Instruction Tuning with Retrieval-based Examples Ranking for Aspect-based Sentiment Analysis

Guangmin Zheng[1], **Jin Wang**[*1], **Liang-Chih Yu**[*2] **and Xuejie Zhang**[1]
[1]School of Information Science and Engineering, Yunnan University, Yunnan, P.R. China
[2]Department of Information Management, Yuan Ze University, Taiwan
Contact: wangjin@ynu.edu.cn, lcyu@saturn.yzu.edu.tw

Abstract

Aspect-based sentiment analysis (ABSA) identifies sentiment information related to specific aspects and provides deeper market insights to businesses and organizations. With the emergence of large language models (LMs), recent studies have proposed using fixed examples for instruction tuning to reformulate ABSA as a generation task. However, the performance is sensitive to the selection of in-context examples; several retrieval methods are based on surface similarity and are independent of the LM generative objective. This study proposes an instruction learning method with retrieval-based example ranking for ABSA tasks. For each target sample, an LM was applied as a scorer to estimate the likelihood of the output given the input and a candidate example as the prompt, and training examples were labeled as positive or negative by ranking the scores. An alternating training schema is proposed to train both the retriever and LM. Instructional prompts can be constructed using high-quality examples. The LM is used for both scoring and inference, improving the generation efficiency without incurring additional computational costs or training difficulties. Extensive experiments on three ABSA subtasks verified the effectiveness of the proposed method, demonstrating its superiority over various strong baseline models. Code and data are released at https://github.com/zgMin/IT-RER-ABSA.

1 Introduction

Aspect-based sentiment analysis (ABSA) (Zhang and Liu, 2017) is a fine-grained text sentiment analysis technique that identifies sentiment information related to specific aspects, and provides deeper market insights for businesses and organizations. ABSA consists of three subtasks: aspect term extraction (ATE), aspect term sentiment classification (ATSC), and aspect sentiment pair extraction (ASPE). ATE identifies the aspects mentioned in the text, and ATSC determines the sentiment polarity associated with each aspect. Furthermore, ASPE jointly performs ATE and ASPE to extract sentiment tuples, including aspect terms and their associated sentiments.

For ABSA tasks, most previous methods have used transformer-based language models (LMs) in either a pipeline or end-to-end framework (Chen and Qian, 2020; Luo et al., 2020; Mao et al., 2021; Marcacini and Silva, 2021; Yang and Li, 2021). By adding task-specific layers to the top of the model, these models are typically initialized from the pretrained checkpoint and then fine-tuned on the downstream samples. Recently, generative models (Hosseini-Asl et al., 2022; Yan et al., 2021; Zhang et al., 2021) have emerged to reformulate ABSA tasks as generation tasks to produce a sequence with a special pattern, for example, `restaurant#positive` and `food#negative`. Based on the instruction-tuning paradigm (Mishra et al., 2022), several studies (Scaria et al., 2023; Varia et al., 2022) have further improved the generative approach using several predefined instruction prompts (Scaria et al., 2023). Instruction tuning allows generative models to tune themselves on a few input-output examples.

The quality of the output generated by the instruction-tuning model is highly dependent on the quality of in-context examples. Well-crafted instructions can help the model generate more accurate and relevant outputs(Luo et al., 2024), whereas poorly crafted instructions can lead to incoherent or irrelevant results. Nevertheless, previous studies typically adopted a fixed strategy to use two or more unchanged examples to generate the instruction template. If the examples are unrepresentative of the target task, the model may be unable to learn effectively.

For a target sample, for example, *The falafel was slightly **overcooked** and **dry**, but the chicken was **satisfactory***, the example *The price was too*

*Corresponding author.

high**, but the cab was **amazing can be appropriate. They share a similar syntactic structure, which can contribute to imitation and generation. However, such an example is unsuitable for another sample, for example, *The staff displays **arrogance**, and the prices are considerably **high** for Brooklyn standards*. Because the opinion of ***high*** in the example may finally impact the judgment of the aspect *price* of the target. Furthermore, the sample *We **enjoyed** our visit and utilized buses and cabs for transportation* seems to have little relevance to the example above. However, the aspect *cab* may be incorrectly considered an aspect term based on the prompt of the word *cab* in the example.

Empirical studies have demonstrated the use of an additional language model as a scorer to produce off-the-shelf sentence embeddings, for example, BM25 (Robertson and Zaragoza, 2009), EPR (Rubin et al., 2021) and LLM-R (Wang et al., 2023), for similarity calculations to retrieve examples from the training set. Several works explored training a prompt retriever to select examples by measuring surface similarity (Li et al., 2022; Liu et al., 2022; Zhang et al., 2022). These methods have two limitations: (i) the similarity calculation typically measures the distance in the latent space, which is independent of the generation target in instruction learning, and (ii) additional encoders are necessary to obtain the representation used for similarity calculations, which also incurs additional computational costs and training difficulties.

This study proposes an instruction-tuning method with retrieval-based example ranking for ABSA tasks, comprising a retriever and inference LM. The LM is a T5 model (Raffel et al., 2019) with an encoder-decoder structure and several instruction-tuned versions (Chung et al., 2022; Wang et al., 2022). The retriever returns the most appropriate examples to form the instruction template. Meanwhile, the LM scores the examples for the retriever and generates the final results.

To achieve the training of LM and retriever simultaneously, an alternating training schema is proposed. For each target sample, the candidates can be divided into positive and negative examples according to the log-likelihood provided by the LM. Then, contrastive learning is applied to force the sample to be near positive examples but distant from negative examples. After composing high-quality instructions, the LM can be finetuned using a generative objective.

Unlike the previously proposed similarity calculation, the retriever evaluates the importance of candidate examples based on the log-likelihood of the LM. This retrieval goal is consistent with the generative objective of the LM. In addition, the LM is used for both scoring and inference, improving the generation performance with tolerable additional computational costs or training difficulties.

Extensive experiments were conducted on ATE, ATSC, and ASPE tasks to verify the effectiveness of the proposed method. The results show that the proposed model substantially improves the performance compared with several strong baselines.

The remainder of this paper is organized as follows. Section 2 briefly reviews the related studies. Section 3 describes the proposed retrieval-based example mining method for instruction learning in ABSA. Section 4 summarizes the experiment settings and empirical results. Finally, Section 5 concludes the paper.

2 Related Work

2.1 Aspect-Based Sentiment Analysis

Aspect-based sentiment analysis (Zhang and Liu, 2017) aims to analyze sentiment polarities toward the specific aspects of a given text. In ABSA, text is decomposed into several aspects, and sentiment polarity, which is typically positive, neutral, or negative, is analyzed for each aspect. This approach provides fine-grained, in-depth sentiment insights, enabling businesses and organizations to better understand market and consumer perspectives.

Most approaches have focused on using encoder structures to accomplish aspect extraction and sentiment identification, such as improved text encoding using attention mechanisms (Marcacini and Silva, 2021; Yuan et al., 2020, 2022), multitask learning (Chen and Qian, 2020), and approaches based on machine reading comprehension (Mao et al., 2021). Some studies (Yan et al., 2021; Zhang et al., 2021) introduced decoders to unify ABSA's previous extraction and classification tasks into generative tasks. Recently, instruction prompts have been introduced into generative methods (Varia et al., 2022), achieve a solid few-shot performance by making the model perform the correct action with an apparent task description. Furthermore, fixed examples were added to the instructions to supplement the task description with more accurate information, yielding significant performance improvements in the ABSA subtasks (Scaria et al.,

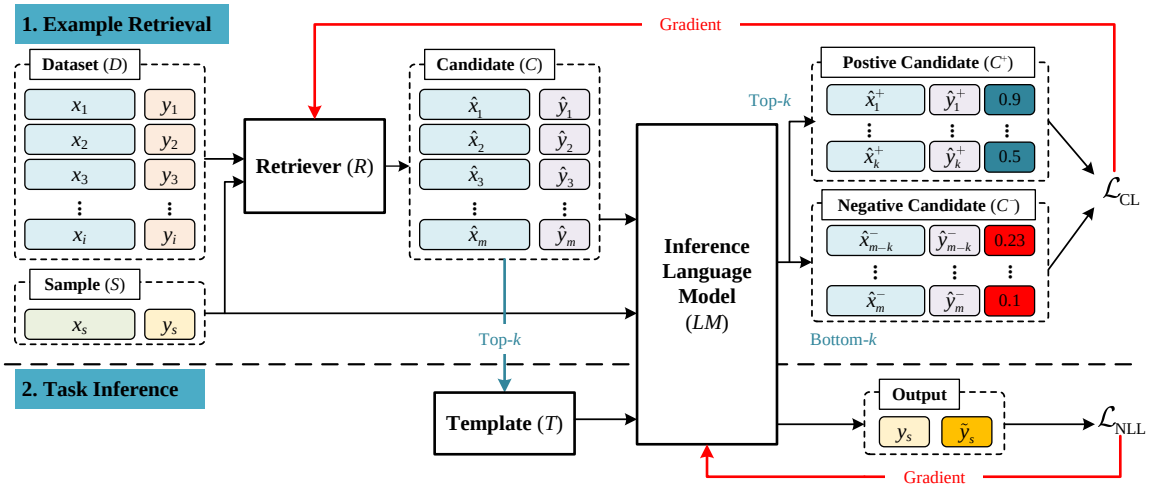


Figure 1: Overall architecture of the proposed instruction tuning with retrieval-based examples ranking for ABSA.

2023).

However, the adaptability of the fixed examples to different review texts is difficult to guarantee. Liu et al. (2022) demonstrate that selecting contextual examples significantly affects the downstream performance. The intricate relationship between selected examples and diverse review contexts underscores the need for a nuanced approach to ensure robust and effective adaptation in various scenarios.

2.2 Prompt Retrieval

With the development of deep learning techniques, dense retrieval (Karpukhin et al., 2020) has become a widely used information retrieval method that utilizes dense vectors to semantically match queries and documents in the latent space. Compared with sparse retrieval methods, dense retrieval exploits the powerful modeling capabilities of pretrained language models (PLMs) and may overcome the linguistic mismatch problem. Therefore, dense retrieval has become popular in current retrieval technology.

In context learning, retrieval enhancement aims to improve the performance of LMs in downstream tasks by retrieving information-rich examples (Li et al., 2023; Luo et al., 2023). In previous studies, unsupervised sentence encoders have often encoded training examples and retrieved their nearest neighbors for each test instance (Liu et al., 2022). In some studies (Das et al., 2021), a supervised prompt retriever was trained to answer questions on a knowledge base. This retriever relies on the surface similarity to perform queries when it receives supervised training tailored to knowledge-based queries. However, these methods tap only into text associations, ignoring how the language model understands these texts.

To address this problem, some researchers (Rubin et al., 2021; Wang et al., 2023) have proposed using the LM to score examples to train retrievers. Unfortunately, these methods are limited because they are only applicable to frozen large language models. This constraint underscores the need for more versatile methodologies that can extend beyond the confines of frozen LMs to enhance the flexibility and generalizability of retriever training strategies.

3 Retrieval-Based Instruction Tuning

Figure 1 shows the overall framework of the proposed instruction-tuning method with retrieval-based example ranking for ABSA. The training process comprises two phases: example retrieval and task inference. For example retrieval, a retriever is used to select several candidates for a given sample. The inference LM was then used to measure the likelihood of ranking the target and candidates as scores. To train the retriever, we selected the top-k candidates as positive samples and the others as negative samples. Subsequently, contrastive learning is performed to propagate the gradients and update the retriever. For task inference, the retriever returns the most suitable examples and forms input instructions according to a predefined template. The negative log-likelihood of the generative results will train the LM.

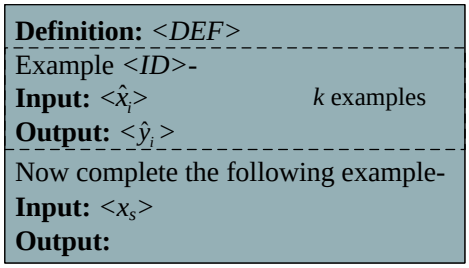


Figure 2: Template of instruction prompts. <DEF> is the definition of the task; <ID>, the identity of the examples; <$\hat{x}_i$> and <$\hat{y}_i$>, the input and output of the examples, respectively; k , the total number of examples; and <x_s>, the input text.

3.1 Instruction Template

Figure 2 shows the instruction prompt template, which comprises a task definition, examples, and input text. In particular, a task <DEF> is available for different subtasks. For a given input text <x_s>, examples <E> $= \{(\hat{x}_i, \hat{y}_i)\}_{i=1}^{k}$ are selected for instruction tuning, and the output conforms to the formats of the different subtasks. Thus, the instruction template for x_s can be formally denoted as $\text{Tmpl}(DEF, \{\hat{x}_i, \hat{y}_i\}_{i=1}^{k}, x_s)$.

Figure 3 shows the instruction prompts generated using the ATE, ATSC, and ASPE templates. For ATSC, the aspect term is spliced together with the review text as input through the fixed prompt *The aspect is*, as shown in the underlined portion of Figure 3.

3.2 Example Retrieval

Candidate Generation. Training set D consists of several input-output pairs, i.e. $D = \{(x_i, y_i)\}_{i=1}^{n}$, where x is the text, and y is the label. For a target sample (x_s, y_s), retriever R returns the top-m candidates from the training set,

$$C = R((x_s, y_s), D) = \{(\hat{x}_j, \hat{y}_j)\}_{j=1}^{m} \quad (1)$$

Sample (x_s, y_s) should be excluded from the retrieval results.

The LM adopted the encoder-decoder architecture of T5, which scores each candidate independently. The scoring function is the log-likelihood of output y_s, which is consistent with the autoregressive decoding objective of the LM, and is denoted as

$$\begin{aligned}\Delta_i &= \log p(y_s|DEF, \hat{x}_i, \hat{y}_i, x_s) \\ &= \sum_{l=1}^{L} \log p(y_s^l \mid DEF, \hat{x}_i, \hat{y}_i, x_s, y_s^{<l}) \end{aligned} \quad (2)$$

where $p(y_s|DEF, \hat{x}_i, \hat{y}_i, x_s)$ is the conditional probability of y given task definition <DEF>, input x_s, and i-th candidate $(\hat{x}_i, \hat{y}_i)$.

The candidate set is sorted in descending order according to the score Δ_i. The top-k candidates form the set of positive samples C^+, whereas the bottom-k candidates form the set of negative samples C^-. Contrastive learning is then applied to train the retriever, allowing the retrieved results to be scored as high as possible.

To reduce the computational cost of the scoring function, only a portion of the training set with ratio r is extracted to select the positive and negative examples for the training of the retriever.

Retriever Training. A contrastive learning objective trains the retriever. The input is (Input: <$\hat{x}_i$ > Output: <$\hat{y}_i$>) for the candidates $c_i = (\hat{x}_i, \hat{y}_i)$, and (Input: <x_s>) for the target sample. Besides, we followed Sentence-T5 (Ni et al., 2022) to average the encoder output as the representation for the candidates and target sample,

$$\mathbf{h}_i = \text{Mean}(\text{Enc}(\hat{x}_i, \hat{y}_i)) \quad (3)$$

$$\mathbf{h}_s = \text{Mean}(\text{Enc}(x_s)) \quad (4)$$

where $\mathbf{h}_i$ and $\mathbf{h}_s$ are the hidden representations of the i-th candidate and target sample, respectively. The distance for contrastive learning measures their inner products, and is denoted as

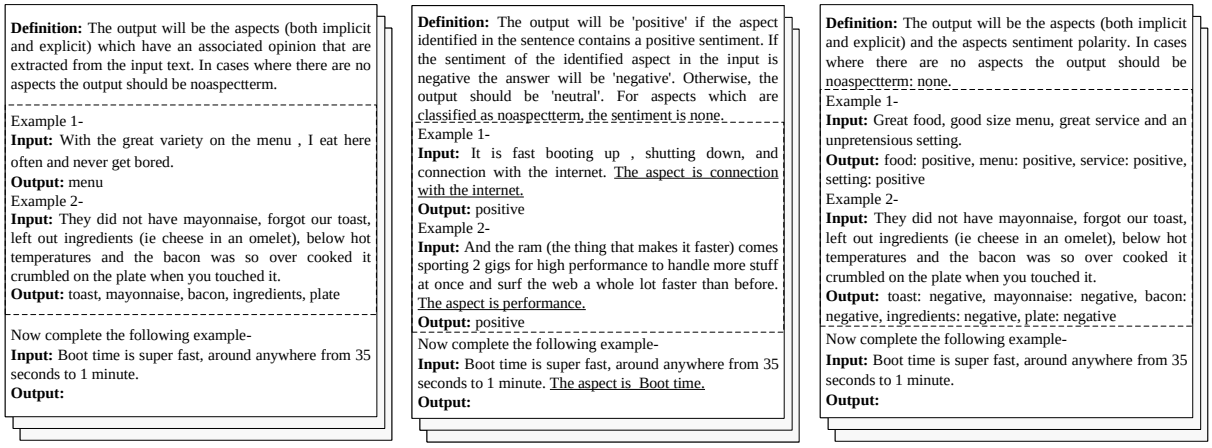
$$\text{Sim}(x_s, c_i) = \mathbf{h}_s^\top \mathbf{h}_i \quad (5)$$

For the target sample, one positive example c_i^+ is randomly selected from the set of positive samples C^+, one negative example from the set of negative samples C^-, and the other negative examples come from $B-1$ positive and $B-1$ negative examples sampled for the other samples in the same batch, where B is the batch size. The objective function minimizes the negative log-likelihood of the positive example for the target sample.

$$\begin{aligned}&\mathcal{L}_{\text{CL}}(DEF, x_s, c^+, c_1^-, \ldots e_{2B-1}^-) = \\ &\quad -\log \frac{e^{\text{Sim}(x_s, c^+)}}{e^{\text{Sim}(x_s, c^+)} + \sum_{j=1}^{2B-1} e^{\text{Sim}(x_s, e_j^-)}}\end{aligned} \quad (6)$$

3.3 Task Inference

We obtain the instruction template for inference by retrieving the top-k examples using the retriever. The LM predicts the probability of generating a



(a) ATE (b) ATSC (c) ASPE

Figure 3: Demonstrations of instruction prompts.

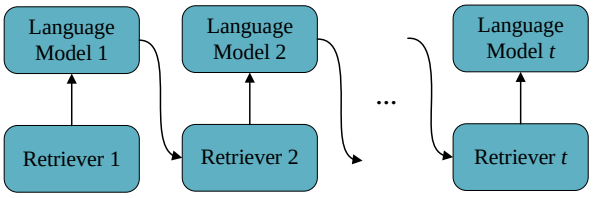


Figure 4: Alternating training schema.

target y_s, trained by minimizing the negative likelihood loss:

$$\mathcal{L}_{\text{NLL}} = -\sum_{l=1}^{L} \log p(y_s^l \mid DEF, \{\hat{x}_i, \hat{y}_i\}_{i=1}^k, x_s, y_s^{<l}) \quad (7)$$

where L is the output length of the target sample.

The LM receives input only in the form of instructions. Examples can be converted into instruction prompts using ATE, ATSC, and ASPE templates. To train the LM, only the top-1 scoring example is applied for fine-tuning,

$$\tilde{y}_s = \text{LM}(\text{Tmpl}(DEF, \hat{x}_1^+, \hat{y}_1^+, x_s)) \quad (8)$$

For inference, instruction prompts are constructed using the top-k examples. Upon receiving instructions, the LM produces an output specific to the current task and input x_s,

$$\tilde{y}_s = \text{LM}(\text{Tmpl}(DEF, \{\hat{x}_i, \hat{y}_i\}_{i=1}^k, x_s)) \quad (9)$$

3.4 Alternating Training Schema

The training of a retriever depends on the scoring of the LM. However, the LM training requires the retriever to form an instruction template. The instruction tuning performance depends on the collaborative effort between the two models.

Thus, we adopted an alternating training schema for the retriever and language models, as shown in Figure 4. In particular, the finetuned LM in the $t-1$ step is used as a scoring model to train the retriever in the t step. The LM in the t step is finetuned by the instruction generated by the updated retriever.

Dataset	Split	#Pos	#Neg	#Neu	#No	#T
Lap14	train	987	866	460	1557	3045
	test	341	128	169	378	800
Rest14	train	2164	805	633	1020	3041
	test	728	196	196	194	800
Rest15	train	912	256	36	482	1315
	test	326	182	34	284	685
Rest16	train	1240	439	69	766	2000
	test	468	117	30	256	676

Table 1: Statistics for experiment datasets. #Pos, #Neg, and #Neu denote the number of aspects with positive, negative, and neutral sentiments, respectively, #No denotes the number of aspect-free terms, and #T denotes the total number of samples.

4 Experiments

4.1 Datasets

Experiments were conducted on the Semeval-2014, 15, and 16 datasets (Pontiki et al., 2014, 2015, 2016), which are benchmark dataset for the ABSA task. The benchmark comprises four datasets, including customer reviews from two domains, laptops (**Lap14**) and restaurants (**Rest14**, **Rest15**, and **Rest16**). The model performance was measured

using F_1 scores for the ATE and ASPE tasks and accuracy for the ATSC task. Table 1 provides the details of the data distribution. Conflict labels were ignored.

4.2 Implementation Details

The LM was initialized using the **flan-t5-base**[1]. AdamW (Loshchilov and Hutter, 2019) was applied to optimize the model with an initial learning rate of 5e-5. The training batch size was 2. The gradient accumulation steps were set to 2. The number of epochs was 4 for the retriever, and 2 for the language model. The maximum sequence length was 128. Data ratio r of the training retriever was set to 0.1. The number of examples was one for the training phase and from zero to seven for the evaluation phase; in the comparative experiments, k was set to 4. The step t of alternating training schema was 3.

4.3 Baselines

The baseline models that emerged from the comparative experiments were categorized into generative and non-generative models,

- **1) Generative methods:** GPT2_{med} (Hosseini-Asl et al., 2022), BARTABSA (Yan et al., 2021), GAS (Zhang et al., 2021), IT-MTL (Varia et al., 2022), InstructABSA (Scaria et al., 2023);

- **2) Non-generative methods:** SPAN (Hu et al., 2019), GRACE (Luo et al., 2020), ABSA-DeBERTa (Marcacini and Silva, 2021), LSAT (Yang and Li, 2021), RACL-BERT (Chen and Qian, 2020), Dual-MRC (Mao et al., 2021), Seq2Path (Mao et al., 2022).

The number of paths k in Seq2Path is 4 and the results are from Mao et al. (2022). The results of other baselines are from Scaria et al. (2023). A more detailed descriptions about baselines are provided in Appendix A.

4.4 Comparative Results

Tables 2, 3, and 4 summarize the results of the proposed method relative to those of the baseline methods. The highest performance is in bold.

For the ATE task, the proposed method achieved F_1 scores of 79.93 and 83.85 on the **Rest15** and **Rest16** datasets, respectively, exceeding the baselines by 3.29% and 2.37%, respectively. In addition, it achieved better scores on the **Lap14** and **Rest14** datasets but was approximately 2% lower than InstructABSA, which used fixed examples. This might be related to the instruction template and the pre-trained language model. InstructABSA utilizes examples that distinguish between sentiment polarity and employs the pre-trained language model **tk-instruct-base-def-pos**[2], which could contribute to the performance variation. Nevertheless, the proposed model only performs worse than InstructABSA on the **Rest14** and **Lap14** datasets for the ATE task. We speculate two potential reasons for this:

1. The **Rest14** and **Lap14** datasets are larger thus have more retrievable examples compared to **Rest15** and **Rest16**, resulting in increased uncertainties in example variations. This might lead the model to rely on the knowledge provided by the examples overly.

2. The ATE task might be relatively straightforward and offer limited knowledge for improvement.

Model	Lap14	Rest14	Rest15	Rest16
GRACE	87.93	85.45	-	-
SPAN	83.35	82.38	-	-
GPT2_{med}	82.04	75.94	-	-
BARTABSA	83.52	87.07	75.48	-
IT-MTL	76.93	-	74.03	79.41
InstructABSA1	91.40	**92.76**	75.23	81.48
InstructABSA2	**92.30**	92.10	76.64	80.32
Ours ($k = 4$)	90.05	90.72	**79.93**	**83.85**

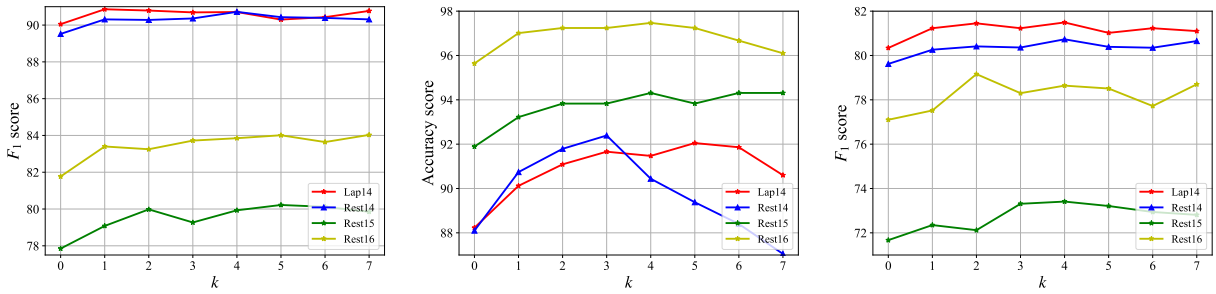
Table 2: ATE subtask results in terms of the F_1 scores (%).

Model	Lap14	Rest14	Rest15	Rest16
SPAN	81.39	89.95	-	-
ABSA-DeBERTa	82.76	89.46	-	-
LSAT	86.31	**90.86**	-	-
RACL-BERT	73.91	81.61	74.91	-
Dual-MRC	75.97	82.04	73.59	-
InstructABSA1	80.62	86.25	83.02	89.1
InstructABSA2	81.56	85.17	84.5	89.43
Ours ($k = 4$)	**91.47**	90.44	**94.31**	**97.47**

Table 3: ATSC subtask results in terms of the accuracy (%).

[1]https://huggingface.co/google/flan-t5-base

[2]https://huggingface.co/allenai/tk-instruct-base-def-pos



(a) ATE (b) ATSC (c) ASPE

Figure 5: The impact of the number of examples on performance (%) on **Rest14**.

Model	Lap14	Rest14	Rest15	Rest16
GRACE	75.97	78.07	-	-
SPAN	68.06	74.92	-	-
GPT2_{med}	53.55	60.07	-	-
GAS	68.64	77.13	66.78	73.64
Seq2Path	70.00	77.01	68.35	75.87
BARTABSA	67.37	73.56	66.61	-
IT-MTL	66.07	-	67.06	74.07
InstructABSA1	78.89	76.16	69.02	74.24
InstructABSA2	79.34	79.47	69.39	73.06
Ours ($k = 4$)	**81.49**	**80.73**	**73.41**	**78.64**

Table 4: ASPE subtask results in terms of the F_1 scores (%).

For the ATSC task, the proposed method achieved optimal performances on the **Lap14**, **Rest15**, and **Rest16** datasets, significantly outperforming the baselines (5-10% improvement) and achieving a similar performance to that of LSAT on the **Rest14** dataset.

For the ASPE task, the proposed method surpassed the baselines on all four datasets, obtaining F_1 scores of 81.49, 80.73, 73.41, and 78.64.

Moreover, for the ATSC task, the model performed better on smaller datasets (**Rest15**, **Rest16**) than on larger datasets (**Lap14** and **Rest14**).

4.5 Number of Examples

To explore the impact of the number of examples on the performance, we used a retriever to extract different numbers of examples for model inference. The maximum number of examples was 7 to prevent the input text from exceeding the maximum sequence length of the model. Figure 5 shows the results.

Overall, the performances show two trends as the number of examples increases: (i) rising and then falling or (ii) rising. The use of examples allows the model to achieve a significant performance gain. As the number of examples increases, the model can acquire more knowledge from beneficial examples. However, this performance enhancement trend is not always significant, and an increase in the number of examples can harm the model performance. Owing to the limited capacity of the retriever and limited number of beneficial examples for the query in the example pool, not all retrieved examples were beneficial or harmful to the query.

4.6 Ablation Study

Table 5 presents the results of the ablation study used to investigate the effectiveness of each component; **w/o alternating training** means that alternating training schema was not used; **w/o retriever**, the retriever was removed and fixed examples were used; **w/o example**, examples were not used; and **w/o instruction**, instruction prompts were not used. The results demonstrate the effectiveness of each part of the proposed method.

For further analysis, the performance decrease in the term **w/o alternating training** suggests that alternating training schema can better narrow the gap between the retriever and LM. Additionally, because the term **w/o retriever** outperforms the term **w/o example** overall, although inferior to the term **w/o example** in some cases, suggests that the model can learn from the examples, but fixed examples function differently for different target samples and are not conducive to inference for some extreme situations.

4.7 The Role of Fine-tuning Language Models

The Table 6 presents the results of our experiments. For the ATSC task, phrases such as *The pizza is good* and *I think the pizza was good* do not require

Model	ATE (F_1)				ATSC (Acc)				ASPE (F_1)			
	L14	**R14**	**R15**	**R16**	**L14**	**R14**	**R15**	**R16**	**L14**	**R14**	**R15**	**R16**
Ours ($k = 4$)	90.05	90.72	79.93	83.85	91.47	90.44	94.31	97.47	81.49	80.73	73.41	78.64
w/o alternating training	89.82	89.13	77.46	80.16	91.37	89.47	93.46	96.55	78.05	77.63	69.05	77.61
w/o retriever	89.77	88.34	75.59	80.06	87.11	87.87	91.64	96.09	79.16	78.11	70.78	77.55
w/o example	89.74	88.03	76.00	79.79	87.59	87.95	91.52	95.86	77.05	76.20	68.29	76.02
w/o instruction	88.68	87.56	74.08	79.58	87.30	86.60	90.67	95.75	76.66	74.97	68.62	76.01

Table 5: Ablation study (%). L14, R14, R15, and R16 denote the datasets **Lap14**, **Rest14**, **Rest15**, and **Rest16**, respectively.

Model	ATE (F_1)				ATSC (Acc)				ASPE (F_1)			
	L14	**R14**	**R15**	**R16**	**L14**	**R14**	**R15**	**R16**	**L14**	**R14**	**R15**	**R16**
Frozen LM ($k = 4$)	41.96	25.85	40.77	39.00	48.83	65.74	55.21	62.92	29.04	12.46	29.83	25.15
Ours ($k = 4$)	90.05	90.72	79.93	83.85	91.47	90.44	94.31	97.47	81.49	80.73	73.41	78.64
$\uparrow$ (%)	46.60	28.49	51.01	46.51	53.38	72.69	58.54	64.55	35.64	15.43	40.63	31.98

Table 6: Exploring the improvement of fine-tuning language models. **Frozen LM** indicates that the parameters of LM are not updated during the training phase, which is similar to the retrieval methods used on LLM. $\uparrow$ (%) represents how much performance improvement ratio our method has compared to **Frozen LM**.

Model	Size	ATE	ASTC	ASPE
T5-base	223M	89.91	87.34	79.12
T5-large	738M	91.71	91.04	81.23
Flan-T5-base	248M	90.72	90.44	80.73
Flan-T5-large	783M	92.64	92.97	82.61

Table 7: Performance (%) with different language models on **Rest14**.

the LM to focus on semantic similarity or the connection between aspects and opinions. The LM can infer the result based on the retrieved example sentence's *output: positive*, resulting in decent performance even without training. For the ATE task, the performance is poorer due to the low overlap in output results. For the more complex ASPE task, the retrieved example sentences should provide diverse assistance to the test sentences (including structural assistance and overlapping aspects). However, an untrained LM lacks this ability and tends to search for answers from the context rather than making inferences.

We further explored the retrieval results of the method that only trains the retriever. Compared to the proposed method, it favors retrieving example sentences with overlapping results and rarely retrieving other types of examples mentioned in the Case Study section. This limitation may hinder its effectiveness in handling complex tasks that require a deeper understanding of the context and relationships between different aspects.

4.8 Effectiveness on Different Language Models

The efficacy of our proposed method was examined across various backbone language models. The performance metrics for different variants in terms of sizes and types are detailed in Table 7. The results underscore the robust effectiveness of our proposed method across commonly used backbone models. Notably, employing larger models leads to discernible performance improvements. Particularly, the instruction pre-training variant, flan-t5, exhibits more substantial gains. This can be attributed to the smaller gap between the pre-training and fine-tuning phases.

4.9 Case Study

Table 8 provides an overview of the retrieval results for specific queries in the ASPE task. For clarity, only the first result retrieved is presented for each query. Notably, examples with IDs 1, 3, and 4 exhibit a structural similarity to the query. For instance, ID 2 mirrors the query's aspects with the same sentiment polarity. While these instances showcase evidently favorable retrieval outcomes, there are cases where the utility of results is unclear. For instance, example ID 5 does not seem directly related to the query; however, the prediction for that particular query is accurate. Determining whether this is due to the LM disregarding irrelevant examples or if the example somehow contributes to the LM's understanding remains an open question.

ID	Query	Example
1	Green Tea creme brulee is a must!	**input:** (The asparagus, truffle oil, parmesan bruschetta is a winner!) **output:** asparagus, truffle oil, parmesan bruschetta: positive
2	Serves really good sushi.	**input:** Best. Sushi. Ever. **output:** Sushi: positive
3	AMAZING.	**input:** Unbelievable. **output:** noaspectterm: none
4	The food was almost always EXCELLENT.	**input:** The food was good. **output:** food: positive
5	I never had an orange donut before so I gave it a shot.	**input:** this one is definintely my least favorite. **output:** noaspectterm: none

Table 8: Best retrieval results in the ASPE task.

5 Conclusion

In this study, we proposed a retrieval-based example mining method for instructional learning in ABSA tasks to improve the performance by selecting effective examples. The proposed method conducts the alternating training of a retriever and LM by employing a two-stage training framework and iterative evolution training scheme. Experiments validated its effectiveness across ATE, ATSC, and ASPE tasks, outperforming existing baseline models.

Future work will extend the proposed method to other tasks and models and refine the training strategies to achieve further performance gains.

Limitations

There are three main limitations to our work compared to previous efforts:

1. The mechanism of retrieval is based on the likelihood score given by the language model, however, this score only focuses on the improvement of the model's performance, and the syntactic mechanism of its work remains to be explored.

2. The choice of the number k of examples constrains the performance of the model. Although we explored the impact of using different numbers of examples on the overall performance, it is undeniable that the optimal number of examples varies for different test inputs. Model performance would be further improved if the appropriate number of examples could be customized for each input.

3. The proposed method is only experimented on the English dataset. Whether it works equally well and whether the retrieved examples have commonalities is still unknown in other languages (e.g., Russian, French, Chinese, etc.). It remains to be explored whether the method will work on mixed language and multilingual datasets.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (NSFC) under Grant Nos.61966038 and 62266051, the Ministry of Science and Technology, Taiwan, ROC, under Grant No.MOST 111-2628-E-155-001-MY2, and the Exam-Exempted Postgraduate Research and Innovation Foundation of Yunnan University under Grant No.TM-23237123. The authors would like to thank the anonymous reviewers for their constructive comments.

References

Zhuang Chen and Tieyun Qian. 2020. Relation-aware collaborative learning for unified aspect-based sentiment analysis. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL-2020)*, pages 3685–3694.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv: 2210.11416*.

Rajarshi Das, Manzil Zaheer, Dung Thai, Ameya Godbole, Ethan Perez, Jay Yoon Lee, Lizhen Tan, Lazaros Polymenakos, and Andrew McCallum. 2021. Case-based reasoning for natural language queries over knowledge bases. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP-2021)*, pages 9594–9611.

Ehsan Hosseini-Asl, Wenhao Liu, and Caiming Xiong. 2022. A generative language model for few-shot aspect-based sentiment analysis. In *Findings of the Association for Computational Linguistics: NAACL 2022 - Findings*, pages 770 – 787.

Minghao Hu, Yuxing Peng, Zhen Huang, Dongsheng Li, and Yiwei Lv. 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 (ACL-2019)*, pages 537–546.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen tau Yih. 2020. Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv: 2004.04906*.

Junlong Li, Zhuosheng Zhang, and Hai Zhao. 2022. Self-prompting large language models for zero-shot open-domain qa. *arXiv preprint arXiv: 2212.08635*.

Xiaonan Li, Kai Lv, Hang Yan, Tianyang Lin, Wei Zhu, Yuan Ni, Guotong Xie, Xiaoling Wang, and Xipeng Qiu. 2023. Unified demonstration retriever for in-context learning. *arXiv preprint arXiv: 2305.04320*.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2022. What makes good in-context examples for gpt-3? In *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 100–114.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *Proceedings of the 7th International Conference on Learning Representations, (ICLR-2019)*.

Huaishao Luo, Lei Ji, Tianrui Li, Nan Duan, and Daxin Jiang. 2020. Grace: Gradient harmonized and cascaded labeling for aspect-based sentiment analysis. In *Findings of the Association for Computational Linguistics Findings of ACL: EMNLP 2020*, pages 54–64.

Man Luo, Xin Xu, Zhuyun Dai, Panupong Pasupat, Mehran Kazemi, Chitta Baral, Vaiva Imbrasaite, and Vincent Y Zhao. 2023. Dr.icl: Demonstration-retrieved in-context learning. *arXiv preprint arXiv: 2305.14128*.

Xiang Luo, Zhiwen Tang, Jin Wang, and Xuejie Zhang. 2024. Duetsim: Building user simulator with dual large language models for task-oriented dialogues. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 5414–5424.

Yue Mao, Yi Shen, Jingchao Yang, Xiaoying Zhu, and Longjun Cai. 2022. Seq2Path: Generating sentiment tuples as paths of a tree. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2215–2225.

Yue Mao, Yi Shen, Chao Yu, and Longjun Cai. 2021. A joint training dual-mrc framework for aspect based sentiment analysis. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence (AAAI-2021)*, volume 35, pages 13543–13551.

Ricardo Marcondes Marcacini and Emanuel Silva. 2021. Aspect-based sentiment analysis using bert with disentangled attention. In *Proceedings of the LatinX in AI (LXAI) Research workshop at ICML 2021*.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. 2022. Cross-task generalization via natural language crowdsourcing instructions. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL-2022)*, pages 3470–3487.

Jianmo Ni, Gustavo Hernández Ábrego, Noah Constant, Ji Ma, Keith B. Hall, Daniel Cer, and Yinfei Yang. 2022. Sentence-t5 (st5): Scalable sentence encoders from pre-trained text-to-text models. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1864–1874.

Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Ion Androutsopoulos, Suresh Manandhar, Mohammad AL-Smadi, Mahmoud Al-Ayyoub, Yanyan Zhao, Bing Qin, Orphée De Clercq, Véronique Hoste, Marianna Apidianaki, Xavier Tannier, Natalia Loukachevitch, Evgeniy Kotelnikov, Nuria Bel, Salud María Jiménez-Zafra, and Gülşen Eryiğit. 2016. Semeval-2016 task 5: Aspect based sentiment analysis. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 19–30.

Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Suresh Manandhar, and Ion Androutsopoulos. 2015. Semeval-2015 task 12: Aspect based sentiment analysis. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval-2015)*, pages 486–495.

Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. 2014. Semeval-2014 task 4: Aspect based sentiment analysis. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval-2014)*, pages 27–35.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv: 1910.10683*.

Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3:333–389.

Ohad Rubin, Jonathan Herzig, and Jonathan Berant. 2021. Learning to retrieve prompts for in-context learning. *arXiv preprint arXiv: 2112.08633*.

Kevin Scaria, Himanshu Gupta, Siddharth Goyal, Saurabh Arjun Sawant, Swaroop Mishra, and Chitta Baral. 2023. Instructabsa: Instruction learning for aspect based sentiment analysis. *arXiv preprint arXiv:2302.08624*.

Siddharth Varia, Shuai Wang, Kishaloy Halder, Robert Vacareanu, Miguel Ballesteros, Yassine Benajiba, Neha Anna John, Rishita Anubhai, Smaranda Muresan, and Dan Roth. 2022. Instruction tuning for few-shot aspect-based sentiment analysis. *arXiv preprint arXiv:2210.06629*.

Liang Wang, Nan Yang, and Furu Wei. 2023. Learning to retrieve in-context examples for large language models. *arXiv preprint arXiv: 2307.07164*.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Anjana Arunkumar, Arjun Ashok, Arut Selvan Dhanasekaran, Atharva Naik, David Stap, Eshaan Pathak, Giannis Karamanolakis, Haizhi Gary Lai, Ishan Purohit, Ishani Mondal, Jacob Anderson, Kirby Kuznia, Krima Doshi, Maitreya Patel, Kuntal Kumar Pal, Mehrad Moradshahi, Mihir Parmar, Mirali Purohit, Neeraj Varshney, Phani Rohitha Kaza, Pulkit Verma, Ravsehaj Singh Puri, Rushang Karia, Shailaja Keyur Sampat, Savan Doshi, Siddhartha Mishra, Sujan Reddy, Sumanta Patro, Tanay Dixit, Xudong Shen, Chitta Baral, Yejin Choi, Noah A. Smith, Hannaneh Hajishirzi, and Daniel Khashabi. 2022. Super-naturalinstructions: Generalization via declarative instructions on 1600+ nlp tasks. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP-2022)*, pages 5085–5109.

Hang Yan, Junqi Dai, Tuo Ji, Xipeng Qiu, and Zheng Zhang. 2021. 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 (ACL-IJCNLP-2021)*, page 2416–2429.

Heng Yang and Ke Li. 2021. Improving implicit sentiment learning via local sentiment aggregation. *arXiv preprint arXiv:2110.08604*.

Li Yuan, Jin Wang, Liang-Chih Yu, and Xuejie Zhang. 2020. Graph attention network with memory fusion for aspect-level sentiment analysis. In *Proceedings of the 1st conference of the asia-pacific chapter of the association for computational linguistics and the 10th international joint conference on natural language processing*, pages 27–36.

Li Yuan, Jin Wang, Liang-Chih Yu, and Xuejie Zhang. 2022. Syntactic graph attention network for aspect-level sentiment analysis. *IEEE Transactions on Artificial Intelligence*.

Lei Zhang and Bing Liu. 2017. Sentiment analysis and opinion mining. *Encyclopedia of Machine Learning and Data Mining*, pages 1152–1161.

Wenxuan Zhang, Xin Li, Yang Deng, Lidong Bing, and Wai Lam. 2021. Towards generative 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 (ACL-IJCNLP-2021)*, pages 504–510.

Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. 2022. Automatic chain of thought prompting in large language models. *arXiv preprint arXiv:2210.03493*.

A Baseline Models

The baseline models that emerged from the comparative experiments were categorized into generative and non-generative models. This section describes these baseline models in detail.

1) Generative methods:

- **GPT2$_{\text{med}}$** (Hosseini-Asl et al., 2022) utilizes unidirectional self-attention and language modeling loss to capture contextual representations and leverage supervision during training.
- **BARTABSA** (Yan et al., 2021) formulates the ABSA extraction and classification tasks as a unified index generation problem.
- **GAS** (Zhang et al., 2021) formulates each task as a generative problem and predictive normalization strategy to optimize the generated outputs.
- **IT-MTL** (Varia et al., 2022) treats ABSA as a sequence-to-sequence modeling task based on instruction tuning, achieving excellent performance with a few shots.
- **InstructABSA** (Scaria et al., 2023) constructs fixed instruction prompts for different tasks to train the Tk-instruct model (Wang et al., 2022), and the examples in the prompts are obtained from combinations of sentiment-positive, -negative, and -neutral examples. InstructABSA1 includes two positive and two negative sentiment examples, while InstructABSA2 adds two neutral sentiment examples.

2) Non-generative methods:

- **SPAN** (Hu et al., 2019) use a span-based labeling scheme to find and classify opinion targets in a sentence, which mitigates the problem of sentimental inconsistencies at the span level.
- **GRACE** (Luo et al., 2020) employs cascade labeling to enhance the interaction between aspect terms and mitigates the labeling imbalance through a gradient harmonization approach.
- **ABSA-DeBERTa** (Marcacini and Silva, 2021) uses a decoupled attention mechanism to separate location and content vectors for sentiment analysis.

- **LSAT** (Yang and Li, 2021) introduces a local sentiment aggregation paradigm that facilitates fine-grained sentiment consistency modeling.
- **RACL-BERT** (Chen and Qian, 2020) allows subtasks to work together in stacked multilayer networks via multitask learning and relationship propagation mechanisms.
- **Dual-MRC** (Mao et al., 2021) converts the original triplet extraction task into two machine reading comprehension (MRC) problems, and jointly trains multiple subtasks.
- **Seq2Path** (Mao et al., 2022) transforms the generation order of sentiment tuples into tree paths. This approach not only effectively addresses the issue of one aspect entity corresponding to multiple opinion words, but also ensures that the generation of each path is independent.