SimRP: Syntactic and Semantic Similarity Retrieval Prompting Enhances Aspect Sentiment Quad Prediction

Zhongquan Jian[1,2], Yanhao Chen[3], Jiajian Li[1], Shaopan Wang[1,2], Xiangjian Zeng[4], Junfeng Yao[3,2,1,6], Xinying An[5,*], Qingqiang Wu[3,2,1,6,*]

[1]Institute of Artificial Intelligence, Xiamen University, Xiamen, China
[2]School of Informatics, Xiamen University, Xiamen, China
[3]School of Film, Xiamen University, Xiamen, China
[4]School of Journalism and Communication, Xiamen University, Xiamen, China
[5]Institute of Medical Information, Chinese Academy of Medical Sciences, Beijing, China
[6]Xiamen Key Laboratory of Intelligent Storage and Computing, School of Informatics, Xiamen University
{jianzq,cyhao,shaopanw,lijiajian}@stu.xmu.edu.cn, {xjzeng,yao0010}@xmu.edu.cn,
an.xinying@imicams.ac.cn, wuqq@xmu.edu.cn

Abstract

Aspect Sentiment Quad Prediction (ASQP) is the most complex subtask of Aspect-based Sentiment Analysis (ABSA), aiming to predict all sentiment quadruples within the given sentence. Due to the complexity of sentence syntaxes and the diversity of sentiment expressions, generative methods gradually become the mainstream approach in ASQP. However, existing generative models are constrained in the effectiveness of demonstrations. Semantically similar demonstrations help in judging sentiment categories and polarities but may confuse the model in recognizing aspect and opinion terms, which are more related to sentence syntaxes. To this end, we first develop Syn2Vec, a method for calculating syntactic vectors to support the retrieval of syntactically similar demonstrations. Then, we propose Syntactic and Semantic **Sim**ilarity **R**etrieval **P**rompting (**SimRP**) to construct effective prompts by retrieving the most related demonstrations that are syntactically and semantically similar. With these related demonstrations, pre-trained generative models, especially Large Language Models (LLMs), can fully release their potential to recognize sentiment quadruples. Extensive experiments in Supervised Fine-Tuning (SFT) and In-context Learning (ICL) paradigms demonstrate the effectiveness of SimRP. Furthermore, we find that LLMs' capabilities in ASQP are severely underestimated by biased data annotations and the exact matching metric. We propose a novel constituent subtree-based fuzzy metric for more accurate and rational quadruple recognition.

Code — https://github.com/jian-projects/simrp

Introduction

Aspect-based Sentiment Analysis (ABSA) is a crucial task in Natural Language Processing (NLP) and real-world applications, aiming to enable machines to understand human concerns and opinions expressed in text (Liu et al. 2020; Brauwers and Frasincar 2022). ABSA has consistently attracted increasing attention in the Pre-trained Language Model (PLM) era, as well as in the Large Language Model (LLM) era, owing to its wide range of applications, including product reviews, social media analysis, and customer feedback analysis (Zhang et al. 2024b). The most representative and challenging subtask in ABSA is Aspect Sentiment Quad Prediction (ASQP), which aims to predict all sentiment quadruples within the given sentence (Zhang et al. 2023, 2024a), each consisting of aspect term, opinion term, aspect category, and sentiment polarity, denoted as (a, o, c, s). As depicted in Figure 1, given a review sentence "*In the end you end up with a fair tab and nothing but a great time !!!*", ASQP requires predicting two containing sentiment quadruples: (*Null, fair, restaurant price, positive*) and (*Null, great, restaurant general, positive*), where *Null* represents an implicit aspect term in this sentence.

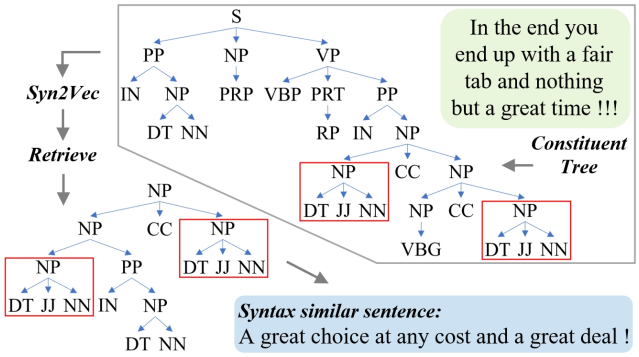


Figure 1: Retrieving sentences with similar syntaxes.

Due to the complexity of ASQP in recognizing all sentiment quadruples, generative models, like T5 (Raffel et al. 2020) and GPT (Brown et al. 2020), with seq2seq paradigm have been mainstream approaches and achieved promising results (Peper and Wang 2022). Initially, studies show that generation objectives greatly influenced the extraction of sentiment quadruples. Representatively, GAS (Zhang et al. 2021b) explored the influences of annotation-style and extraction-style generation objectives, demonstrating the superiority of the extraction-style mode in ASQP. Another

*Corresponding authors.

method is to paraphrase sentiment quadruples into natural sentences (Zhang et al. 2021a; Hu et al. 2022), which follows attributes of generative models, therefore achieving outstanding performance. Subsequently, MvP (Gou, Guo, and Yang 2023) further explored the impact of quadruple generation order on model performance and proposed voting on the most reasonable quadruples generated in different orders, significantly enhancing predictive capability.

With the rise of In-context Learning (ICL) (Dong et al. 2022; Xu et al. 2024), researchers find that model input also plays a crucial role in activating the reasoning capabilities of PLM and LLM (Zhao et al. 2023). Furthermore, adding detailed instructions and demonstrations to task prompts can significantly activate the model performance on downstream tasks, whether in zero-shot inference or supervised training (Yang et al. 2024). For instance, Sun et al. (2024) showed that integrating enriched domain insights from the knowledge base enhances the input, improving model performance. Recently, Instruct-ABSA (Scaria et al. 2024) proposed an instruction learning paradigm that can extend the input with several positive, negative, and neutral demonstrations, and instruction tune the backbone model to address the ABSA subtasks, yielding significant performance improvements. Consequently, the following research has begun to investigate optimal demonstration selection. Numerous studies (Liu et al. 2022; Min et al. 2022) found that choosing demonstrations semantically and label-wise closer to the actual input is more effective.

In the deep learning era, semantic similarity retrieval has been successfully applied to various NLP and computer vision tasks, demonstrating superior performance in acquiring valuable information relevant to the target task. For the ASQP task, semantically similar demonstrations provide similar sentiment insights to aid in judging sentiment categories and polarities (Zhang et al. 2023). However, recognizing aspect and opinion terms is more closely tied to sentence syntax, emphasizing the need for syntactically similar demonstrations, as they offer explicit syntactic cues for identifying these terms. As the example shown in Figure 1, the target sentence has the same constituent subtree (CsT) as the retrieval sentence, allowing quadruples in the target sentence to be intuitively recognized by referencing the quadruple recognition pattern of syntactically similar demonstrations. Therefore, with the syntactically and semantically similar demonstrations, the generative models, especially LLMs, can fully release their potential in recognizing sentiment quadruples, and thus improve the performance of ASQP.

To this end, we first develop Syn2Vec, a syntactic vector calculation method that allows representing sentence syntaxes as vectors to support the retrieval of syntactically similar demonstrations. Syn2Vec is built upon the vocabulary of CsTs, meaning that syntactic vectors are word vectors derived from CsTs present in sentences. Thus, syntactic similarity between sentences can be measured by comparing their syntactic vectors, for example, using Mean Square Error (MSE). Building on Syn2Vec, we introduce Syntactic and Semantic Similarity Retrieval Prompting (**SimRP**), which constructs effective prompts by retrieving demonstrations that are both syntactically and semantically similar. Specifically, we first use Syn2Vec to retrieve syntactically similar demonstrations from the training set, then use Sentence-BERT model (Reimers and Gurevych 2019) to select the most semantically similar ones from this subset. To verify the effectiveness of the proposed SimRP, we conduct extensive experiments in Supervised Fine-Tuning (SFT) and ICL paradigms on two public ASQP datasets, where the ICL paradigm is implemented under zero-shot and few-shot settings based on GPT-4 and Llama-3 models. In addition, we propose a novel CsT-based fuzzy metric to evaluate the accuracy of quadruple recognition, which can better reflect the potential of LLMs in ASQP. In summary, the main contributions of this work are as follows:

- We propose SimRP to address the ASQP task by constructing effective prompts with syntactic and semantically similar demonstrations, which can effectively activate the reasoning capabilities of PLMs and LLMs.
- We developed a novel Syn2Vec to represent sentence syntaxes as vectors to enable the retrieval of syntactically similar demonstrations, after which Sentence-BERT is used to select the most semantically similar ones.
- Experiments in SFT and ICL paradigms demonstrate the effectiveness of the proposed SimRP, achieving state-of-the-art results on two public ASQP datasets. The proposed CsT-based fuzzy evaluation metric verifies the potential of LLMs in ASQP.

Methodology

Definition

Given a review sentence x, ASQP aims to identify all aspect-based sentiment quadruples $\{(a, o, c, s)\}$, where each corresponds to the aspect term, opinion term, aspect category, and sentiment polarity, respectively. Generally, the aspect term a and opinion term o are typically text spans in x, and a can also be represented by a specific tag *Null* if it is not explicitly mentioned in x, *i.e.,* $a \in U_x \cup$ *Null* and $o \in U_x$, where U_x is the set of possible continuous spans of x. The aspect category c falls into a predefined category set U_c, and the sentiment polarity s belongs to one of the sentiment class $U_s \in \{\textit{Positive}, \textit{Neutral}, \textit{Negative}\}$.

Motivation

With the emergence of PLMs and LLMs, generative models have been widely applied to NLP tasks, including the ASQP task. The goal of generative models is to maximize the likelihood of the next token prediction, formulated as:

$$\hat{y}_t = p(y_t \mid prompt_x, y_{<t}) \tag{1}$$

where $prompt_x$ denotes the input sequence that is usually composed of task instruction, demonstrations, and the input sentence x. y denotes the target sequence, and $y_{<t}$ represents the previous tokens of y before the t-th token. The templates of $prompt_x$ and y for the ASQP task are introduced in the section of *Implementation Details*. In our work, we focus on the construction of $prompt_x$, which plays a crucial role in activating the reasoning capabilities of generative models, especially for LLMs. More specifically, we attempt to seek suitable types of demonstrations for the ASQP task.

Algorithm 1: Syn2Vec

Initial vocabulary $V = \varnothing$. ▷ initialize the vocabulary V
for Each sentence x in the Training set T **do**
  x_q denotes the arbitrary word of a or o in x.
  Parse x to obtain its CsTs: *spaCy*(x).
  for cst in **set** (*spaCy*(x)) **do**
    if x_q in cst & cst not in V **then**
      $F(cst) = 0$. ▷ initialize the frequency of cst
      $V = V \cup \{cst\}$. ▷ add cst to V
    end if
    Increment $F(cst)$. ▷ count the frequency of cst
  end for
end for
Sort V by F in descending order.
Calculate the IDF value of each cst in V: *IDF*(cst).
return V.

Retrieval Augmentation Generation (RAG) (Ram et al. 2023; Zhao et al. 2024) is a widely used method for enhancing the effectiveness of generative models by retrieving useful knowledge from external sources or training data (Wang et al. 2022). In the ASQP task, aspect and opinion terms are closely related to sentence syntax. Existing semantic-based retrieval methods (Karpukhin et al. 2020) are not well-suited for retrieving demonstrations with similar syntaxes. To address this gap, we introduce Syn2Vec, a method that extracts syntactic information by representing sentence syntax as vectors. Using Syn2Vec, syntactically similar demonstrations can be retrieved from the training set, similar to how semantically similar demonstrations are retrieved.

Syn2Vec

Dense retrieval (Karpukhin et al. 2020; Zhao et al. 2024) excels in retrieving similar text from knowledge sources. In this section, we introduce Syn2Vec, a method that encodes syntactic information in sentences as vectors, enabling dense retrieval and then obtaining syntactically similar demonstrations. In a nutshell, Syn2Vec is a function that maps a sentence x to its syntactic vector v_x, represented as:

$$v_x = \text{Syn2Vec}(x) \tag{2}$$

where $v_x \in \mathbb{R}^d$ is a sparse vector with d as the dimension.

In NLP, sentence syntax is often represented by its constituent tree (CT), extracted using tools like spaCy or Stanford NLP. Building on the accurate parsing of CTs, we developed Syn2Vec to represent sentence syntaxes as word vectors based on the vocabulary of CsTs. As outlined in Algorithm 1, the vocabulary construction process is as follows:
(1) For a sentence x, we first utilize the spaCy tool to parse its CT, then extract all CsTs using regex matching. This process is represented as *spaCy*(x), for simplicity. As the example illustrated in Figure 1, a non-leaf node and its descendant nodes form a CsT, *e.g., (NP DT JJ NN)*. (***line 4***)
(2) We compile all CsTs from training set sentences into the CsT set. For ASQP, we filter CsTs to retain only those aiding aspect-opinion pair extraction, forming a refined CsT vocabulary V. This is done simply by checking whether an aspect or opinion term appears within each CsT. (***lines 2-12***)
(3) Generally, CsT frequencies indicate how commonly syntax patterns extract aspect-opinion pairs, *i.e.,* a sentence with a frequent CsT is more likely to yield these pairs. Hence, we sort CsTs by frequency and calculate the Inverse Document Frequency (IDF) of each CsT as its weight. (***lines 13-14***)

$$\textit{IDF}(cst) = \log\left(N/F(cst)\right) \tag{3}$$

where N is the total number of sentences, and $F(cst)$ is the number of sentences containing cst. The intuition behind using IDF values to weight CsTs is that less frequent CsTs are more informative for distinguishing sentence syntax.

Finally, the word vector of v_x is represented as follows:

$$v_x = \left\{\textit{IDF}(V_i) * \textit{Count}\left(\textit{spaCy}(x), V_i\right)\right\}_{i=1}^{d} \tag{4}$$

where V_i is the i-th CsT in V, and *Count*() is use to count the frequency of V_i in *spaCy*(x).

SimRP

Our goal is to retrieve the most effective demonstrations, offering both syntactic and semantic references to enlighten the generative model to generate intact and correct sentiment quadruples. With the developed Syn2Vec, sentences can be represented as syntactic vectors. Due to the sparsity of syntactic vectors, we employ MSE to measure the syntactic similarity between sentences, and retrieve syntactically similar demonstrations for x from the training set T.

$$\text{SynSim}(x, T) = \arg\max_{x_i \in T} \|x^{syn} - x_i^{syn}\|_2 \tag{5}$$

where $x^{syn} = \text{Syn2Vec}(x)$ represents the syntactic vector of x. $\| * \|_2$ denotes the calculation of Euclidean distance, known as the L2 norm. We pick top-k similar samples as syntactically similar demonstrations, denoted as T^{Syn}. Subsequently, we rank demonstrations in T^{Syn} according to their semantic similarity to the input sentence x.

$$\text{SemSim}(x, T^{\text{Syn}}) = \arg\max_{x_i \in T^{\text{Syn}}} \frac{x^{sem} \cdot x_i^{sem}}{\|x^{sem}\|\|x_i^{sem}\|} \tag{6}$$

where $x^{sem} = \text{Sentence-BERT}(x)$ represents the semantic vector of x. We further select top-k' demonstrations $T_{\text{Sem}}^{\text{Syn}}$ for prompt construction. Hence, the task prompt is achieved:

$$prompt_x = \{\textit{Instruction}, T_{\text{Sem}}^{\text{Syn}}, x\} \tag{7}$$

where *Instruction* refers to the task instruction tailored for the downstream task.

Experimental Setup

Datasets

Experiments are carried out on two widely used ASQP datasets, *i.e.,* Rest15 and Rest16, with their statistics shown in Table 1. These datasets are initially constructed based on the SemEval task (Pontiki et al. 2015, 2016) and have undergone multiple annotations (Peng et al. 2020a; Wan et al. 2020). Zhang et al. (2021a) aligned these datasets and served as the standard datasets for the ASQP task finally. Each sentence contains multiple sentiment quadruples, and a predicted quadruple is correct only if all its elements exactly match the gold quadruple. Following previous works (Zhang et al. 2021b; Gou, Guo, and Yang 2023), the F1 score is used as the main evaluation metric in our experiments.

Dataset	Rest15			Rest16		
	Train	Valid	Test	Train	Valid	Test
N	834	209	537	1264	316	544
Positive	1005	252	453	1369	341	583
Neutral	34	14	37	62	23	40
Negative	315	81	305	558	143	176
Total	1354	347	795	1989	507	799

Table 1: Dataset statistics for Rest15 and Rest16. N denotes the number of sentences. *Positive*, *Neutral*, and *Negative* represent the number of quadruples for each sentiment.

Implementation Details

We evaluate the proposed SimRP in two paradigms: SFT and ICL. In the SFT paradigm, we fine-tune the PLM using labeled data, while in ICL, we leverage LLMs to perform the ASQP task in zero-shot and few-shot settings. The dimension of the syntactic vector d is set to 2^8 in our experiments.

Supervised Fine-Tuning (SFT) We employ the T5-large[1] pre-trained model as the backbone. Greedy search decoding is used by default. During the model training, the max epoch number is set to 10, and the batch size is set to 4 for all experiments. AdamW with an initial learning rate of 8e-5 is used as the optimizer, and linear scheduling is applied to adjust the learning rate. Setting $k = 10$ in the data preparation stage, *i.e.,* we retrieve the top 10 syntactically similar demonstrations from the training set for each sentence. The numbers of concatenated demonstrations k' are set to 5 and 1 for rest15 and rest16, respectively. To increase the diversity of training samples, we randomly pick k' out of k demonstrations during the training stage. While in the testing stage, we concatenated the top k' similar demonstrations to construct the prompt. Following the successful application of Gou, Guo, and Yang (2023), the target sequence is linearized as $y = [A]\ a\ [O]\ o\ [C]\ c\ [S]\ s\ [SSEP]$..., where $[A]$, $[O]$, $[C]$, and $[S]$ are special tokens for differentiating aspect, opinion, category, and sentiment, respectively, and multiple quadruples are concatenated using the special token $[SSEP]$. SFT directly optimizes model parameters to map the input sentence to the target sequence. Therefore, we simply concatenate the input sentence with demonstrations to construct the prompt, omitting any instructional description. Some examples of prompts are shown in the Appendix. We employ the standard cross entropy loss to optimize the parameters of the generative model:

$$\mathcal{L} = \sum_{t=1}^{n} y_t \log(\hat{y}_t) \tag{8}$$

where n is the length of the target sequence y, y_t is a one-hot vector representing the ground truth of t-th word in y, and $\hat{y}_t$ denotes the corresponding predicted probability distribution.

Recent advanced ASQP models are compared to evaluate our proposed SimRP, including **GAS** (Zhang et al. 2021b), **Paraphrase** (Zhang et al. 2021a), **DLO/ILO** (Hu et al. 2022), **MVP** (Gou, Guo, and Yang 2023), **GenDa** (Wang et al. 2023a), **MUL** (Hu et al. 2023), **OTCL** (Li et al. 2024), and **IVLS** (Nie et al. 2024), where OTCL and IVLS are non-generative methods, and the others are generative methods.

In-context Learning (ICL) ICL is a crucial technique for LLMs to tackle complex tasks based on a few annotated demonstrations without additional training or gradient updates. Our proposed SimRP emphasizes demonstration selection for ICL, making it particularly well-suited for using LLMs to tackle the ASQP task. Following the instruction template in Zhang et al. (2024b), we construct the prompt for LLMs by the proposed SimRP. Some cases of prompts are shown in the Appendix due to space limitations. Experiments are conducted based on GPT-4o[2] (*i.e.,* GPT-4o-mini and GPT-4o) and Llama-3.1[3] (*i.e.,* Llama-3.1-8B and Llama-3.1-70B) models under zero-shot and few-shot settings. In the ICL paradigm, k is also set to 10, and the most similar demonstrations are used to construct the prompt.

Compared methods including: **ThoR** (Fei et al. 2023) designed a three-step prompting principle to induce the implicit aspect, opinion, and sentiment polarity step by step. **LLMs for SA** (Zhang et al. 2024b) is a comprehensive study of sentiment analysis tasks using LLMs, where samples are randomly added with diverse demonstrations.

Results and Analysis in the SFT Paradigm

Main Results

We execute experiments three times with fixed seeds of [2024, 2025, 2026], and report the mean values in Table 2. The best results are marked in bold and the second-best underlined. Overall, our approach greatly outperforms existing methods, with an average F1 score improvement of 2.20%.

Additionally, we have the following observations. (1) ASQP remains challenging for generative models, with conversational non-generative models (*e.g.,* OTCL and IVLS) generally outperforming them, particularly IVLS, which performs best on both datasets. (2) Generative models possess great potential for improvement, as evidenced by the substantial performance gains achieved by MvP, which optimize the output views of different quadruple-generation orders. (3) Based on the achievements of existing generative methods, SimRP focuses on the optimization of input sequences, providing more syntactically and semantically similar demonstrations for the input, which significantly activates the reasoning capabilities of generative models. (4) The performance of generative models for ASQP is underestimated, the more flexible and comprehensive CsT-based fuzzy metric is better suited for evaluation.

Impacts of Demonstration Number

In this section, we investigate the impacts of integrated demonstrations in the SFT paradigm, experimental results are illustrated in Figure 2. It can be found that demonstrations play a crucial role in enhancing the performance of

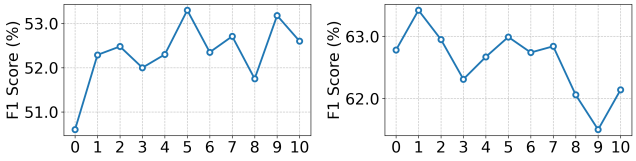
[1]https://huggingface.co/google-t5/t5-large

[2]https://chatgpt.com/

[3]https://llama.meta.com/

Methods	Rest15			Rest16			Avg.(F1)
	Pre	Rec	F1	Pre	Rec	F1	
GAS† (Zhang et al. 2021b)	45.31	46.70	45.98	54.54	57.62	56.04	51.01
Paraphrase (Zhang et al. 2021a)	46.16	47.72	46.93	56.63	59.30	57.93	52.43
DLO (Hu et al. 2022)	47.08	49.33	48.18	57.92	61.80	59.79	53.99
ILO (Hu et al. 2022)	47.08	50.38	49.05	57.58	61.17	59.32	54.19
MvP (Gou, Guo, and Yang 2023)	-	-	51.04	-	-	60.39	55.72
GenDA (Wang et al. 2023a)	49.74	50.29	50.01	60.08	61.70	60.88	55.45
MUL (Hu et al. 2023)	49.12	50.39	49.75	59.24	61.75	60.47	55.11
OTCL (Li et al. 2024)	47.86	50.77	49.27	58.31	62.02	60.11	54.69
IVLS (Nie et al. 2024)	54.46	48.53	<u>51.28</u>	62.69	59.75	<u>61.04</u>	<u>56.16</u>
SimRP (Ours)	53.12	53.50	**53.30**$_{\uparrow 2.02}$	62.74	64.12	**63.42**$_{\uparrow 2.38}$	**58.36**$_{\uparrow 2.20}$
using CsT-based fuzzy metric	*56.97*	*57.61*	*57.29*	*68.37*	*68.46*	*68.42*	*63.00*

Table 2: Comparison results in terms of precision (Pre, %), recall (Rec, %) and F1 score (F1, %), with the best in bold and the second best underlined. † denotes the results quoted from Wang et al. (2023a), the others are cited from their original papers.



(a) Rest15 (b) Rest16

Figure 2: Impact of demonstration number k'.

Methods	Rest15	Rest16	Avg.(F1)
SimRP	53.30	63.42	58.36
- [Syn.]	52.58$_{\downarrow 0.72}$	62.97$_{\downarrow 0.45}$	57.78$_{\downarrow 0.58}$
- [Sem.]	52.44$_{\downarrow 0.86}$	62.83$_{\downarrow 0.59}$	57.64$_{\downarrow 0.72}$
- [Syn., Sem.]	51.00$_{\downarrow 2.30}$	62.18$_{\downarrow 1.24}$	56.59$_{\downarrow 1.77}$

Table 3: Ablation studies in the SFT paradigm.

generative models, especially on Rest15. However, more demonstrations seem useless, as the number of demonstrations increases, the results on Rest15 no longer enhanced, while on Rest16 significantly declined. The potential reason is that retrieved demonstrations have been exposed to the model during training, and the model has learned the mapping from inputs to outputs after fine-tuning. In the inference stage, the model may be confused by more retrieved demonstrations, leading to a performance decline.

Ablation Study

Furthermore, we conducted ablation studies to evaluate the roles of various types of retrieval demonstrations. Experimental results are tabulated in Table 3, where "-[Syn.]" denotes the removal of SynSim, namely using semantically similar demonstrations only. Similarly, "-[Sem.]" denotes the removal of SemSim, namely using syntactically similar demonstrations only. "-[Syn., Sem.]" means that demonstrations are randomly selected from the training set.

From Table 3, both syntactically and semantically similar demonstrations contribute to performance improvement, with semantically similar ones playing a more crucial role, as removing them causes a greater decline.

Results and Analysis in the ICL Paradigm

Main Results

Table 4 reports the experimental results of using LLMs to address the ASQP task. Experiments are executed under zero-shot and few-shot settings. We can observe that: (1) Longitudinally, as the backbone model size increases, LLM performance in ASQP improves significantly, *e.g.,* Llama-3.1-70B and GPT-4o outperform their lower scaled versions greatly. GPT-4o achieves the best results among all LLMs under different settings. (2) Transversely, as the number of demonstrations increases, the accuracy of quadruple generation improves across all models, except for lower scaled models in "LLMs for SA", whose performance tends to decrease with a single demonstration. (3) Comparison among these three methods, SimRP surpasses the other methods in all settings, especially equipped large-scale models as the backbone. Demonstrating the superiority of syntactically and semantically demonstrations for the ASQP task.

Ablation Study

Similarly, we execute ablation studies to investigate the roles of different types of demonstrations in the ICL paradigm. Experimental results of GPT-4o are tabulated in Table 5, where "Syn." and "Sem." share the same meanings as those in Table 3. It's noted that we have the same observations as in the SFT paradigm, where semantically similar demonstrations are more important than syntactically similar ones. Syntactically retrieval demonstrations also play a crucial role in enhancing the performance of LLMs, as removing "Syn." leads to a significant performance decline, especially under the 1-shot setting. These observations emphasize the importance of syntactically and semantically similar demonstrations in activating LLMs' reasoning in ASQP.

Methods	Backbone	Rest15				Rest16			
		0-shot	1-shot	5-shot	10-shot	0-shot	1-shot	5-shot	10-shot
THOR (Fei et al. 2023)	Llama-3.1-8B	7.88	8.02	8.65	10.01	9.44	10.78	11.37	11.95
	Llama-3.1-70B	17.62	22.49	26.78	31.47	27.61	30.04	34.34	35.91
	GPT-4o-mini	18.18	21.52	24.29	29.36	29.83	31.15	32.84	34.05
	GPT-4o	30.79	33.37	35.12	36.81	37.23	38.92	42.05	43.57
LLMs for SA (Zhang et al. 2024b)	Llama-3.1-8B	8.61	8.82	8.66	10.69	11.43	9.99	11.67	12.59
	Llama-3.1-70B	18.20	20.79	27.81	32.84	29.01	30.99	35.04	37.10
	GPT-4o-mini	19.80	18.92	24.93	30.11	31.43	29.42	33.84	35.45
	GPT-4o	34.56	35.62	36.29	37.08	39.15	39.61	43.36	45.00
SimRP (Ours)	Llama-3.1-8B	8.88	9.26	14.13	15.26	12.55	15.46	16.71	21.60
	Llama-3.1-70B	18.12	26.83	35.56	38.19	28.71	33.73	40.91	43.81
	GPT-4o-mini	19.47	26.46	33.33	35.55	31.25	39.93	40.44	43.13
	GPT-4o	34.44	37.88	41.08	**43.17**	39.33	45.50	48.62	**49.74**

Table 4: Experimental results of LLMs under zero-shot and few-shot settings.

Methods	Rest15		Rest16	
	1-shot	10-shot	1-shot	10-shot
SimRP	**37.88**	**43.17**	**45.50**	**49.74**
-[Syn.]	36.48	42.57	44.27	49.39
-[Sem.]	35.45	40.52	42.56	48.15
-[Syn., Sem.]	35.65	36.82	38.03	44.64

Table 5: Ablation studies in the ICL paradigm.

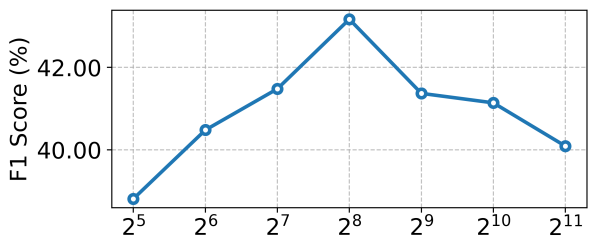


Figure 3: Influences of different syntactic vector dimensions.

Influence of d in Syn2Vec

As illustrated in Equation (4), the dimension of the syntactic vector d is a crucial hyperparameter for accurately representing sentence syntax, which further influences the retrieval of syntactically similar demonstrations. Hence, we conduct the parameter searching experiments to acquire the optimal d. Figure 3 presents the experimental results of GPT-4o on Rest15 under the 10-shot setting, with d varying from 2^5 to 2^{11}. The model's performance clearly improves as d increases, peaking at $d = 2^8$, after which it declines.

Error Analysis

In this section, we conduct an error analysis to investigate the capability of LLM in ASQP. Table 6 presents several examples from Rest15, where ID denotes the sample index, and predictions are generated by GPT-4o under the 10-shot setting. From Table 6, we can observe that LLM correctly identifies *"wings"* in the 4th case but is marked incorrect for omitting *"with chimichurri"*. The last four cases show similar situations where LLM generates correct but more detailed opinion terms. It's worth noting that the 29th case is misannotated, where the opinion term should be *"wasn't a fan"* rather than *"fan"* due to the negative sentiment. These observations highlight the rationality of LLMs' predictions and demonstrate that existing exact-matching metrics fail to fully reflect the model's performance.

Generally, detailed opinion words, co-occurred under a CsT, are rational and reasonable in the ASQP task. Hence, we propose a CsT-based fuzzy matching metric. We first extend the ground truth of aspect and opinion terms according to their CsTs. As the example illustrated in Figure 1, the labeled opinion term "fair" is extended to "a fair tab" according to the CsT of *(NP DT JJ NN)*. A predicted quadruple is considered correct if the aspect or opinion terms are included in the extended ground truth. More examples are provided in the Appendix. The performance of GPT-4o is re-evaluated using the CsT-based fuzzy metric, and the results are presented in Table 7. "Add$_{\pm w}$" in the table represents the simple fuzzy metric that extends the ground truth by adding w words forward and backward, respectively. Thus, "Add$_{\pm 0}$" is the original exact matching metric. For a fair comparison, we report the average number of words added by the CsT-based fuzzy metric, shown at subscripts of results.

From Table 7, it can be found that the proposed CsT-based fuzzy metric significantly outperforms the other fuzzy matching metrics, with fewer adding words and higher F1 scores. In addition, using the CsT-based fuzzy metric, GPT-4o under the 10-shot setting has achieved performance comparable to fine-tuned generative models on Rest15.

Related Work

Aspect-based Sentiment Analysis

ABSA is a fine-grained sentiment analysis task and has evolved into a series of subtasks, including Aspect-level sen-

ID	Sentence	Ground Truth	Predictions
4	*Guacamole at pacifico is yummy, as are the wings with chimichurri.*	*(Guacamole, yummy, positive, FQ) (wings with chimichurri, yummy, positive, FQ)*	*(Guacamole, yummy, positive, FQ) (wings, yummy, positive, FQ)*
6	*In the end you end up with a fair tab and nothing but a great time !!!*	*(Null, fair, positive, RP) (Null, great, positive, RG)*	*(Null, a fair tab, positive, RP) (Null, a great time, positive, RG)*
29	*Also, I personally wasn't a fan of the portobello and asparagus mole.*	*(portobello and asparagus mole, fan, negative, FQ)*	*(portobello and asparagus mole, wasn't a fan, negative, FQ)*
37	*I'd definitely go back again.*	*(Null, go back, positive, RG)*	*(Null, go back again, positive, RG)*
48	*She just nodded and walked off.*	*(Null, walked off, negative, SG)*	*(Null, nodded and walked off, negative, SG)*

Table 6: Error analysis on the Rest15 dataset. FQ, RP, RG, FQ, and SG are abbreviations of different categories.

Metrics	Rest15		Rest16	
	1-shot	10-shot	1-shot	10-shot
$\text{Add}_{\pm 0}$	37.88	43.17	45.50	49.74
$\text{Add}_{\pm 1}$	44.28	51.49	52.63	56.49
$\text{Add}_{\pm 2}$	47.76	55.31	55.46	60.03
$\text{Add}_{\pm 3}$	49.39	57.67	57.50	61.52
$\text{Add}_{\pm 4}$	51.13	59.13	58.97	62.78
CsT.	$53.46^{+3.0}_{-1.8}$	$59.25^{+2.9}_{-2.0}$	$59.20^{+3.0}_{-3.1}$	$63.12^{+2.8}_{-2.6}$

Table 7: Experimental results by fuzzy matching metrics.

timent Classification (ALSC) (Tang, Qin, and Liu 2016), Aspect Opinion Pair Extraction (AOPE) (Zhao et al. 2020), Aspect Sentiment Triplet Extraction (ASTE) (Peng et al. 2020b), and ASQP (Zhang et al. 2021b). ASTE and ASQP, being more challenging compound tasks, have gained increased attention in recent years, proposing various end-to-end methods to address them. These methods include span-based methods (Cai, Xia, and Yu 2021; Xu, Chia, and Bing 2021), table-filling methods (Wu et al. 2020; Chen et al. 2022), machine reading comprehension-based methods (Mao et al. 2021; Chen et al. 2021), and generative methods (Zhang et al. 2021b; Gao et al. 2022; Gou, Guo, and Yang 2023). Among them, generative methods have gradually become the mainstream due to their implementation simplicity, task universality and ability to exploit rich label information.

The core idea of generative ASQP is to transform sentiment elements into a label sequence and then use the seq2seq paradigm to learn the matching relationships between the input text and the label sequence. For instance, Zhang et al. (2021a) introduced a paraphrase modeling framework, transforming the quadruple prediction task into a text generation task by using paraphrase sentences as target sequences. MVP (Gou, Guo, and Yang 2023) explored the impact of quadruple generation order on the model's performance and proposed to vote the most reasonable quadruples generated in different orders, significantly enhancing the model's predictive capability.

In-Context Learning for ASQP

ICL is one of the key techniques in which LLMs can tackle complex tasks based on a few annotated demonstrations without additional training or gradient updates (Zhao et al. 2023). Hence, demonstrations in ICL play the most crucial role in activating the reasoning capabilities of LLMs. Numerous studies focused on how to provide better demonstrations for LLMs. Representatively, Liu et al. (2022) found that examples closely related to the target data in the embedding space yield better results. Building on this idea, Wang et al. (2022) proposed enhancing inputs by retrieving similar examples from the training set using BM25.

Owing to the developments LLM and ICL, addressing complex ABSA subtasks, such as ASQP, has become more feasible (Wang et al. 2023b). Zhong et al. (2023) observed that the zero-shot performance of LLMs is comparable to fine-tuned BERT models on the sentiment analysis tasks. Zhang et al. (2024b) further demonstrated that LLMs can match the performance of smaller models specifically trained for simple sentiment analysis tasks.

Conclusion

In this paper, we present Syn2Vec, which encodes sentence syntax as sparse vectors and uses MSE to compute syntactic similarity, allowing the retrieval of syntactically similar demonstrations. Subsequently, Sentence-BERT is utilized to select the semantically similar ones as the most related demonstrations. With these syntactically and semantically similar demonstrations, the proposed SimRP significantly enhances the reasoning capabilities of generative models. Experimental results on the standard ASQP datasets demonstrate that SimRP outperforms existing methods, both in SFT and ICL paradigms. Furthermore, we find that LLMs' performance is significantly underestimated by the exact matching evaluation metric, as their generated results are logical but may differ from annotated labels, which are often subjective and non-unique. Therefore, we propose a CsT-based fuzzy metric to better evaluate the performance of LLMs in the ASQP task. Comparison results demonstrate the superiority of the proposed CsT-based fuzzy metric.

Acknowledgments

This work is supported by the 3D visualization digital twin integrated control system (No.2023CXY0111), the solfeggio ear training intelligent robot and cloud platform research and development project for music education (No.2024CXY0102), and the public technology service platform project of Xiamen City (No.3502Z20231043).

References

Brauwers, G.; and Frasincar, F. 2022. A Survey on Aspect-Based Sentiment Classification. *ACM Computing Surveys*, 55(4).

Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; Agarwal, S.; Herbert-Voss, A.; Krueger, G.; Henighan, T.; Child, R.; Ramesh, A.; Ziegler, D.; Wu, J.; Winter, C.; Hesse, C.; Chen, M.; Sigler, E.; Litwin, M.; Gray, S.; Chess, B.; Clark, J.; Berner, C.; McCandlish, S.; Radford, A.; Sutskever, I.; and Amodei, D. 2020. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems*, volume 33, 1877–1901.

Cai, H.; Xia, R.; and Yu, J. 2021. Aspect-Category-Opinion-Sentiment Quadruple Extraction with Implicit Aspects and Opinions. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, 340–350.

Chen, H.; Zhai, Z.; Feng, F.; Li, R.; and Wang, X. 2022. Enhanced Multi-Channel Graph Convolutional Network for Aspect Sentiment Triplet Extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, 2974–2985.

Chen, S.; Wang, Y.; Liu, J.; and Wang, Y. 2021. Bidirectional Machine Reading Comprehension for Aspect Sentiment Triplet Extraction. In *The 35th AAAI Conference on Artificial Intelligence, the 33rd Conference on Innovative Applications of Artificial Intelligence, and the 11th Symposium on Educational Advances in Artificial Intelligence*, 12666–12674.

Dong, Q.; Li, L.; Dai, D.; Zheng, C.; Wu, Z.; Chang, B.; Sun, X.; Xu, J.; and Sui, Z. 2022. A survey on in-context learning. *arXiv preprint arXiv:2301.00234*.

Fei, H.; Li, B.; Liu, Q.; Bing, L.; Li, F.; and Chua, T.-S. 2023. Reasoning Implicit Sentiment with Chain-of-Thought Prompting. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, 1171–1182.

Gao, T.; Fang, J.; Liu, H.; Liu, Z.; Liu, C.; Liu, P.; Bao, Y.; and Yan, W. 2022. LEGO-ABSA: A Prompt-based Task Assemblable Unified Generative Framework for Multi-task Aspect-based Sentiment Analysis. In *Proceedings of the 29th International Conference on Computational Linguistics*, 7002–7012.

Gou, Z.; Guo, Q.; and Yang, Y. 2023. MvP: Multi-view Prompting Improves Aspect Sentiment Tuple Prediction. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, 4380–4397.

Hu, M.; Bai, Y.; Wu, Y.; Zhang, Z.; Zhang, L.; Gao, H.; Zhao, S.; and Huang, M. 2023. Uncertainty-Aware Unlikelihood Learning Improves Generative Aspect Sentiment Quad Prediction. In *Findings of the Association for Computational Linguistics*, 13481–13494.

Hu, M.; Wu, Y.; Gao, H.; Bai, Y.; and Zhao, S. 2022. Improving Aspect Sentiment Quad Prediction via Template-Order Data Augmentation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 7889–7900.

Karpukhin, V.; Oguz, B.; Min, S.; Lewis, P.; Wu, L.; Edunov, S.; Chen, D.; and Yih, W.-t. 2020. Dense Passage Retrieval for Open-Domain Question Answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, 6769–6781.

Li, Z.; Yang, Z.; Li, Y.; and Li, X. 2024. Opinion-Tree-guided Contrastive Learning for Aspect Sentiment Quadruple Prediction. In *27th International Conference on Computer Supported Cooperative Work in Design*, 1944–1951.

Liu, H.; Chatterjee, I.; Zhou, M.; Lu, X. S.; and Abusorrah, A. 2020. Aspect-Based Sentiment Analysis: A Survey of Deep Learning Methods. *IEEE Transactions on Computational Social Systems*, 7(6): 1358–1375.

Liu, J.; Shen, D.; Zhang, Y.; Dolan, B.; Carin, L.; and Chen, W. 2022. What Makes Good In-Context Examples for GPT-3? *DeeLIO 2022*, 100.

Mao, Y.; Shen, Y.; Yu, C.; and Cai, L. 2021. A Joint Training Dual-MRC Framework for Aspect Based Sentiment Analysis. In *The 35th AAAI Conference on Artificial Intelligence, the 33rd Conference on Innovative Applications of Artificial Intelligence, and the 11th Symposium on Educational Advances in Artificial Intelligence*, 13543–13551.

Min, S.; Lyu, X.; Holtzman, A.; Artetxe, M.; Lewis, M.; Hajishirzi, H.; and Zettlemoyer, L. 2022. Rethinking the Role of Demonstrations: What Makes In-Context Learning Work? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 11048–11064.

Nie, Y.; Fu, J.; Zhang, Y.; and Li, C. 2024. Modeling implicit variable and latent structure for aspect-based sentiment quadruple extraction. *Neurocomputing*, 586: 127642.

Peng, H.; Xu, L.; Bing, L.; Huang, F.; Lu, W.; and Si, L. 2020a. Knowing What, How and Why: A Near Complete Solution for Aspect-based Sentiment Analysis. *Proceedings of the AAAI Conference on Artificial Intelligence*, 8600–8607.

Peng, H.; Xu, L.; Bing, L.; Huang, F.; Lu, W.; and Si, L. 2020b. Knowing What, How and Why: A Near Complete Solution for Aspect-Based Sentiment Analysis. In *The 34th AAAI Conference on Artificial Intelligence, the 32nd Innovative Applications of Artificial Intelligence Conference, and the 10th AAAI Symposium on Educational Advances in Artificial Intelligence*, 8600–8607.

Peper, J.; and Wang, L. 2022. Generative Aspect-Based Sentiment Analysis with Contrastive Learning and Expressive Structure. In *Findings of the Association for Computational Linguistics*, 6089–6095.

Pontiki, M.; Galanis, D.; Papageorgiou, H.; Androutsopoulos, I.; Manandhar, S.; AL-Smadi, M.; Al-Ayyoub, M.; Zhao, Y.; Qin, B.; De Clercq, O.; Hoste, V.; Apidianaki, M.; Tannier, X.; Loukachevitch, N.; Kotelnikov, E.; Bel, N.; Jiménez-Zafra, S. M.; and Eryiğit, G. 2016. SemEval-2016 Task 5: Aspect Based Sentiment Analysis. In *Proceedings of the 10th International Workshop on Semantic Evaluation*, 19–30.

Pontiki, M.; Galanis, D.; Papageorgiou, H.; Manandhar, S.; and Androutsopoulos, I. 2015. SemEval-2015 Task 12: Aspect Based Sentiment Analysis. In *Proceedings of the 9th International Workshop on Semantic Evaluation*, 486–495.

Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; and Liu, P. J. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research*, 21(140): 1–67.

Ram, O.; Levine, Y.; Dalmedigos, I.; Muhlgay, D.; Shashua, A.; Leyton-Brown, K.; and Shoham, Y. 2023. In-Context Retrieval-Augmented Language Models. *Transactions of the Association for Computational Linguistics*, 11: 1316–1331.

Reimers, N.; and Gurevych, I. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 3982–3992.

Scaria, K.; Gupta, H.; Goyal, S.; Sawant, S.; Mishra, S.; and Baral, C. 2024. InstructABSA: Instruction Learning for Aspect Based Sentiment Analysis. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics*, 720–736.

Sun, X.; Zhang, K.; Liu, Q.; Bao, M.; and Chen, Y. 2024. Harnessing domain insights: A prompt knowledge tuning method for aspect-based sentiment analysis. *Knowledge-Based Systems*, 298: 111975.

Tang, D.; Qin, B.; and Liu, T. 2016. Aspect Level Sentiment Classification with Deep Memory Network. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 214–224.

Wan, H.; Yang, Y.; Du, J.; Liu, Y.; Qi, K.; and Pan, J. Z. 2020. Target-aspect-sentiment joint detection for aspect-based sentiment analysis. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, 9122–9129.

Wang, A.; Jiang, J.; Ma, Y.; Liu, A.; and Okazaki, N. 2023a. Generative Data Augmentation for Aspect Sentiment Quad Prediction. In *Proceedings of the The 12th Joint Conference on Lexical and Computational Semantics*, 128–140.

Wang, S.; Xu, Y.; Fang, Y.; Liu, Y.; Sun, S.; Xu, R.; Zhu, C.; and Zeng, M. 2022. Training Data is More Valuable than You Think: A Simple and Effective Method by Retrieving from Training Data. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, 3170–3179.

Wang, Z.; Xie, Q.; Ding, Z.; Feng, Y.; and Xia, R. 2023b. Is ChatGPT a Good Sentiment Analyzer? A Preliminary Study. *arXiv preprint arXiv:2304.04339*.

Wu, Z.; Ying, C.; Zhao, F.; Fan, Z.; Dai, X.; and Xia, R. 2020. Grid Tagging Scheme for Aspect-oriented Fine-grained Opinion Extraction. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, 2576–2585.

Xu, L.; Chia, Y. K.; and Bing, L. 2021. Learning Span-Level Interactions for Aspect Sentiment Triplet Extraction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, 4755–4766.

Xu, X.; Liu, Y.; Pasupat, P.; Kazemi, M.; et al. 2024. In-context learning with retrieved demonstrations for language models: A survey. *arXiv preprint arXiv:2401.11624*.

Yang, S.; Jiang, X.; Zhao, H.; Zeng, W.; Liu, H.; and Jia, Y. 2024. FaiMA: Feature-aware In-context Learning for Multi-domain Aspect-based Sentiment Analysis. *arXiv preprint arXiv:2403.01063*.

Zhang, H.; Cheah, Y.; Alyasiri, O. M.; and An, J. 2024a. Exploring aspect-based sentiment quadruple extraction with implicit aspects, opinions, and ChatGPT: a comprehensive survey. *Artificial Intelligence Review*, 57(2): 17.

Zhang, W.; Deng, Y.; Li, X.; Yuan, Y.; Bing, L.; and Lam, W. 2021a. Aspect Sentiment Quad Prediction as Paraphrase Generation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 9209–9219.

Zhang, W.; Deng, Y.; Liu, B.; Pan, S.; and Bing, L. 2024b. Sentiment Analysis in the Era of Large Language Models: A Reality Check. In *Findings of the Association for Computational Linguistics: NAACL 2024*, 3881–3906.

Zhang, W.; Li, X.; Deng, Y.; Bing, L.; and Lam, W. 2021b. Towards Generative Aspect-Based Sentiment Analysis. In *The 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, 504–510.

Zhang, W.; Li, X.; Deng, Y.; Bing, L.; and Lam, W. 2023. A Survey on Aspect-Based Sentiment Analysis: Tasks, Methods, and Challenges. *IEEE Transactions on Knowledge and Data Engineering*, 35(11): 11019–11038.

Zhao, H.; Huang, L.; Zhang, R.; Lu, Q.; and Xue, H. 2020. SpanMlt: A Span-based Multi-Task Learning Framework for Pair-wise Aspect and Opinion Terms Extraction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 3239–3248.

Zhao, W. X.; Liu, J.; Ren, R.; and Wen, J. 2024. Dense Text Retrieval Based on Pretrained Language Models: A Survey. *ACM Transactions on Information Systems*, 42(4): 89:1–89:60.

Zhao, W. X.; Zhou, K.; Li, J.; Tang, T.; Wang, X.; Hou, Y.; Min, Y.; Zhang, B.; Zhang, J.; Dong, Z.; et al. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*.

Zhong, Q.; Ding, L.; Liu, J.; Du, B.; and Tao, D. 2023. Can chatgpt understand too? a comparative study on chatgpt and fine-tuned bert. *arXiv preprint arXiv:2302.10198*.