQ**. For each sentiment triplet instead of a sentence, we have a **FOQ**, a **BAQ** and a **SQ**.

3.4 Query-oriented Adapter

In the query-oriented adapter tuning stage, we train three query-oriented adapters: aspect adapter, opinion adapter and sentiment adapter. These adapters retain task-specific features which can be utilized to generate task-specific representation. For more convenient use, we plug adapter layers among different transformer layers of the pre-trained model instead of changing the internal structure of the pre-trained model, as shown in Figure 3. Following the structure of the adapter proposed by [20], we devise each query-oriented adapter to consist of three adapter layers that contain two transformer layers and two projection layers.

Specifically, for aspect adapter tuning, we need to identify all aspects from the context (i.e., aspect term extract (ATE)). Besides, to fully capture task-specific features and enhance the interaction between the aspect term and opinion term, we ask the model not only FAQ (e.g., *what aspect?*) but also BAQ (e.g., *what aspect given the opinion slow?*) for aspect adapter tuning. Although FAQ and BAQ perform the same task (i.e., ATE), the constraints of BAQ are stronger compared to FAQ, further enhancing the model's ability to capture task-specific features. For a given sentence, the input of aspect adapter training is described as follows:

$$\begin{aligned} In_{asp} &= [CLS]FAQ[SEP]sentence, \\ In_{asp} &= [CLS]BAQ_x[SEP]sentence, \end{aligned} \tag{1}$$

where FAQ is the query *"what aspect?"*, BAQ$_x$ is the query *"what aspect given the opinion$_x$"*, x represents the x-th triplet in the sentence. Then the constructed sentences In_{asp} are input into the frozen pre-trained model (i.e., BERT):

$$H_{asp} = BERT(In_{asp}) \in \mathbb{R}^{N \times (l+i+2) \times d}, \tag{2}$$

where N denotes that the pre-trained model consists of N layers of transformer, and each transformer could generate a hidden embedding of dimension $(l + i + 2) \times d$. i and 2 denote the length of query and identifier of BERT (i.e., [CLS] and [SEQ])[9], respectively. d is the embedding dimension.

Then we fuse the output of a certain layer's transformer in pre-trained with the output of the adapter layer and input it into the next adapter layer, as shown in Figure 3:

$$\begin{aligned} U_{asp}^{n+1} &= [u_1^{n+1}, ..., u_{l+i+2}^{n+1}] \\ &= [h_1^n + u_1^n, ..., h_{l+i+2}^n + u_{l+i+2}^n], \end{aligned} \tag{3}$$

where h^n represents the n-th layer of the transformer block, and u^n represents the corresponding adapter layer for that transformer block. we use a simple and efficient point-wise addition operation to fuse the representation of the pre-trained model and the adapter.

Due to the lack of training for the adapter, directly inserting the adapter into the pre-trained model for training would result in a significant semantic gap. To address this issue, we use a layer-wise knowledge distillation algorithm. Specifically, we use BERT-large as the teacher to distill the transformer layer output. We use mean squared error (MSE) as the distillation loss:

$$\mathcal{L}_{kd}^1 = \sum_{i=0}^{N'} MSE(U_{asp}^i, H_{tea}^i), \tag{4}$$

where N' is total number of layers that need to be distilled. H_{tea} is the output of specific transformer layer in the teacher model.

Subsequently, we obtain the output representation from the last layer of the adapter and treat ATE as binary classification tasks. We propose two binary classifiers to predict the start position and the end position of the answer, respectively.

$$\begin{aligned} p(y_a^s) &= softmax(U_{asp} W_s), \\ p(y_a^e) &= softmax(U_{asp} W_e), \end{aligned} \tag{5}$$

where $W_s \in \mathbb{R}^{d\times 2}$ and $W_s \in \mathbb{R}^{d\times 2}$ are learnable parameters. The loss function of ATE is defined as cross-entropy:

$$\mathcal{L}^1_{asp} = -\sum_{a=1}^{L+i+2} [y^s_a log(p^s_a) + y^e_a log(p^e_a)], \tag{6}$$

where p_a is the predicted score, y_a is the ground truth. Given the cross-entropy loss and knowledge distillation loss, our loss for aspect adapter training is as follows:

$$\mathcal{L}^1_{all} = \mathcal{L}^1_{asp} + \mathcal{L}^1_{kd}, \tag{7}$$

For the opinion adapter, the input can be described as follows:

$$\begin{aligned} In_{opi} &= [CLS]BOQ[SEP]sentence, \\ In_{opi} &= [CLS]FOQ_x[SEP]sentence, \end{aligned} \tag{8}$$

where BOQ is the query *"what opinion?"*, FOQ is the query *"what opinion given the aspect$_x$"*. Subsequently, we train the opinion adapter using a similar method as mentioned above.

For the sentiment adapter, the input can be described as follows:

$$In_{sen} = [CLS]SQ_x[SEP]sentence, \tag{9}$$

where SQ is the query *"what sentiment given the aspect$_x$ and the opinion$_x$?"*. For the (n+1)-th sentiment adapter layer's input, we only need to combine the n-th adapter layer's *[CLS]* vector with the correponding transformer layer's *[CLS]* vector:

$$\begin{aligned} H_{sen} &= BERT_{[CLS]}(In_{sen}) \in \mathbb{R}^{N\times(1)\times d}, \\ U^{n+1}_{sen} &= U^n_{[CLS]} + H^n_{sen}. \end{aligned} \tag{10}$$

Similar to aspect adapter, we use a layer-wise knowledge distillation algorithm to distill the transformer layer output. Then we use a three-class classfier to predict sentiment polarity:

$$p(y^{sen}_a) = softmax(U_{sen}W_{sen}), \tag{11}$$

where $W_{sen} \in R^{d\times 3}$ is learnable parameters. The loss function of SC is defined as cross-entropy, as shown in equation (6). The loss for sentiment adapter training consists of knowledge distillation loss and SC loss.

3.5 Query-oriented Encoder

In this stage, we freeze the trained adapter in the first stage and train the shared encoder to obtain query-oriented encoder. Given a sentence and its corresponding query, we obtain its task-specific representation by fusing the representation of the last transformer layer of the shared encoder (i.e., BERT) and the representation of the last adapter layer of the query-oriented adapter. Specifically, we regard FAQ and BAQ as ATE task and fuse the representation of the shared encoder and aspect adapter as follows:

$$\begin{aligned} [w_1, ..., w_L] &= transformer_M(In_{asp}), \\ [u_1, ..., u_L] &= asp\_adapter_N(In_{asp}), \\ [h_1, ..., h_L] &= [w_1 + u_1, ..., w_L + u_L], \end{aligned} \tag{12}$$

where *M* and *N* denote the last layer of BERT and the adapter, L is the length of In_{asp}, symbol $+$ denotes the point-wise addition. Through the joint generation of the shared encoder and the query-oriented adapter, $[h_1, ..., h_L]$ is the task-specific representation that retains the feature of the ATE task. Subsequently, we adopt equation (5) to predict the span of the aspect term and calculate the ATE loss $\mathcal{L}^2_{asp}$ as equation (6).

Similarly, we fuse the representation of the shared encoder and opinion adapter for FOQ and BOQ and obtain the OTE loss $\mathcal{L}^2_{opi}$. We fuse the representation of the shared encoder and sentiment adapter for SQ obtain the OTE loss $\mathcal{L}^2_{sen}$. The final loss is the combination of the above loss:

$$\mathcal{L} = \mathcal{L}^2_{asp} + \mathcal{L}^2_{opi} + \mathcal{L}^2_{sen}. \tag{13}$$

It is worth noting that in the first stage, the three subtasks are trained independently without interference, while in the second stage, the three subtasks are jointly trained.

3.6 Inference

In the same manner as Chen et al. [2], during inference stage, we use bidirectional MRC to obtain sentiment triplet. Specifically, in the forward direction, we first ask the model FAQ and obtain the aspect terms. Subsequently, based on the aspect terms provided by the model, we ask FOQ and obtain the opinion terms. In the backward direction, we first ask the model BOQ and obtain the opinion terms. Subsequently, based on the opinion terms provided by the model, we ask BAQ and obtain the aspect terms. Each aspect-opinion pair is valid only if its probability is higher than the given threshold. Finally, we ask the model SQ and obtain the sentiment polarity about aspect-opinion pair.

4 Experiment

4.1 Experimental Datasets and Setup

Datasets. For the reliability and authority of experimental results, we conduct experiments on four popular benchmark datasets that were created from the SemEval Challenges for ASTE task [23]. Three datasets are in restaurant domain and one dataset is in laptop domain. The statistics of these datasets are shown in Table 1.

Experimental Setup. In this paper, our goal is to demonstrate the necessity of injecting task-specific features into the BMRC framework. We use *bert-large-uncased* as our shared encoder and the teacher model in all our experiments. We train the teacher model following the method proposed by Chen et al. [2]. Each task-specific adapter contains three adapter layers. The structure of adapter layers is designed following k-adapter

Table 1: Statistic of ASTE-Data-v2[23]. #s, #t, #a, and #o represent the number of sentences, triplets, aspect terms, and opinion terms

Datasets		#s	#t	#a	#o
res14	Train	1266	2338	2051	2061
	Dev	310	577	500	497
	Test	492	994	848	844
lap14	Train	906	1460	1280	1254
	Dev	219	346	295	302
	Test	328	543	463	466
res15	Train	605	1013	862	935
	Dev	148	249	213	236
	Test	322	485	432	460
res16	Train	857	1394	1198	1300
	Dev	210	399	296	319
	Test	326	514	452	474

[20]. The bert-large layers where adapter layers plug in are $\{0, 11, 23\}$. The model size of the task-specific adapter is much smaller than the pre-trained model, which makes the task-specific feature capture process more efficient. It is worth noting that the shared encoder is fixed in the adapter tuning step and the adapter is fixed in the encoder tuning step. We use AdamW [14] for optimization with weight decay 0.01 and warmup rate 0.1. The learning rate is set 1e-3 for classifier and 1e-5 for bert-large respectively. All experiments are performed on a single NVIDIA GTX 3090 with 24G GPU memory.

4.2 Baselines and Evaluations

We compare QoMRC with the existing state-of-the-art methods. Peng-two-stage [16], Unified [24], SPAN-ASTE [22], EMC-GCN [1], BMRC [2], BDTF [31], COM-MRC [26], RBMRC [11], MvP [4]:

- **Peng-two-stage** [16] is a two-stage pipeline method. In the first stage, they extract both aspect-sentiment pairs and opinion. In the second stage, they pair up the extraction results into triplets.
- **Unified** [24] is a generative framework. They exploit the pre-training sequence-to-sequence model BART to solve all ABSA subtasks.
- **SPAN-ASTE** [22] is a span-level prediction method. They model the interaction between the span of aspect terms and opinion terms when predicting their sentimental polarity.
- **EMC-GCN** [1] utilizes a biaffine attention module to model the relation probability distribution between words in a sentence and transforms the sentence to a multi-channel graph.
- **BMRC** [2] formalizes the ASTE task as a bidirectional machine reading comprehension task.
- **BDTF** [31] represents each triplet as a relation region in the 2D table and transforms the ASTE task into detection and classification of relation regions.

Table 2: Main results (F1-score) on the ASTE-Data-v2 datasets [23]. The symbol * denotes that the corresponding results are retrieved from [26]. For fair comparison, we implement other baselines with *bert-large*.

Model	**res14**	**lap14**	**res15**	**res16**
Peng-two-stage*	51.46	42.87	52.32	54.21
Unified*	65.25	58.68	59.26	67.62
SPAN-ASTE*	71.85	59.38	63.27	70.26
EMC-GCN	71.10	58.23	59.02	68.09
BMRC	71.55	57.11	58.88	66.52
BDTF	74.09	**64.23**	65.03	72.18
COM-MRC	69.70	59.64	65.39	70.08
RBMRC	74.55	62.80	63.86	72.30
MvP	74.30	63.33	65.89	73.48
QoMRC	**75.83**	63.88	**66.53**	**73.53**

Table 3: Results (Precision and Recall) on the ASTE-Data-v2 datasets [23].

Model	**res14**		**lap14**		**res15**		**res16**	
	P	**R**	**P**	**R**	**P**	**R**	**P**	**R**
Peng-two-stage*	43.24	63.66	37.38	50.38	48.07	57.51	46.96	64.24
Unified*	65.52	64.99	61.41	56.19	59.14	59.38	66.60	68.68
SPAN-ASTE*	72.89	70.89	63.44	55.84	62.18	64.45	69.45	71.17
EMC-GCN	69.76	72.49	64.30	53.21	62.79	55.67	63.47	73.42
BMRC	71.18	71.93	61.02	53.67	60.23	57.58	67.25	65.81
BDTF	75.21	72.94	**69.00**	**60.07**	**67.56**	62.68	**71.43**	72.96
COM-MRC	68.51	70.93	61.33	58.04	67.40	63.51	67.26	73.15
RBMRC	75.59	73.54	66.06	59.85	63.60	64.12	70.56	74.12
QoMRC	**75.91**	**75.75**	68.90	59.54	65.09	**68.04**	70.81	**76.46**

- **COM-MRC** [26] consists of three closely-related components: a context augmentation strategy, a discriminative model and an inference method to address the issue of interference between multiple aspect terms.
- **RBMRC** [11] optimize BMRC by incorporating four improvements: word segmentation, span matching, probability generation, and exclusive classifiers.
- **MvP** [4] introduces element order prompts to guide the language model to generate multiple sentiment tuples, each with a different element order, and then selects the most reasonable tuples by voting.

We measure the experimental results with standard evaluation metrics, including Precision, Recall and F1 [16, 12]. The criterion for measuring the correctness of sentiment triplet predictions is that aspect term, opinion term, and sentiment polarity are all predicted correctly.

Table 4: The F1-score of ablation study on four datasets.

Model	res14	lap14	res15	res16
QoMRC	**75.83**	**63.88**	**66.53**	73.53
-w/o KD	75.52	62.56	66.20	72.97
-w/o aspect adapter	75.06	62.65	64.06	72.46
-w/o opinion adapter	74.94	61.86	64.02	**73.61**
-w/o sentiment adapter	75.00	62.00	64.85	72.90

4.3 Results

The main results (F1-score) are shown in Table 2. Our proposed QoMRC framework outperforms all baselines in metric F1 except for BDTF on the lap14 dataset. Specifically, on the res14, lap14, res15, and res16 datasets, the F1 scores of our method obtain gains of 1.33, 1.08, 2.67, and 1.23, compared with another strong BMRC-base method (i.e., RBMRC). This improvement indicates that our proposed QoMRC framework can efficiently inject task-specific features into the MRC framework to alleviate the model confusion about subtasks. Our method achieves an improvement of 1.74, 1.50 and 1.35 points over BDTF on the res14, res15 and res16 datasets. However, on the lap14 datasets, our method shows a decrease of 0.35 points. The reason for this is that BDFT is a table-filling approach, which has stronger constraints on the pairing of aspect terms and opinion terms. On the other hand, our method does not focus on how to correctly pair aspect terms and opinion terms, but rather injects task-specific features, which causing our model have lower F1 on lap14. To further analyze the results, we presented the precision and recall in Table 3. From Table 3, it can be observed that BDFT outperforms our method in some metrics. Indeed, the performance further supports the analysis that BDFT has a stronger ability to pair aspect terms and opinion terms. Despite that, our method still outperforms all existing state-of-the-art models in terms of overall performance. The experimental results further demonstrate the importance of capturing task-specific features.

4.4 Ablation Study

In this subsection, we conduct an ablation study on four datasets to further demonstrate the effectiveness of different modules of QoMRC. The results are shown in F1-score in Table 4. We have four ablation settings, which are: removing the knowledge distillation module (w/o KD), removing the aspect adapter module (w/o aspect adapter), removing the opinion adapter module (w/o opinion adapter), and removing the sentiment adapter module (w/o sentiment adapter). Specifically, removing the KD module indicates the adapters are trained without the guidance of a trained teacher model. Removing the aspect adapter, opinion adapter and opinion adapter indicates not injecting the features of the aspect term extraction (ATE) task, opinion term extraction (OTE) task, and sentiment classification (SC) task into the model, respectively. Based on the results in Table 4, it is evident that all the ablation variants exhibit significant decreases. The result

Table 5: The case study conducted on the res14 dataset. We compare our QoMRC framework with RBMRC.

Sentence	RBMRC	QoMRC
I trust the people at Go Sushi, it never disappoints	{people, never disappoints, pos}✗	{people, trust, pos}✔
Try the ribs, sizzling beef and couple it with coconut rice	{ribs, try, pos}✔ {sizzling, try, pos}✗	{ribs, try, pos}✔ {beef, try, pos}✔

Table 6: The subtasks results (F1-score) on the res14 dataset.

Model	**ATE**	**OTE**	**ASPE**	**AOPE**
RBMRC	85.65	88.05	80.19	78.94
QoMRC	**86.92**	**88.25**	81.36	**80.36**
-w/o KD	85.99	88.20	81.20	79.28
-w/o aspect adapter	85.12	88.08	81.12	79.80
-w/o opinion adapter	86.58	87.88	79.58	79.19
-w/o sentiment adapter	86.36	88.00	**81.42**	79.51

strongly supports the validity and non-redundancy of our QoMRC framework. The performances of the four ablation variants will drop by an average of 0.63, 1.39, 1.34 and 1.26 points, respectively.

4.5 Case Study

In order to provide a more comprehensive understanding of the impact of capturing task-specific features for mitigating model confusion subtasks, we conduct case study on the res14 dataset, and compare the results with RBMRC. As shown in Table 5, RBRMC misidentify opinion *trust* as *never disappointment* in the first example, and aspect *beef* as *sizziling* in the second example. With the help of task-specific features, QoMRC can make correct inference. The case study further verify the validity of our QoMRC method.

4.6 Subtasks Experiment

To further demonstrate the effectiveness of injecting task-specific features into MRC framework in addressing model subtask confusion in MRC framework, we conducted experiments involving four subtasks. The results on the res14 and res15 are shown in F1-score in Table 6 and Table 7. ATE, OTE, ASPE and AOPE denote aspect term extraction, opinion term extraction, aspect-sentiment pair extraction and aspect-opinion pair extraction. We also conducted experiments using RBMRC establish a comparison. It can be observed that our QoMRC framework outperforms the strong baseline RBMRC in all subtasks. Due to the removal of task-specific features, the results for four subtasks show a significant decrease. The results provide strong evidence for the importance of task-specific features in the MRC framework.

Table 7: The subtasks results (F1-score) on the res15 dataset.

Model	ATE	OTE	ASPE	AOPE
RBMRC	81.20	80.13	72.79	71.50
QoMRC	**81.95**	**82.37**	**73.88**	**73.46**
-w/o KD	81.68	81.21	72.24	72.97
-w/o aspect adapter	79.13	79.56	72.51	71.33
-w/o opinion adapter	81.10	78.45	71.05	70.28
-w/o sentiment adapter	81.01	80.13	73.27	71.64

Table 8: Comparison of our model with gpt-3.5-turbo.

Model	res14	lap14	res15	res16
gpt-3.5-turbo (zero-shot)	60.42	42.19	40.20	45.97
gpt-3.5-turbo (few-shot)	62.36	44.65	41.25	50.47
QoMRC	**75.83**	**63.88**	**66.53**	**73.53**

4.7 Comparison with ChatGPT

As a new emerging technology, large language models have shown excellent performance in natural language tasks, especially in generation tasks. To provide a more comprehensive evaluation of our model, we have decided to compare it with ChatGPT (gpt-3.5-turbo). Due to budget constraints, we tested gpt-3.5-turbo with 100 random samples for each dataset. To provide a fairer comparison, we have two experimental settings: zero-shot and few-shot. For the zero-shot setting, we directly concatenate the prompt and context as input to ChatGPT to obtain the results, such as *give the sentiment triplet in the following sentence: Services are slow, but the people were friendly.*. For the few-shot setting, we include one examples in the prompt. The experimental results are shown in Table 8. Compared to zero-shot and few-shot settings of gpt-3.5-turbo, our method shows significant improvements.

5 Conclusion

In this paper, we propose a novel Query-oriented Machine Reading Comprehension (QoMRC) for ASTE task which can be regarded as a two-stage method. Specifically, in the first stage, we train three query-oriented adapters which can capture task-specific features by utilizing predefined queries. In addition, we employed a technique called layer-wise task-specific knowledge distillation to address a significant semantic gap between pre-trained models(e.g., BERT) and initialized adapters. In the second stage, we freeze trained adapters and fuse its representation and the representation from the shared encoder to obtain task-specific representation while also preserving task-shared features. Extensive experiment results on benchmark datasets show the efficacy of our proposed method, and indicate the necessity of capturing task-specific features.

Acknowledgements. This research was supported by grants from the National Natural Science Foundation of China (Grants No. 62337001), the Anhui Provincial Natural Science Foundation of China (No. 2308085QF229) and the Fundamental Research Funds for the Central Universities. This work was also supported by Ant Group through CCF-Ant Research Fund.

References

1. Chen, H., Zhai, Z., Feng, F., Li, R., Wang, X.: 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 (2022)
2. Chen, S., Wang, Y., Liu, J., Wang, Y.: Bidirectional machine reading comprehension for aspect sentiment triplet extraction. In: Proceedings of the AAAI conference on artificial intelligence. vol. 35, pp. 12666–12674 (2021)
3. Chen, Y., Keming, C., Sun, X., Zhang, Z.: A span-level bidirectional network for aspect sentiment triplet extraction. In: Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing. pp. 4300–4309 (2022)
4. Gou, Z., Guo, Q., Yang, Y.: Mvp: Multi-view prompting improves aspect sentiment tuple prediction. arXiv preprint arXiv:2305.12627 (2023)
5. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531 (2015)
6. Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., De Laroussilhe, Q., Gesmundo, A., Attariyan, M., Gelly, S.: Parameter-efficient transfer learning for nlp. In: International Conference on Machine Learning. pp. 2790–2799. PMLR (2019)
7. Hu, M., Liu, B.: Mining and summarizing customer reviews. In: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 168–177 (2004)
8. Jiao, X., Yin, Y., Shang, L., Jiang, X., Chen, X., Li, L., Wang, F., Liu, Q.: Tinybert: Distilling bert for natural language understanding
9. Kenton, J.D.M.W.C., Toutanova, L.K.: Bert: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of NAACL-HLT. pp. 4171–4186 (2019)
10. Li, X., Yin, F., Sun, Z., Li, X., Yuan, A., Chai, D., Zhou, M., Li, J.: Entity-relation extraction as multi-turn question answering. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. pp. 1340–1350 (2019)
11. Liu, S., Li, K., Li, Z.: A robustly optimized bmrc for aspect sentiment triplet extraction. In: Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 272–278 (2022)
12. Liu, Y., Wu, H., Huang, Z., Wang, H., Ning, Y., Ma, J., Liu, Q., Chen, E.: Techpat: technical phrase extraction for patent mining. ACM Transactions on Knowledge Discovery from Data **17**(9), 1–31 (2023)
13. Liu, Y., Zhang, K., Huang, Z., Wang, K., Zhang, Y., Liu, Q., Chen, E.: Enhancing hierarchical text classification through knowledge graph integration. In: Findings of the Association for Computational Linguistics: ACL 2023. pp. 5797–5810 (2023)
14. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101 (2017)
15. Mao, Y., Shen, Y., Yu, C., Cai, L.: A joint training dual-mrc framework for aspect based sentiment analysis. In: Proceedings of the AAAI conference on artificial intelligence. vol. 35, pp. 13543–13551 (2021)

16. Peng, H., Xu, L., Bing, L., Huang, F., Lu, W., Si, L.: Knowing what, how and why: A near complete solution for aspect-based sentiment analysis. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 34, pp. 8600–8607 (2020)
17. Romero, A., Ballas, N., Kahou, S.E., Chassang, A., Gatta, C., Bengio, Y.: Fitnets: Hints for thin deep nets. arXiv preprint arXiv:1412.6550 (2014)
18. Sanh, V., Debut, L., Chaumond, J., Wolf, T.: Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. arXiv preprint arXiv:1910.01108 (2019)
19. Sun, S., Cheng, Y., Gan, Z., Liu, J.: Patient knowledge distillation for bert model compression. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). pp. 4323–4332 (2019)
20. Wang, R., Tang, D., Duan, N., Wei, Z., Huang, X.J., Ji, J., Cao, G., Jiang, D., Zhou, M.: K-adapter: Infusing knowledge into pre-trained models with adapters. In: Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021. pp. 1405–1418 (2021)
21. Wu, Z., Ying, C., Zhao, F., Fan, Z., Dai, X., Xia, R.: Grid tagging scheme for aspect-oriented fine-grained opinion extraction. In: Findings of the Association for Computational Linguistics: EMNLP 2020. pp. 2576–2585 (2020)
22. Xu, L., Chia, Y.K., Bing, L.: Learning span-level interactions for aspect sentiment triplet extraction. 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. 4755–4766 (2021)
23. Xu, L., Li, H., Lu, W., Bing, L.: Position-aware tagging for aspect sentiment triplet extraction. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 2339–2349 (2020)
24. Yan, H., Dai, J., Ji, T., Qiu, X., Zhang, Z.: A unified generative framework 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 1: Long Papers). pp. 2416–2429 (2021)
25. Yang, B., Cardie, C.: Extracting opinion expressions with semi-markov conditional random fields. In: Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning. pp. 1335–1345 (2012)
26. Zhai, Z., Chen, H., Feng, F., Li, R., Wang, X.: Com-mrc: A context-masked machine reading comprehension framework for aspect sentiment triplet extraction. In: Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing. pp. 3230–3241 (2022)
27. Zhang, C., Li, Q., Song, D., Wang, B.: A multi-task learning framework for opinion triplet extraction. arXiv preprint arXiv:2010.01512 (2020)
28. Zhang, K., Liu, Q., Qian, H., Xiang, B., Cui, Q., Zhou, J., Chen, E.: Eatn: An efficient adaptive transfer network for aspect-level sentiment analysis. IEEE Transactions on Knowledge and Data Engineering **35**(1), 377–389 (2021)
29. Zhang, K., Zhang, H., Liu, Q., Zhao, H., Zhu, H., Chen, E.: Interactive attention transfer network for cross-domain sentiment classification. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 33, pp. 5773–5780 (2019)
30. Zhang, K., Zhang, K., Zhang, M., Zhao, H., Liu, Q., Wu, W., Chen, E.: Incorporating dynamic semantics into pre-trained language model for aspect-based sentiment analysis. arXiv preprint arXiv:2203.16369 (2022)
31. Zhang, Y., Yang, Y., Li, Y., Liang, B., Chen, S., Dang, Y., Yang, M., Xu, R.: Boundary-driven table-filling for aspect sentiment triplet extraction. In: Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing. pp. 6485–6498 (2022)
32. Zuo, S., Zhang, Q., Liang, C., He, P., Zhao, T., Chen, W.: Moebert: from bert to mixture-of-experts via importance-guided adaptation. arXiv preprint arXiv:2204.07675 (2022)