In-Context Example Retrieval from Multi-Perspectives for Few-Shot Aspect-Based Sentiment Analysis

Qianlong Wang[1♮], Hongling Xu[1♮], Keyang Ding[1], Bin Liang[2], Ruifeng Xu[1,3,4*]
[1]Harbin Institute of Technology (Shenzhen)
[2]The Chinese University of Hong Kong [3]Peng Cheng Laboratory, China
[4]Guangdong Provincial Key Laboratory of Novel Security Intelligence Technologies, China
qlwang15@outlook.com, xuhongling1114@gmail.com, keyang.ding@stu.hit.edu.cn
bin.liang@cuhk.edu.hk, xuruifeng@hit.edu.cn

Abstract

In this paper, we focus on few-shot aspect-based sentiment analysis (ABSA) and try to solve it with in-context learning (ICL) paradigm. However, the effectiveness of ICL is highly affected by retrieved in-context examples. Previous works generally leverage the semantic similarity between the candidate examples and test input to retrieve examples. However, they may yield sub-optimal results for this task. This is because considering only the overall semantic perspective may leave some useful examples, which have syntactic structural relevance to the test input or share identical sentiments and similar aspects to one unretrievable. To address this shortcoming, we advocate retrieving in-context examples for few-shot ABSA by simultaneously considering three perspectives, overall semantics, syntactic structure relevance, and aspect-sentiment semantics. To achieve this, we construct positive and negative pairs from these three perspectives and train the demonstration retriever using contrastive learning. Experimental results on four ABSA datasets show that our retrieval framework can significantly outperform baselines across the board. Moreover, to understand factors influencing ICL performance on few-shot ABSA, we conduct extensive analysis in various scenarios, which can inspire and advance future research.

Keywords: Few-shot Aspect-based Sentiment Analysis, In-Context Learning, Large Language Models

1. Introduction

Aspect-based sentiment analysis (ABSA) aims to identify sentiment polarities of aspects mentioned in reviews. Generally, ABSA involves two foundational subtasks, aspect extraction (AE) and aspect sentiment classification (ASC). For example, given a review "*Food is good, but service is dreadful.*", AE aims to detect two aspects "*food*" and "*service*", and ASC predicts their corresponding sentiment polarities as *positive* and *negative*, respectively.

Recent studies (Xu et al., 2018; Xue and Li, 2018; Li et al., 2019a) proposed deep neural models to tackle ABSA. These models can achieve satisfactory results by exploiting extensive labeled data to optimize parameters. With the advent of pre-trained language models (PLMs) such as BERT (Devlin et al., 2019), the performance of these ABSA models is further considerably improved (Li et al., 2019b; Mao et al., 2021). When armed with PLMs, thousands of fine-grained annotations are still required for fine-tuning task-specific parameters to reach state-of-the-art performance on ABSA (Xu et al., 2019). If this condition is not met, these ABSA models will perform poorly. For example, our probing experiments on the Laptop dataset (Pontiki et al., 2014) show that BERT only attains a 13.8% F1 score while having 100 labeled reviews. However, in real-world scenarios such as e-commerce, manually labeling sufficient review data for each product or domain can be expensive and time-consuming. Thus, we focus on few-shot ABSA in this work, which aims at handling ABSA with only a small number of labeled data.

Large language models (LLMs) (Brown et al., 2020; Touvron et al., 2023) have shown an impressive few-shot ability on several NLP tasks. To expect LLMs to perform better on few-shot tasks, in-context learning (ICL) (Dong et al., 2022) paradigm is becoming a flourishing research direction. This paradigm can generate a prediction of the test input by conditioning on few-shot input-output examples (also known as *in-context examples* or *demonstrations*), without requiring any updates to parameters. Previous studies (Liu et al., 2022; Min et al., 2022) found that LLMs are highly sensitive to the choice of in-context examples. One typical strategy for retrieving helpful in-context examples is to leverage the overall semantic similarity between the candidate examples and test input.

Given a test input "*The coffee aroma here is unpleasant.*", this strategy will retrieve the first and second examples in Table 1 as in-context examples because of the overall semantic relevance ("*cafe*" and "*coffee*"). However, these two retrieved examples may not enable LLMs to handle ABSA adequately. This is because other valuable examples having similar syntactic structural organization or

♮ Equal contribution.
* Corresponding author.

Candidate Examples	**POS Sequence**	**Label**
(1) The ambiance at the *cafe* is refreshing.	DT NN IN DT NN VBZ JJ .	(ambiance, positive)
(2) The *coffee* here tastes like any other.	DT NN RB VBZ IN DT JJ .	(coffee, neutral)
(3) The staff here seems disinterested.	*DT NN RB VBZ JJ .*	(staff, negative)
(4) I wasn't impressed by the *beverage*.	PRP VBD RB VBN IN DT NN .	(beverage, negative)
....	....	....

Table 1: Some candidate examples. The first and second elements in parentheses are the aspect and corresponding sentiment polarity, respectively.

aspect-sentiment clues are lacking. In other words, for ABSA, retrieving demonstrations requires considering syntactic structure relevance and aspect-sentiment semantics rather than only overall semantics. Here, the syntactic structure relevance can be exploited to find examples with similar syntactic patterns to the test input, which is important for identifying aspect-sentiment pairs. For example, this relevance can retrieve the third example as its POS sequence is similar to that of the test input ("*DT NN RB VBZ JJ .*"[1] similar to "*DT NN NN RB VBZ JJ .*"). This example is informative for extracting aspect "*coffee*" and recognizing its modifier "*unpleasant*" in the test input due to similar syntactic structures. The aspect-sentiment semantics can help retrieve examples with identical sentiments and similar aspects, thus providing strong clues for understanding the fine-grained sentiment of the test input. For example, if considering this point, the fourth example will be retrieved due to its semantics ("*coffee*" similar to "*beverage*") and sentiment ("*unpleasant*" similar to "*negative*"). This example contributes to comprehending the negative sentiment of "*coffee*" in the test input. In a nutshell, in the ABSA task, the retrieval of in-context examples should also consider syntactic structure relevance and aspect-sentiment semantics rather than only overall semantics to provide more effective and valuable guidance to ICL.

Inspired by this, we retrieve useful in-context examples for few-shot ABSA by simultaneously considering three perspectives: overall semantics, syntactic structure relevance, and aspect-sentiment semantics. To this end, we construct positive and negative pairs from these three perspectives and train the demonstration retriever via contrastive learning (Chen et al., 2020). Specifically, given a set of candidate examples, (1) for each example, we treat its augmented version as the positive sample and those of other examples as negative samples. In this way, the retriever can yield comprehensive feature representations and distinguish each example in terms of overall semantics. (2) We take the POS sequence of each example (derived from the syntactic structure) as the positive sample and those of other examples as negative samples. By this, syntactic structure relevance is modeled which can be used to find in-context examples with similar aspect-sentiment organization to the test input. (3) for each example, we consider its aspect-sentiment semantics as the positive sample and those of other examples as negative samples. Based on this, the retriever pays more attention to aspects and sentiments, which hope to retrieve examples with the same sentiment and similar aspects, thus providing inference clues for the test input. These constructed positive and negative pairs enable the retriever to compare and rank candidate examples in terms of overall content, syntactic structure, and aspect-sentiment semantics. As a result, the probability of retrieving examples that are highly relevant and valuable to the test input will be increased. Subsequently, we select the top-k examples as in-context examples and feed them to LLMs for guiding inference of the test input.

We summarize our contribution as follows:

- To the best of our knowledge, we are the first to study the retrieval of in-context examples for application in few-shot ABSA. To make these examples more effective, we consider overall semantics, syntactic structure relevance, and aspect-sentiment semantics to retrieve them.
- We conduct evaluation experiments on four ABSA datasets. The results show that our retrieval framework[2] brings significant improvements over other peers.
- To analyze the factors affecting the performance when solving few-shot ABSA with ICL, we conduct more discussions on in-context examples, which can shed light on future work.

2. Related Work

2.1. Aspect-Based Sentiment Analysis

ABSA is the task of identifying aspects and associated sentiment polarities in review texts. It involves two fundamental subtasks (*i.e.,* AE and ASC). Earlier works on ABSA focused on combining word embeddings and neural network models (Xu et al.,

[1] We see that the lexeme NN corresponds to aspect, while the lexeme JJ refers to modifier.

[2] The code and data are available through the following repository `https://github.com/qlwang25/multi-perspectives_example_retrieval`.

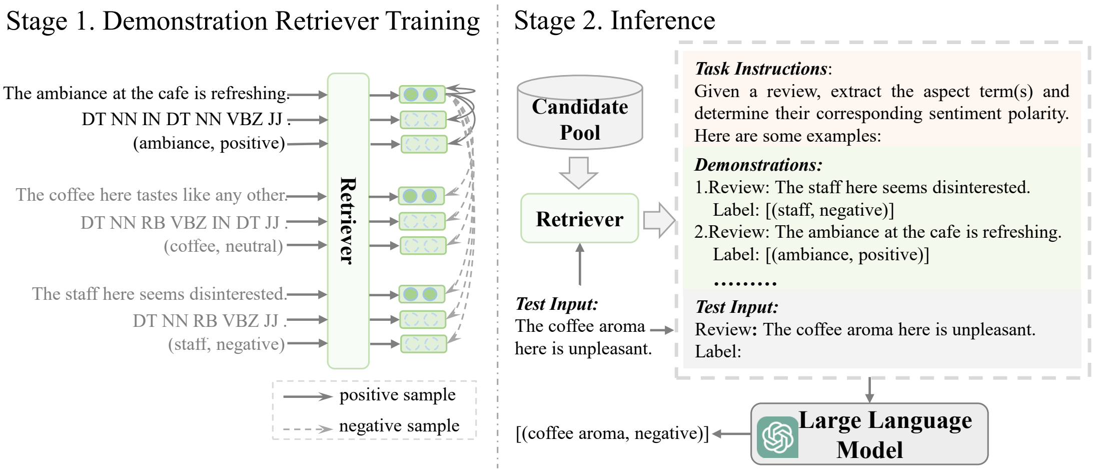

Stage 1. Demonstration Retriever Training
Stage 2. Inference
The ambiance at the cafe is refreshing.
DT NN IN DT NN VBZ JJ
(ambiance, positive)
The coffee here tastes like any other.
DT NN RB VBZ IN DT JJ
(coffee, neutral)
The staff here seems disinterested.
DT NN RB VBZ JJ
(staff, negative)
Retriever
positive sample
negative sample
Candidate Pool
Retriever
Test Input:
The coffee aroma here is unpleasant.
Task Instructions:
Given a review, extract the aspect term(s) and determine their corresponding sentiment polarity. Here are some examples:
Demonstrations:
1.Review: The staff here seems disinterested.
Label: [(staff, negative)]
2.Review: The ambiance at the cafe is refreshing.
Label: [(ambiance, positive)]
....
Test Input:
Review: The coffee aroma here is unpleasant.
Label:
Large Language Model
[(coffee aroma, negative)]

Figure 1: Overview of our proposed framework.

2018; Xue and Li, 2018). With the advent of PLMs (Devlin et al., 2019; Lewis et al., 2020), several models are proposed for ABSA using PLMs as the feature extractor (Li et al., 2019b; Wang et al., 2020; He et al., 2022). Although commendable efforts have been devoted to improving performance, these neural models often require a large amount of fine-grained "aspect-sentiment" annotated data on each domain, which is laborious. To alleviate this problem, a few studies (Hosseini-Asl et al., 2022; Wang et al., 2023c) attempted to address ABSA in the few-shot setting. Despite their progress, they generally use extensive domain-specific unlabeled data (*e.g.*,100k) to continue post-training to bridge the domain gap. It may be a limitation because (1) for new domains, there is a lack of substantial unannotated data available, and (2) data privacy concerns may restrict acquisition. Besides, they excessively tailor themselves to the limited training data, hindering their generalization to unseen data. Unlike them, in this paper, we use the ICL paradigm to tackle few-shot ABSA, which can alleviate these shortcomings and yield better results.

2.2. In-Context Learning

ICL (Brown et al., 2020) is an emerging learning paradigm, which allows LLMs to perform several downstream tasks with few-shot examples without updating the model parameters. The existing works on ICL can be broadly divided into two streams. The first stream focuses on understanding the underlying mechanisms and principles of this paradigm (Xie et al., 2021; Min et al., 2022; Garg et al., 2022). For instance, Wang et al. (2023a) investigated the working mechanism of ICL through an information flow lens. The second stream explores different strategies for selecting and formatting in-context examples for LLMs. The supporting point behind this type of work is that the performance of ICL strongly relies on the example surface, including example selection (Gonen et al., 2022; Rubin et al., 2022; Zhang et al., 2022; Agrawal et al., 2023), example ordering (Liu et al., 2022), example formatting (Honovich et al., 2023; Wang et al., 2023b), and so on. Among them, example selection (*a.k.a.*, demonstration retrieval) has sparked considerable interest and produced some representative literature. For example, Liu et al. (2022) used a KNN-based method to retrieve similar demonstrations to the test input. Our work proposes to consider multi-perspectives to retrieve high-quality in-context examples for few-shot ABSA and thus falls into the second stream.

3. Our Method

3.1. Problem Definition

Given a candidate pool $P = \{(x_i, y_i)\}_{i=1}^{K}$ of input-output examples, where x is a review text, y is the corresponding label tuple, and K is a relatively small size (*e.g.*, 100), our goal is to train a demonstration retriever R. For each test input x_{test}, it can retrieve k-shot in-context examples $Demo = \{(x_j, y_j)\}_{j=1}^{k}$ from P. Based on the test input and in-context examples, a frozen language model predicts an output y'_{test}.

Thus, the primary objective of this retriever is to retrieve k representative in-context examples from P such that the predicted output y'_{test} is as close as possible to the ground-truth output y_{test}.

3.2. Overview

Figure 1 depicts the proposed framework that comprises two stages. Stage 1: We construct positive and negative sample pairs from three perspectives to train the demonstration retriever via contrastive learning. Stage 2: For each test input, the trained retriever is exploited to retrieve k-shot valuable in-context examples from the candidate pool. These retrieved examples along with task instructions and test input are fed to LLM for inference.

3.3. Demonstration Retriever Training

Provided with a candidate pool P and test input, the demonstration retriever aims to retrieve k-shot relevant examples from P to help LLM decode the target output. Regrettably, previous works (Liu et al., 2022; Min et al., 2022) are concerned more with the relevance of the overall semantics and not as much with syntactic structure relevance and aspect-sentiment semantics. They may lead to low-quality demonstrations retrieved and sub-optimal results because ABSA depends heavily on syntax and sentiment knowledge (Wang et al., 2020; Li et al., 2022). Thus, we consider that providing demonstrations with similar syntactic structures or aspect-sentiment semantics can provide more obvious hints to infer aspects and sentiments in the test input. In this work, we focus on learning a demonstration retriever via contrastive learning, which considers three perspectives (overall semantics, syntactic structure relevance, and aspect-sentiment semantics) simultaneously to retrieve representative in-context examples for few-shot ABSA.

Contrastive learning aims to make the retriever learn representations by distinguishing between similar and dissimilar samples. It will pull similar (or positive) pairs closer in a feature space and push dissimilar (or negative) pairs farther apart. Supposed a set of paired samples $D = \{(x_i, x_i^+)\}_{i=1}$, where x_i and x_i^+ are positive pairs, we take the cross-entropy objective with in-batch negatives (Chen et al., 2017). Hence, the training objective for (x_i, x_i^+) in a mini-batch of N pairs is defined as follows:

$$\ell_{(x_i, x_i^+)} = -\log \frac{e^{\mathrm{sim}(x_i, x_i^+)/\tau}}{\sum_{j=1}^{N} e^{\mathrm{sim}(x_i, x_j^+)/\tau}} \tag{1}$$

where τ is a temperature hyper-parameter and $\mathrm{sim}(\cdot,\cdot)$ denotes cosine similarity function[3]. Based on this, we construct positive and negative pairs from three perspectives so that the demonstration retriever can perceive overall semantics, syntactic structure relevance, and aspect-sentiment semantics of examples.

Overall Semantics. A fundamental condition for the retriever to be effective is to understand each sample and distinguish which texts are similar and which are different. This is because retrieving examples that are semantically similar to the test input may be helpful. For this purpose, we view each sample and its augmented version as a positive pair, and the augmented versions of other samples as negative pairs.[4] The motivation behind this is that a text and its corresponding augmented one tend to have consistent context semantics. Here, we consider the encoded output of the sample as overall semantics (*i.e.*, contextualized embedding corresponding to the [CLS] token) and execute an additional different dropout operation to obtain an augmented version (Chen et al., 2020). Specifically, a sample is fed into BERT twice with dropout (which can randomly set the output of some neurons to zero) to obtain original and augmented contextualized embeddings, respectively.

Syntactic Structure Relevance. When applying ICL to ABSA, retrieving in-context examples based on overall semantic similarity alone may be sub-optimal. This is because solving ABSA also requires reference examples with similar syntactic patterns to the test input, which help to recognize aspect-sentiment label tuples (Dai et al., 2021; Oh et al., 2021). Linguistically speaking, similar syntactic patterns usually imply similar ways of organizing content. For instance, a noun phrase may be an aspect while an adjective or verb associated with it may be sentiment modification. Thus, such examples could guide the reasoning process of LLMs, just as analogical reasoning does (Yasunaga et al., 2023).

Here, we use the POS sequence of text to stand for its syntactic structure.[5] To model syntactic structural relevance, we treat each sample and its POS sequence as positive pairs, while other POS sequences from other samples are treated as negative pairs. As a result, by comparing the test input with the POS sequence of each candidate example, the retriever can find examples with similar syntactic patterns. These examples may be highly correlated with the test input in terms of organizational aspects and sentiment, thus facilitating LLM to extract aspects and determine their sentiments.

Aspect-Sentiment Semantics. In addition to overall semantic and syntactic structural relevance, the retriever should retrieve examples from the candidate pool that share the same sentiment and similar aspects as the test input. Such examples would provide better clues in ICL for completing ABSA, especially the ASC subtask. Based on this line of thought, we define *aspect-sentiment semantics*,

[3] Calculating similarity requires obtaining contextualized embedding for each sample x_i using BERT. We omit the symbol BERT in this function for simplicity.

[4] Here, "*augmented version*" refers to a transform or data augmentation of the original sample.

[5] We use NLTK to obtain the POS sequence.

which aims to highlight the aspects and the associated sentiment in the review.[6]

To derive aspect-sentiment semantics, we convert the label tuple to text with a template, "*In this sentence, the aspect* $aspect$ *is* $sentiment$", where $aspect$ and $sentiment$ are the first and second elements of the aspect-sentiment tuple, respectively. If more than one tuple exists, it is appended. To make the retriever pay more attention to aspects and sentiment descriptions and enhance relevance retrieval, we push each sample and its aspect-sentiment semantics closer together and pull those of other samples further away. In this way, the retriever could find more relevant examples by comparing aspect-sentiment semantics. They tend to have similar aspects and the same sentiments as the test input, which offers clear hints to reasoning.

Loss Function. Based on the constructed positive and negative pairs from the three perspectives described above, we optimize the retriever using the following loss:

$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N} \ell_{(x_i, x_i^{OS+})} + \ell_{(x_i, x_i^{SR+})} + \ell_{(x_i, x_i^{AS+})} \qquad (2)$$

where N is mini-batch size; $\ell_{(\cdot,\cdot)}$ refers to Eq. 1; x_i^{OS+}, x_i^{SR+}, and x_i^{AS+} are positive samples corresponding to x_i constructed from overall semantics, syntactic structure relevance, and aspect-sentiment semantics perspectives, respectively. Thus, a mini-batch has $N * 3$ positive pairs and $N * (N-1) * 3$ negative pairs for training. This contrastive loss will enable the comparison of candidate examples from three perspectives, thereby increasing the probability of retrieving examples that are highly potentially relevant to the test input.

3.4. Inference

After training the demonstration retriever, it can retrieve examples that satisfy the following conditions: semantically similar to the test input, syntactically structure related to one, potentially have similar aspects and identical sentiment as the test input. In this case, we can select representative in-context examples judiciously for inference. To facilitate inference, we exploit a prompt for test input, which consists of the following components:

Task Instructions. Providing language models with natural language descriptions of the task can improve ICL significantly (Radford et al., 2019). Hence, as shown in the right half of Figure 1, we provide a human-written sentence as a succinct description of ABSA to improve inference.

		Lap14	Rest14	Books	Clothing
Train	#s	1,323	1,848	1,511	1,144
	#t	2,079	3,336	1,972	1,409
Dev	#s	150	150	211	159
	#t	249	272	275	198
Test	#s	417	604	421	318
	#t	638	1,119	563	376

Table 2: Statistics for ABSA datasets. #s and #t denote the number of samples and the number of aspect-sentiment tuples, respectively.

Demonstrations. Given a test input x_{test}, we first compare it with overall semantics, POS sequences, and aspect-sentiment semantics of all candidate examples in cosine similarity in turn, and then perform a final ranking based on the sum of three rankings, and finally retrieve the top-k examples as in-context examples $Demo$. In $Demo$, we organize the input order of the examples in turn according to the ranking. Besides, as shown in Figure 1, each example is wrapped in a uniform format "*Review:* $text$ *Label:* $tuple\ list$". Such wrapping regulates the output format of the test input as LLMs will (very likely) generate outputs that mimic the format of demonstrations.

Test Input. The test review is also wrapped in this format, leaving the label portion blank because we expect LLM to generate an output sequence according to the defined format. This wrapped test is fed into LLM along with the first two components:

$$y'_{test} \leftarrow LLM([TI, Demo, x_{test}]) \qquad (3)$$

here, TI and $Demo$ refer to task instructions and in-context examples, respectively. In this case, LLMs could achieve remarkable performance rivaling previous supervised methods even with only a limited number of demonstrations. Finally, we obtain the content generated by LLM, eliminate redundant outputs, and acquire valid aspect-sentiment pairs.

4. Experiments

4.1. Experimental Settings

Datasets. To evaluate the proposed framework, we conduct experiments on four ABSA datasets. **Lap14** and **Rest14** are constructed based on the original SemEval 2014 Challenges (Pontiki et al., 2014). **Books** and **Clothing** are relabeled based on their 5-core version by Cai et al. (2023).[7] Each sample in four datasets contains a review with one or multiple aspect-sentiment tuples. The statistics

[6] The overall semantics tends to provide broader contextual information for text, while aspect-sentiment semantics focuses on the specific aspect and sentiment.

[7] The initial set of sentiment labels is [1, 2, 3, 4, 5], where 1, 3, and 5 indicate the most negative, neutral, and the most positive, respectively.

	Lap14			Rest14			Books			Clothing		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
FT-ALL	58.37	62.95	60.57	69.82	75.65	72.62	58.67	63.89	61.17	68.66	75.94	72.12
FT-CP	13.95	13.75	13.85	35.87	38.06	36.84	39.65	29.19	33.64	46.83	48.13	47.47
2-shot in-context examples												
Random	38.88	34.01	36.28	52.21	44.23	47.89	24.02	20.78	22.28	30.61	28.98	29.78
BM25	44.76	40.17	42.33	55.58	52.45	53.97	30.54	27.88	29.15	36.68	35.90	36.29
BERT-PT	42.78	38.55	40.56	51.76	51.02	51.39	25.92	22.38	24.02	36.61	34.57	35.56
Instructor	42.33	38.08	40.09	54.95	54.06	54.50	30.58	27.70	29.07	35.93	34.21	35.10
EPR	40.58	34.79	37.46	55.71	50.13	52.77	26.73	23.26	24.88	33.87	32.97	33.42
UDR	40.12	38.01	39.03	55.81	52.37	53.51	28.98	25.61	27.19	34.88	34.21	34.54
Ours	46.44	43.52	**44.93**	58.96	56.12	**57.50**	38.86	35.34	**37.02**	43.75	42.81	**43.28**
4-shot in-context examples												
Random	49.06	41.06	44.70	60.02	50.84	55.05	33.76	27.88	30.54	37.86	34.04	35.85
BM25	51.18	45.29	48.19	62.45	58.71	60.52	36.80	33.92	35.30	52.99	49.46	51.16
BERT-PT	44.75	43.26	45.39	58.78	57.10	57.93	34.18	28.59	31.14	41.66	39.89	40.76
Instructor	50.88	45.14	47.84	59.15	57.46	58.29	38.69	33.74	36.05	49.70	45.47	47.50
EPR	50.38	41.22	45.34	60.20	53.52	56.66	33.67	28.95	31.13	48.22	43.35	45.65
UDR	49.68	44.41	46.88	58.34	56.25	57.27	35.90	31.16	33.36	49.01	44.83	46.82
Ours	51.98	49.21	**50.56**	63.93	61.30	**62.59**	47.41	42.27	**44.69**	56.69	52.92	**54.74**

Table 3: Main results (%). FT-ALL is only used as a reference for the upper limit with no comparability to other methods. The best score across all methods is **highlighted**, and the second-best one is underlined.

of datasets are summarized in Table 2.[8] *Here, to simulate a low-resource scenario and for simplicity, we use the first 100 samples from the training set as a candidate pool from which in-context examples are selected.*

Implementation Details. For the demonstration retriever, we utilize BERT-base-uncased[9] (Devlin et al., 2019) to initialize and AdamW to optimize its parameters. The epoch, batch size, learning rate, dropout rate, and temperature are set to 40, 32, 3e-5, 0.1, and 0.5, respectively. After obtaining the candidate pool, we train the retriever with three random seeds and select the best-trained one based on the performance of the development set. For ICL, unless otherwise specified, we retrieve the top-4 candidates as in-context examples for each test input and use LLaMA-13B[10] (Touvron et al., 2023) as the default LLM for inference. The evaluation metrics are precision (**P**), recall (**R**), and **F1** based on the exact match of the aspect and its sentiment polarity.

4.2. Baselines

Random randomly select examples from the candidate pool as demonstrations. **BM25** (Robertson and Zaragoza, 2009), a classical sparse retriever, ranks candidate examples based on the test input terms appearing in each candidate. **BERT-PT** (Xu et al., 2019) uses post-training methods to fine-tune BERT to enhance the performance of retrieving examples. **Instructor** (Su et al., 2022) is a competitive text embedding model where each text input and an instruction explaining the use case are embedded into a vector. **EPR** (Rubin et al., 2022) exploits feedback from LLMs to distinguish between positive and negative samples and trains the demonstration retriever. **UDR** (Li et al., 2023) is a demonstration retrieval model trained on many tasks where training signals are cast into a unified ranking formulation.

In addition, we fine-tune BERT to offer supervised performance under different data settings, *i.e.*, **FT-ALL** and **FT-CP**. Here, ALL and CP refer to all training data and candidate pool examples, respectively.

4.3. Main Results

We compare our framework with previous methods and report the results in Table 3. From this table, we have the following observations:

(1) Our framework outperforms all competitive methods and achieves substantial improvements on the four datasets. For example, in the 2-shot setting, compared to EPR, our framework obtains 7.47%, 4.73%, 12.14%, and 9.86% improvement of F1 scores on the four datasets, respectively. This suggests that when applying ICL to solving ABSA, considering three perspectives in this paper to retrieve in-context examples is effective.

[8] We remove samples with *conflict* polarity or without explicit aspects.

[9] https://huggingface.co/google-bert/bert-base-uncased

[10] https://huggingface.co/huggyllama/llama-13b

	Lap14			Rest14			Books			Clothing		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
FT-CP	13.95	13.75	13.85	35.87	38.06	36.84	39.65	29.19	33.64	46.83	48.13	47.47
few-shot ABSA solutions												
FSGPT	26.57	30.22	28.27	39.87	42.90	41.34	42.72	32.20	36.72	47.92	49.06	48.48
FSABSA	29.33	34.54	31.77	48.17	48.62	48.39	41.23	41.08	41.15	49.30	50.45	49.86
data augmentation solutions												
ADD	27.31	32.97	29.87	42.15	49.21	45.42	37.21	37.70	37.45	51.44	54.05	53.72
DELETE	28.57	34.22	31.14	42.81	49.21	45.80	40.14	40.22	40.18	47.79	54.86	51.11
SWAP	29.50	34.22	31.69	43.19	50.29	46.48	40.95	40.48	40.71	49.64	53.52	51.51
Ours	51.98	49.21	**50.56**	63.93	61.30	**62.59**	47.41	42.27	**44.69**	56.69	52.92	**54.74**

Table 4: Comparison (%) of our framework with some low-resource solutions. Augmentation operations only involve modifications to the context token of aspects in the review text. **SWAP**: randomly swap two tokens; **ADD**: randomly insert some sampled tokens; **DELETE**: randomly remove some tokens; FSABSA requires large amounts of unlabeled data.

	Rest14			Clothing		
	P	R	F1	P	R	F1
Ours	63.93	61.30	**62.59**	56.69	52.92	**54.74**
w/o OS	61.93	58.89	60.37	55.83	51.83	53.75
w/o SR	63.44	59.24	61.27	57.01	52.15	54.48
w/o AS	62.48	58.80	60.58	55.61	52.21	53.86
w/o TI	59.37	59.42	59.40	49.07	49.46	49.27

Table 5: Ablation study (%).

(2) Among methods, Random has the worst results. This reflects the necessity to retrieve high-quality relevant demonstrations for each test input. We also notice that BM25 is a simple yet competitive baseline. The potential reason is that BM25 may retrieve examples that have identical aspects (*e.g.*, "*coffee*") with the test input.

(3) Among datasets, all methods (except FT) yield the best results on the Rest14 dataset and the worst ones on the Books dataset. This is because, most aspects in the Rest14 are easily extractable food names and have clear sentiment expressions, while aspects in the Books are expressed in a variety of ways (*e.g.*, "*way*" and "*intro*").

(4) Although ICL-based methods lag behind results supervised on the full dataset (see FT-ALL), they are generally far superior to results supervised in equivalent resource scenarios (see FT-CP). For instance, compared to FT-CP, our framework gains nearly 31% improvements in F1 scores on the Lap14 when only using 2-shot examples. This reveals that utilizing ICL to address few-shot ABSA is a simple and effective approach.

4.4. Ablation Study

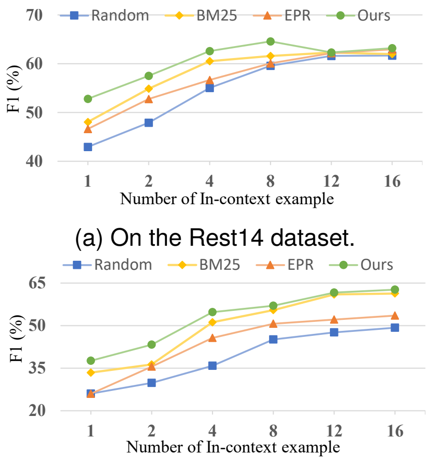
Here, we conduct an ablation study to evaluate the effect of each perspective on the performance. To this end, we discard overall semantics (*w/o OS*), syntactic structure relevance (*w/o SR*), and aspect-sentiment semantics (*w/o AS*) in turn. The ablation results are reported in Table 5. We can observe that removing any of the three perspectives generally causes performance degradation, which indicates that each helps retrieve high-quality demonstrations. Among these, the syntactic structure relevance has the most moderate effect on performance. This is within our expectation since how to encode syntactic structures needs to be deeply explored rather than roughly encoding POS sequences. Another interesting observation is that task instructions are more important for performance improvement (see *w/o TI*). This indicates that the simplest way of unleashing LLM power is to describe tasks to it.

4.5. Comparison with Other Solutions

To further show the effectiveness of the proposed framework, we compare it with some representative solutions for dealing with ABSA under low resources. Here, we select two few-shot ABSA solutions (FSGPT (Hosseini-Asl et al., 2022) and FSABSA (Wang et al., 2023c)) and three data augmentation solutions. Table 4 reports the experimental results. Our framework performs better than all solutions, achieving the biggest improvement of 22.29% in F1 scores. This suggests that if useful in-context examples are picked judiciously, solving few-shot ABSA with the ICL paradigm is a more promising solution.

4.6. Discussions

Impact of the Number of In-Context Examples. To investigate the impact of the number of in-context examples k, we pick four retrieval methods and vary k from $1, 2, 4, 8, 12$ to 16. Figure 2 depicts the experimental results. We can draw two conclusions: (1) Our framework consistently yields superior results across varying amounts of in-context examples. In



(a) On the Rest14 dataset.

(b) On the Clothing dataset.

Figure 2: Impact of in-context example number.

	Rest14			Clothing		
	P	R	F1	P	R	F1
LLaMA-7B						
Random	47.73	47.18	47.46	30.82	31.64	31.23
BM25	52.94	55.40	54.14	43.03	46.01	44.47
EPR	48.77	49.77	49.27	36.65	43.08	39.60
Ours	54.86	57.90	**56.34**	47.13	50.26	**48.64**
LLaMA-13B						
Random	60.02	50.84	55.05	37.86	34.04	35.85
BM25	62.45	58.71	60.52	52.99	49.46	51.16
EPR	60.20	53.52	56.66	48.22	43.35	45.65
Ours	63.93	61.30	**62.59**	56.69	52.92	**54.74**
LLaMA-30B						
Random	61.91	52.01	56.94	40.83	39.09	39.94
BM25	64.93	60.23	62.49	48.55	49.20	48.87
EPR	64.03	56.47	60.01	45.65	43.35	44.47
Ours	69.33	63.44	**66.26**	54.41	52.39	**53.38**

Table 6: Comparative performance (%) of LLaMA with different sizes when using 4-shot examples.

addition, on the Clothing dataset, the proposed framework using 2 in-context examples still outperforms the Random method using 8 ones. This shows that the quality of retrieved examples is more important than their quantity. (2) The overall performance generally improves as we increase the number of retrieved examples, except for after 8 examples on Rest14. This indicates that more in-context examples may bring more knowledge to better guide LLMs for inference.

Performance of Using LLaMA with Different Sizes. In the above experiments, we use LLaMA-13B as LLM for inference. Thus, a question naturally arises about whether utilizing a larger-scale LLaMA will improve performance. To answer this question, we explore the performance of using LLaMA with different sizes. Here, we vary the size of LLaMA from 7B to 30B. Table 6 presents the overall results. We discover that increasing size tends to exhibit a substantial performance boost. We speculate that this is because larger-scale models learn more valuable semantics and make better use of ICL (Wei et al., 2023). Moreover, our framework improves consistently over other retrieval methods regardless of model sizes.

	Rest14			Clothing		
	P	R	F1	P	R	F1
GPT-J-6B						
Random	40.38	33.60	36.68	27.29	26.86	27.07
BM25	42.10	40.03	41.04	34.73	32.97	33.83
EPR	44.42	37.35	40.58	24.86	23.93	24.39
Ours	47.76	45.84	**46.78**	35.40	36.43	**35.91**
Baichuan 2-7B						
Random	40.96	36.46	38.58	22.28	21.27	21.76
BM25	49.49	48.07	48.77	38.52	36.17	**37.31**
EPR	44.20	40.21	42.11	29.47	27.12	28.25
Ours	55.20	51.20	**53.12**	37.98	36.17	37.05
ChatGPT						
Random	58.76	65.59	61.99	32.45	48.93	39.02
BM25	61.65	70.42	65.74	42.40	57.18	48.69
EPR	62.85	69.25	**65.90**	38.31	53.19	44.54
Ours	62.21	68.27	65.10	43.39	61.17	**50.77**

Table 7: Comparative performance (%) of different LLMs under 2-shot demonstrations.

	Rest14			Clothing		
	P	R	F1	P	R	F1
TI-1+DF-1	63.93	61.30	**62.59**	56.69	52.92	54.74
TI-1+DF-2	63.32	59.24	61.21	55.83	53.45	54.61
TI-2+DF-1	63.85	58.89	61.27	55.84	52.12	53.92
TI-2+DF-2	61.99	58.89	60.40	58.52	54.78	**56.59**

Table 8: Results (%) under different prompt templates. TI and DF are shorthand for task instructions and demonstration formats, respectively.

Effectiveness on Different LLMs. To understand the effectiveness of our framework more comprehensively, we conduct comparative experiments on LLMs of different categories. Here, we choose GPT-J-6B (Wang and Komatsuzaki, 2021), Baichuan 2-7B (Baichuan, 2023), and ChatGPT (gpt-3.5-turbo, 175B)[11] as LLMs for in-context inference. From Table 7, we can conclude that: (1) Our framework generally achieves the best results, suggesting that retrieving in-examples from three perspectives proposed in the paper can facilitate different LLMs to address few-shot ABSA. (2) Random has the worst results overall. However, surprising results can be attained when exploiting ChatGPT, indicating that examples have a slight impact on performance under super-scale LLMs.

Impact of Prompt Template on Performance. In the prompt template, task instructions (TI) and demonstration formats (DF) can affect the performance of ICL (Dong et al., 2022). To understand this point intuitively, we utilize another TI and DF

[11] https://platform.openai.com/docs/models/gpt-3-5

here and perform combination experiments.[12] For simplicity, we abbreviate them as TI-2 and DF-2, respectively, and TI and DF in Figure 1 as TI-1 and DF-1, respectively. Table 8 presents the experimental results. We can find that the extent to which templates affect performance depends on the dataset. In addition, the performance fluctuates more when TI is changed. This is because TI provides important and explicit guidance on the tasks that LLM is required to perform.

5. Conclusion

This paper leverages the ICL paradigm to address few-shot ABSA. To select useful in-context examples for each test input from the candidate pool, which can offer informative clues for LLM to predict aspect-sentiment pairs, we consider three perspectives to retrieve examples, *i.e.*, overall semantics, syntactic structure relevance, and aspect-sentiment semantic. To examine the effectiveness of this consideration, we conduct extensive experiments on four ABSA datasets. The results show that our retrieval framework can outperform some strong competitors. In addition, we perform some in-depth analyses on in-context examples and present some insights between performance and factors, which can shed light on future work.

6. Acknowledgements

This work was supported in part by the National Natural Science Foundation of China under Grants 62176076, the Natural Science Foundation of Guangdong under Grant 2023A1515012922, Shenzhen Foundational Research Funding under Grant JCYJ20220818102415032, The Major Key Project of PCL under Project PCL2023A09, and Guangdong Provincial Key Laboratory of Novel Security Intelligence Technologies under Grant 2022B1212010005k.

7. References

Sweta Agrawal, Chunting Zhou, Mike Lewis, Luke Zettlemoyer, and Marjan Ghazvininejad. 2023. In-context examples selection for machine translation. In *Findings of ACL*, pages 8857–8873.

Baichuan. 2023. Baichuan 2: Open large-scale language models. *arXiv preprint arXiv:2309.10305*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in NeurIPS*, 33:1877–1901.

Hongjie Cai, Nan Song, Zengzhi Wang, Qiming Xie, Qiankun Zhao, Ke Li, Siwei Wu, Shijie Liu, Jianfei Yu, and Rui Xia. 2023. Memd-absa: A multi-element multi-domain dataset for aspect-based sentiment analysis. *arXiv preprint arXiv:2306.16956*.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR.

Ting Chen, Yizhou Sun, Yue Shi, and Liangjie Hong. 2017. On sampling strategies for neural network-based collaborative filtering. In *Proceedings of ACM KDD*, pages 767–776.

Junqi Dai, Hang Yan, Tianxiang Sun, Pengfei Liu, and Xipeng Qiu. 2021. Does syntax matter? a strong baseline for aspect-based sentiment analysis with roberta. In *Proceedings of NAACL*, pages 1816–1829.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL*, pages 4171–4186.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, and Zhifang Sui. 2022. A survey for in-context learning. *arXiv preprint arXiv:2301.00234*.

Shivam Garg, Dimitris Tsipras, Percy S Liang, and Gregory Valiant. 2022. What can transformers learn in-context? a case study of simple function classes. *Advances in NeurIPS*, 35:30583–30598.

Hila Gonen, Srini Iyer, Terra Blevins, Noah A Smith, and Luke Zettlemoyer. 2022. Demystifying prompts in language models via perplexity estimation. *arXiv preprint arXiv:2212.04037*.

Kai He, Rui Mao, Tieliang Gong, Chen Li, and Erik Cambria. 2022. Meta-based self-training and re-weighting for aspect-based sentiment analysis. *IEEE Transactions on Affective Computing*.

Or Honovich, Uri Shaham, Samuel R Bowman, and Omer Levy. 2023. Instruction induction: From few examples to natural language task descriptions. In *Proceedings of ACL*, pages 1935–1952.

[12] TI: *Please perform Unified Aspect-Based Sentiment Analysis task. Given a review, tag all [aspect, sentiment] pairs.* DF: *Input:* $text$ *Output:* $tuple\ list$

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

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of ACL*, pages 7871–7880.

Jia Li, Yuyuan Zhao, Zhi Jin, Ge Li, Tao Shen, Zhengwei Tao, and Chongyang Tao. 2022. Sk2: Integrating implicit sentiment knowledge and explicit syntax knowledge for aspect-based sentiment analysis. In *Proceedings of ACM CIKM*, pages 1114–1123.

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*.

Xin Li, Lidong Bing, Piji Li, and Wai Lam. 2019a. A unified model for opinion target extraction and target sentiment prediction. In *Proceedings of AAAI*, pages 6714–6721.

Xin Li, Lidong Bing, Wenxuan Zhang, and Wai Lam. 2019b. Exploiting bert for end-to-end aspect-based sentiment analysis. In *Proceedings of W-NUT*, pages 34–41.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, William B Dolan, Lawrence Carin, and Weizhu Chen. 2022. What makes good in-context examples for gpt-3? In *arXiv preprint arXiv:2101.06804*, pages 100–114.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2022. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In *Proceedings of ACL*, pages 8086–8098.

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

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Rethinking the role of demonstrations: What makes in-context learning work? In *Proceedings of EMNLP*, pages 11048–11064.

Shinhyeok Oh, Dongyub Lee, Taesun Whang, Il-Nam Park, Seo Gaeun, EungGyun Kim, and Harksoo Kim. 2021. Deep context-and relation-aware learning for aspect-based sentiment analysis. In *Proceedings of ACL (Volume 2: Short Papers)*, pages 495–503.

Maria Pontiki, Dimitrios Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. 2014. Semeval-2014 task 4: Aspect based sentiment analysis. In *Proceedings of SemEval*, pages 27–35.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI Technical Report*.

Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends in Information Retrieval*, 3(4):333–389.

Ohad Rubin, Jonathan Herzig, and Jonathan Berant. 2022. Learning to retrieve prompts for in-context learning. In *Proceedings of NAACL*, pages 2655–2671.

Hongjin Su, Jungo Kasai, Yizhong Wang, Yushi Hu, Mari Ostendorf, Wen-tau Yih, Noah A Smith, Luke Zettlemoyer, Tao Yu, et al. 2022. One embedder, any task: Instruction-finetuned text embeddings. *arXiv preprint arXiv:2212.09741*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Ben Wang and Aran Komatsuzaki. 2021. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. `https://github.com/kingoflolz/mesh-transformer-jax`.

Kai Wang, Weizhou Shen, Yunyi Yang, Xiaojun Quan, and Rui Wang. 2020. Relational graph attention network for aspect-based sentiment analysis. In *Proceedings of ACL*, pages 3229–3238.

Lean Wang, Lei Li, Damai Dai, Deli Chen, Hao Zhou, Fandong Meng, Jie Zhou, and Xu Sun. 2023a. Label words are anchors: An information flow perspective for understanding in-context learning. *arXiv preprint arXiv:2305.14160*.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023b. Self-instruct: Aligning language model with self generated instructions. In *Proceedings of ACL*, pages 13484–13508.

Zengzhi Wang, Qiming Xie, and Rui Xia. 2023c. A simple yet effective framework for few-shot aspect-based sentiment analysis. In *Proceedings of ACM SIGIR*, pages 1765–1770.

Jerry Wei, Jason Wei, Yi Tay, Dustin Tran, Albert Webson, Yifeng Lu, Xinyun Chen, Hanxiao Liu, Da Huang, Denny Zhou, et al. 2023. Larger language models do in-context learning differently. *arXiv preprint arXiv:2303.03846*.

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. 2021. An explanation of in-context learning as implicit bayesian inference. In *ICLR*.

Hu Xu, Bing Liu, Lei Shu, and S Yu Philip. 2018. Double embeddings and cnn-based sequence labeling for aspect extraction. In *Proceedings of ACL (Volume 2: Short Papers)*, pages 592–598.

Hu Xu, Bing Liu, Lei Shu, and S Yu Philip. 2019. Bert post-training for review reading comprehension and aspect-based sentiment analysis. In *Proceedings of NAACL*, pages 2324–2335.

Wei Xue and Tao Li. 2018. Aspect based sentiment analysis with gated convolutional networks. In *Proceedings of ACL*, pages 2514–2523.

Michihiro Yasunaga, Xinyun Chen, Yujia Li, Panupong Pasupat, Jure Leskovec, Percy Liang, Ed H Chi, and Denny Zhou. 2023. Large language models as analogical reasoners. *arXiv preprint arXiv:2310.01714*.

Yiming Zhang, Shi Feng, and Chenhao Tan. 2022. Active example selection for in-context learning. In *Proceedings of EMNLP*, pages 9134–9148.