COF: Adaptive Chain of Feedback for Comparative Opinion Quintuple Extraction

Qingting Xu[1], Kaisong Song[2,3], Chaoqun Liu[1], Yangyang Kang[4], Xiabing Zhou[1], Jun Lin[3], Yu Hong[1,*]

[1]School of Computer Science and Technology, Soochow University, Suzhou, China
[2] Northeastern University, China [3] Tongyi Lab, Alibaba Group, China
[4] Zhejiang University, Hangzhou, Zhejiang, China
{qtxu0801, cqliunlp, tianxianer}@gmail.com, {kaisong.sks, linjun.lj}@alibaba-inc.com,
yangyangkang@zju.edu.cn, zhouxiabing@stu.suda.edu.cn

Abstract

Comparative Opinion Quintuple Extraction (COQE) aims to extract all comparative sentiment quintuples from product review text. Each quintuple comprises five elements: subject, object, aspect, opinion and preference. With the rise of Large Language Models (LLMs), existing work primarily focuses on enhancing the performance of COQE task through data augmentation, supervised fine-tuning and instruction tuning. Instead of the above pre-modeling and in-modeling design techniques, we focus on innovation in the post-processing. We introduce a model-unaware adaptive chain-of-feedback (COF) method from the perspective of inference feedback and extraction revision. This method comprises three core modules: dynamic example selection, self-critique and self-revision. By integrating LLMs, COF enables dynamic iterative self-optimization, making it applicable across different baselines. To validate the effectiveness of our approach, we utilize the outputs of two distinct baselines as inputs for COF: frozen parameters few-shot learning and the SOTA supervised fine-tuned model. We evaluate our approach on three benchmarks: Camera, Car and Ele. Experimental results show that, compared to the few-shot learning method, our approach achieves $F1$ score improvements of 3.51%, 2.65% and 5.28% for exact matching on the respective dataset. Even more impressively, our method further boosts performance, surpassing the current SOTA results, with additional gains of 0.76%, 6.54%, and 2.36% across the three datasets.

1 Introduction

COQE is a crucial subtask in affective computing (Ma et al., 2020; Liu, 2012; Schouten and Frasincar, 2015; Kumar et al., 2020; Zhang et al., 2022a). COQE extracts all quintuples from each sentence-level text, where each quintuple consists of a subject, an object, an aspect, an opinion, and a preference. In these quintuples, the subject and object refer to two comparative entities, the aspect denotes the comparative attribute of the entities, and the opinion is a word or phrase expressing comparative sentiment. All these four elements originate from the given text. The preference is a four-class classification, categorized into *Better*, *Equal*, *Different* and *Worse*. See the example in Figure 1, given the input text, the correct quintuple result is (*It, Rebel XT, picture, better, Better*).

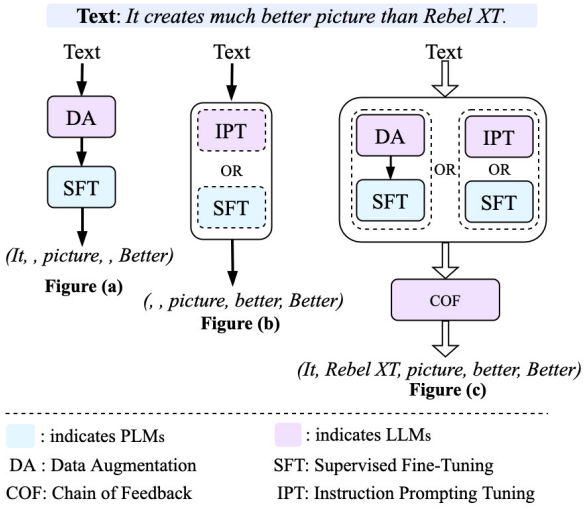

Figure 1: Results of a COQE example obtained from three perspectives.

Optimization of the COQE task can be approached from three different stages or perspectives: as shown in Figure 1(a) for data processing (e.g., data augmentation), Figure 1(b) for model training (e.g., pre-trained language models and prompting), and Figure 1(c) for post-processing feedback and optimization. Xu et al. (2023b) leverage the powerful generative capabilities of LLMs to enhance COQE from a data augmentation perspective. To mitigate error propagation in pipeline-based method (Liu et al., 2021), Yang et al. (2023) and Xu et al. (2023a) both propose an end-to-end model from the model training perspective, further

*Corresponding author.

Proceedings of the 31st International Conference on Computational Linguistics, pages 3236–3247
January 19–24, 2025.

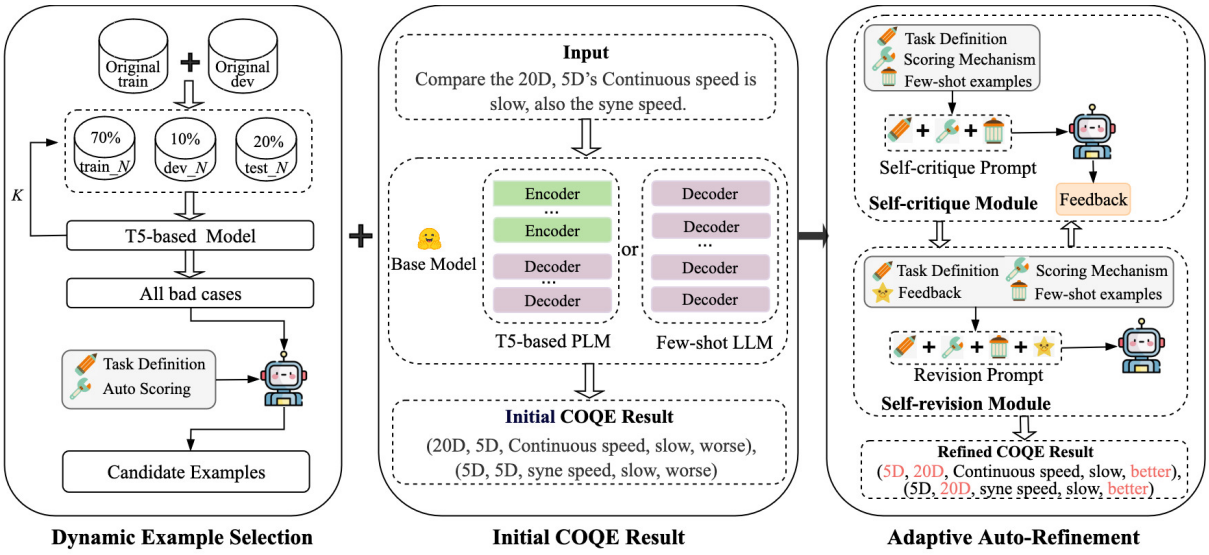

Figure 2: An overview of our adaptive chain of feedback method. In this paper, we utilize GPT-4 for direct few-shot learning and the current SOTA T5-based results as the initial input for COF, respectively.

improving COQE performance.

Unlike existing work focuses on the stages before and during modeling, this paper introduces a model-unaware adaptive chain-of-feedback method (COF) from the perspective of feedback and revision. This method consists of three core components, namely dynamic example selection, self-critique and self-revision. It's worth noting that this method can be applicable to various baseline models. Specifically, the dynamic example selection module uses the cosine similarity of dependency relations to select reference demonstrations for the current input text dynamically. Following this, the self-critique module assesses the initial quintuple result. If the score falls below the predefined threshold, the self-revision process is triggered to regenerate the quintuple. Otherwise, the initial output remains unchanged. We select the quintuple results from different baseline models as the initial input for COF and validate the effectiveness of our method. Experimental results demonstrate significant performance improvements on three datasets.

The core contributions of this paper are summarized as follows:

- We propose an innovative adaptive chain of feedback method tailored for the comparative opinion quintuple extraction task, which is post-processing, model-unware and can be applied to various baseline models.
- We design a method for dynamically selecting reference demonstrations based on syntax and semantics, avoiding manual intervention.
- We conduct experiments on three datasets, and the results demonstrate that our method significantly outperforms various baseline models.

2 Related Work

Comparative Sentence Analysis. Jindal and Liu (2006) first propose the task of comparative sentence mining. Park and Blake (2012) integrate both syntactic and semantic features, leveraging three diverse classifier to accurately identify comparative sentences. Arora et al. (2017) first pioneer to incorporate neural networks into the analysis of multi-constituent extraction in comparative review. Panchenko et al. (2019) build a new dataset called CompSent-19, which is specifically designed for the extraction of triplets: subject, object and preference. Ma et al. (2020) introduce a novel model, ED-GAT, which is an entity-aware dependency-based network. This model utilizes a multi-hop graph attention mechanism to analy the dependency graph representation of sentences. Recently, Liu et al. (2021) introduce the new task of comparative opinion quintuple extraction, aiming to extract all quintuples from each product review. Additionally, they propose a pipeline baseline for this task. To avoid error propagation, (Xu et al., 2023a) and (Yang et al., 2023) separately employ an end-to-end model and a generative model to address this issue. Furthermore, (Xu et al., 2023b) enhance the performance of COQE through a data-centric augmentation method.

Large Language Models. Since the release of

Comparative Sentence:
*The **Canon 30D** can have the **BG-E2 battery grip like** the **20D**.*
Elements:
Subject: *Canon 30D*
Object: *20D*
Aspect: *BG-E2 battery grip*
Opinion: *like*
Preference: *Equal*
Quintuples:
{(*Canon 30D, 20D, BG-E2 battery grip, like, Equal*)}

Table 1: An example of COQE task.

GPT-3 (Brown et al., 2020), LLMs have advanced rapidly (Zhao et al., 2023; Hadi et al., 2023; Chang et al., 2024). LLMs are notable not only for their massive model sizes but also for their powerful reasoning and generation capabilities (Zhao et al., 2023; Chang et al., 2024). However, few-shot learning based on LLMs still exhibits shortcomings in extraction tasks , as confirmed by recent studies (Wang et al., 2023; Han et al., 2023; Ma et al., 2023; Yuan et al., 2023; Wu et al., 2024). Directly fine-tuning an LLM requires substantial computational resources. Moreover, supervised fine-tuning on the LLM can also result in a decrease in its general capabilities. It is essential to make effective use of LLMs to serve specific tasks.

Chain-Of-Thought Prompting. Chain-Of-Thought (COT) prompting (Wei et al., 2022) improves the reasoning capabilities of LLMs by incorporating prompts or reference demonstrations. Firstly, Kojima et al. (2022) demonstrate that performance on six NLP tasks could be improved by simply adding the "let's think step by step" prompt. Subsequently, Zhang et al. (2022b) sample k examples and then integrate these examples and their reasoning processes into the prompt as in-context learning. Zhou et al. (2023) propose INFORM, which selects reference demonstrations based on information entropy.

3 Approach

In this section, we first define the task of COQE and then provide the implementation details of COF, as shown in Figure 2. Specifically, the core of COF consists of three main components: dynamic example selection, self-critique, and self-revision.

3.1 Task Definition

Given a sentiment expression sentence *X*, the COQE task extracts all quadruples {subject, object, aspect, opinion} and classifies them into four categories: *Better*, *Equal*, *Different* and *Worse*. The subject and object are the entities compared, the aspect is their comparison attribute, and the opinion expresses the sentiment toward this attribute. The opinion must be explicit, while the subject, object, and aspect can be implicit. If implicit, they are labeled as "unknown". Table 1 shows an example from the camera domain.

3.2 Baselines

In this paper, we introduce a versatile optimization method (COF), which is applicable across various models. To demonstrate its effectiveness, we employ the SOTA model and utilize few-shot learning with GPT-4 as baseline, respectively. The output quintuple results from these models are subsequently fed into COF as initial inputs.

• **T5-based Model**

Following the work of Liu et al. (2021), we select T5 (Raffel et al., 2020; Xue et al., 2021) as our baseline model. T5 model, a transformers-based model, is composed of an encoder and a decoder. For each given sentences $X = \{x_1, ..., x_n\}$, we adopt T5-encoder as the text encoder to obtain hidden representation:

$$H^e = \text{T5-Encoder}(X) \tag{1}$$

where $H^e \in \mathbb{R}^{n \times d}$ represents the output hidden representation of T5-Encoder, n is the maximum length of X, while d is the hidden dimension.

Subsequently, T5-decoder takes the encoder output H^e and previous decoder outputs $Y_{<t}$ to obtain the last hidden state:

$$h_t^d = \text{T5-Decoder}(H^e, Y_{<t}) \tag{2}$$

where, $h_t^d \in \mathbb{R}^d$. Given the h_t^d, we employ a nonlinear feed-forward network to predict the conditional probability:

$$P_t = Softmax(h_t^d W + b) \tag{3}$$

where, W and b are all trainable parameters.

• **Few-shot Learning for COQE** Acquiring annotated data typically demands significant resources and time. Few-shot learning, on the other hand, is designed to tackle this issue, with its primary aim being the creation of models that can effectively learn from a small number of training examples.

To achieve few-shot learning COQE, we use prompts **P** to generate quintuple results $\mathcal{Y}$. The prompt designed in this paper consists of three

	Camera	Car	Ele
Subject	1,649	1,520	950
Object	1,316	2,121	1,980
Aspect	1,368	1,917	1,602
Opinion	2,163	2,171	2,089
Preference	2,442	2,695	2,289
#Comsen	1,705	1,747	1,800
Non-#Comsen	1,599	1,800	1,800
Multi-#Comsen	500	550	361

Table 2: Statistics of three datasets, "#Comsen" indicates the number of comparative sentences.

main elements: task definition (**T**), demonstration examples (**D**) and the input text.

$$\mathcal{Y} = \mathcal{M}(T \oplus D \oplus X) \tag{4}$$

where, $P = T \oplus D \oplus X$, the symbol $\oplus$ denotes the concatenation operation. $\mathcal{M}$ represents the GPT-4 model. In this paper, if not particularly pointed out, $\mathcal{M}$ stands for GPT-4 model.

To mitigate the impact of dialogue history, we independently generate responses for each test sample. In this paper, we select m (m=5) demonstration examples for few-shot learning COQE. The quintuple output generated from the few-shot learning of GPT-4 serves as our initial input for COF.

3.3 Dynamic Example Selection

In few-shot learning, the quality and quantity of demonstration selection can significantly affect the quality of inference results (Song et al., 2023; Liu et al., 2024). This is because many complex inferences necessitate the intervention of expert knowledge and careful consideration of data distribution. Although some researchers (Zhou et al., 2023; Zhang et al., 2022b) have given a selection of reference demonstrations, they are not directly applicable to the COQE tasks. Static demonstration selection requires strong expert knowledge and has certain limitations. Therefore, we propose an automatic dynamic reference demonstrations selection method that does not require the introduction of additional labeled data. The details of the dynamic example selection are as follows:

• **Exampular set construction** We propose a cross-validation based hard-sample selection method. Specifically, we consolidate the original training and development sets from the COQE dataset thereby creating a new dataset named *dataset_N*. Then, we randomly divide *dataset_N* into training (*train_N*), development (*dev_N*), and test (*test_N*) subsets at a ratio of 7:1:2. We undertake supervised fine-tuning on *dataset_N* utilizing current SOTA model (Liu et al., 2021), subsequently identifying and retaining the poorly performing cases from the *test_N*. We repeat this operation K times to amass a significant collection of underperforming case samples, which is called *dataset_S*. For our experiments, we set K to 4. According to our experiments, the final statistics of incorrect examples from the three datasets are shown in Table 4.

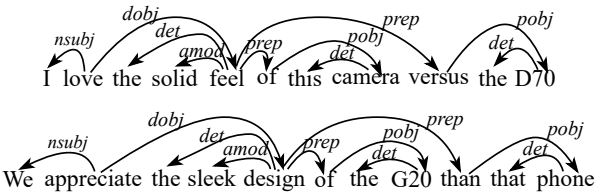

Figure 3: The result of dependency analysis of two sentences with different words and meanings.

• **Candidate demonstration scoring** Upon completing the collection of all incorrect instances, we leverage GPT-4 [1] to achieve automatic scoring for these instances by constructing appropriate prompts. Details regarding the prompts can be found in the Appendix A. For a given sentence, if there are multiple different erroneous quintuple outputs, we merge them into a single erroneous sample. Ultimately, for each erroneous sample in Table 4, we provide a corresponding error reason.

• **Demonstration selection** For COQE tasks, extraction accuracy relies heavily on syntactic structure, not solely on lexical meaning. Current demonstration selection methods overlook syntactic patterns, focusing solely on semantics. Thus, we propose a dynamic selection method based on dependency relations. We utilize the spaCy[2] toolkit to parse sentences and obtain each word along with its dependency relations. For example in Figure 3, the two sentences have different words, but they have almost the same dependency parsing results. Simply calculating word similarity is not enough to select the most suitable reference demonstration. To identify the most relevant sentences, we compute the cosine similarity between the current sentence and all sentences in the error candidate set *dataset_S*. This comprehensive analysis incorporates both word-level meaning and dependency parsing results, ensuring a sophisticated un-

[1] https://openai.com/blog/chatgpt
[2] https://spacy.io/

Models	Camera			Car			Ele		
	EM	*PM*	*BM*	*EM*	*PM*	*BM*	*EM*	*PM*	*BM*
MS (LSTM) (Liu et al., 2021)	9.05	-	-	10.28	-	-	14.90	-	-
MS (BERT) (Liu et al., 2021)	13.36	23.26	25.25	29.75	38.46	39.62	30.73	40.83	41.87
Span-graph (Fei et al., 2020)	14.53	27.13	30.15	32.08	42.78	42.05	34.86	44.92	46.72
OneEE (Cao et al., 2022)	14.10	26.09	29.06	32.46	41.14	42.53	33.51	43.62	45.27
UniCOQE (Yang et al., 2023)	**31.95**	42.39	44.44	36.55	51.60	53.80	35.46	51.47	54.05
UniCOQE*	31.16	41.01	43.12	36.45	57.36	61.05	35.76	52.27	57.17
UniCOQE* + COF (static)	28.33	37.13	38.76	40.38	58.47	65.16	37.51	52.87	57.71
UniCOQE* + COF (dynamic)	31.92	**43.55**	**45.67**	**42.99**	**59.36**	**66.10**	**38.12**	**53.55**	**60.76**

Table 3: Three different matching $F1$-score for various COQE methods, with the best results highlighted in bold. The first four lines of experimental results are sourced from Liu et al. (2021). The symbol '-' indicates that performance is not reported in the paper, whereas the mark '*' denotes the performance we reproduced.

	Camera	Car	Ele
#Err	2,148	1,886	1,830
#Sent	1,016	906	912
#Aver	2.11	2.08	2.01

Table 4: Statistics of error sample for three datasets. "#Err" denotes the total number of incorrect samples collected in each dataset, "#Sent" represents the count of incorrect sentences, and "Aver" shows the average number of incorrect samples per sentence.

derstanding of semantic and structural relations. Subsequently, we rank the sentences and return the top m most similar ones, along with their respective scores. Finally, we use these m samples as reference demonstrations.

3.4 Adaptive Auto-Refinement

As illustrated in Figure 2, adaptive auto-refinement is fundamentally composed of two key components: self-critique and self-revision. The self-critique module is responsible for scoring the initial quintuple results of the current input text. If the score falls below a predefined threshold, the self-revision module is invoked to regenerate the correct quintuple outcome for the current text.

Self-Critique The self-critique module is designed to evaluate the given input text along with its corresponding quintuple results. To accomplish this, we utilize GPT-4 and a specially crafted prompt $\mathbf{P_c}$ to score the quintuple. $\mathbf{P_c}$ includes task definition $\mathbf{T}$, scoring mechanism $\mathbf{S}$ and few-shot demonstration examples $\mathbf{D}$.

$$\mathcal{Y}_c = \mathcal{M}(T \oplus S \oplus D \oplus X) \tag{5}$$

The scoring results include the reason and score, which is called *feedback*. Each element within the quintuple carries a value of one point, summing up to a full score of five points for a complete quintuple (for Table 1 COQE result, the total score is five points). Given that a text comprises q quintuples, the threshold is set to q*5. If the score equals the threshold, it signifies the correctness of the current quintuples, thereby eliminating the need for further optimization. However, if the score falls below the threshold, the self-revision module is invoked to improve the results.

Self-Revision The self-revision module enhances the COQE results of the input text. Contrasting with the self-critique module, the prompt $\mathbf{P_r}$ in the self-revision module incorporates additional feedback $\mathbf{F}$ generated during the first stage.

$$\mathcal{Y}_r = \mathcal{M}(T \oplus S \oplus D \oplus F \oplus X) \tag{6}$$

Auto Iterative Optimization The adaptive auto-refinement method enables continuous self-iteration. In our experiments, we set a maximum number k of iterations. When the score generated by the self-critique module equals the current threshold, or the current iteration number exceeds the set maximum number, we stop the iterative optimization and output the optimized result.

4 Experimentation

4.1 Datasets and Evaluation Metrics

Datasets We conduct various experiments on three datasets: Camera, Car and Ele. The statistics overview of these datasets are shown in Table 2.

• **Camera** is a English dataset. Kessler and Kuhn (2014) annotate quintuple (subject, object, aspect, scale, predicate) from the camera domain comment text. On this basis, Liu et al. (2021) retain the first three elements and annotate comparative opinion and preference, forming a new COQE dataset.

• **Car** is a Chinese COQE dataset. Based on the newly released COAE dataset from the car domain by Tan et al. (2013), Liu et al. (2021) further enriches it by adding annotations for comparative opinion and comparative preference.
• **Ele** is also a Chinese dataset. This dataset shares the same source and annotation rules as dataset Car. The only difference is the domain, which comes from the field of electronic products.

Evaluation Metrics We utilize Precision (*P*), Recall (*R*) and F1-scores (*F*1) to evaluate the performance of our method. To provide a more comprehensive assessment for the COQE task, follow Liu et al. (2021)'s work, we employ three different matching strategies: Exact Match (*EM*), Proportional Match (*PM*), and Binary Match (*BM*).

EM checks if the predicted results align perfectly with the gold standard across all prediction elements. *PM* adopts a more relaxed approach, granting some flexibility in the comparison by assessing proportional similarities between the predicted and the desired quintuples. *BM* simplifies the assessment to a straightforward binary outcome: if there's any overlap between the predicted and the expected quintuple, the score is 1, else it's considered a mismatch, the score is 0. The formulas of these three matching strategies are as follows:

$$EM = \begin{cases} 0 & \exists\,(p_i \neq g_i) \\ 1 & otherwise \end{cases} \tag{7}$$

$$PM = \begin{cases} 0 & \exists\,(p_i \cap g_i = \varnothing) \\ \frac{\sum_i len(p_i \cap g_i)}{\sum_i len(g_i)} & otherwise \end{cases} \tag{8}$$

$$BM = \begin{cases} 0 & \exists\,(p_i \cap g_i = \varnothing) \\ 1 & otherwise \end{cases} \tag{9}$$

where, g_i and p_i represent the i-th element in each gold and predicted quintuple, respectively. The variable i ranges from 1 to 5.

4.2 Hyperparameter Settings

The self-critique and self-revision modules both utilize GPT-4. We set the maximum number of iterations to 5 for both modules. Additionally, we adjust the temperature parameter to 0.5 and set the maximum number of tokens per generation to 500 for GPT-4. For the three datasets, we set m = 5 for static example selection, m = 3 and K = 4 for dynamic example selection. We run all fine-tuning experiments in a single Tesla V100.

4.3 Compared Models

For comparative evaluation, we consider the following models:
• **MS (LSTM)** utilizes LSTM (Schuster and Paliwal, 1997) as the text encoder and CRF to extract possible properties in a sentence. (Liu et al., 2021) generate possible quadruples via Cartesian product, choose valid ones, and classify them into four preference categories.
• **MS (BERT)** adopts BERT (Devlin et al., 2019) as text encoder, which is the only difference from MS (LSTM) (Liu et al., 2021).
• **Span-graph** treats entity relation extraction task as a quintuple prediction problem and design an end-to-end model to solve it (Fei et al., 2020).
• **OneEE** is a one-stage model that efficiently identifies relations among trigger or argument words through an adaptive event fusion module and a distance-aware predictor (Cao et al., 2022).
• **UniCOQE** is a generative extraction model designed to address the error propagation problem in the pipeline-based model, achieving the current state-of-the-art performance Yang et al. (2023).

4.4 Main Results

We propose an adaptive chain of feedback method through the dynamic selection of reference demonstrations (dynamic). Additionally, we validate the approach of adaptive optimization using a few fixed, manually selected reference demonstrations (static). We conduct experiments on three distinct datasets, and the results are presented in Table 3.

It can be concluded that our COF approach achieves significant improvements on all three datasets compared to the baseline. Specifically, on the two Chinese datasets, both static and dynamic reference demonstration selection methods demonstrate substantial improvements. The most notable improvement is observed on the Car dataset, where the *EM*, *PM* and *BM* *F*1 scores increase by 6.54%, 2.0%, 5.05%, respectively. It is worth noting that the selection of static reference demonstrations relies heavily on expert knowledge and a deep understanding of the task. This selection holds a significant influence over the experimental results.

4.5 COF for Few-shot Learning COQE

There is undoubtedly a concern that the improvements observed with our COF method may be attributed to the inherent capabilities of the GPT-4 model itself. To address this concern, we first em-

Dataset	Model	*EM*			*PM*			*BM*		
		P	***R***	***F1***	***P***	***R***	***F1***	***P***	***R***	***F1***
Camera	Few-shot	10.42	10.29	10.35	19.22	18.98	19.1	21.04	20.78	20.91
	Few-shot + COF (static)	12.94	12.94	12.85	23.98	23.64	23.81	25.68	25.31	25.49
	Few-shot + COF (dynamic)	**13.96**	**13.76**	**13.86**	**24.35**	**24.00**	**24.17**	**26.25**	**25.87**	**26.06**
Car	Few-shot	22.96	24.00	23.47	43.82	45.81	44.79	50.08	52.35	51.19
	Few-shot + COF (static)	25.04	25.91	25.47	46.44	48.05	47.23	52.94	54.78	53.85
	Few-shot + COF (dynamic)	**25.81**	**26.43**	**26.12**	**47.73**	**48.89**	**48.30**	**54.50**	**55.83**	**55.15**
Ele	Few-shot	18.26	21.96	19.94	32.27	38.79	35.23	35.62	42.83	38.89
	Few-shot + COF (static)	19.78	23.91	21.65	35.23	42.58	38.56	39.57	47.83	43.31
	Few-shot + COF (dynamic)	**21.09**	**25.22**	**22.97**	**36.24**	**43.33**	**39.47**	**40.91**	**48.91**	**44.55**

Table 5: Three distinct matching strategies for precision, recall and $F1$-score for few-shot examples COQE.

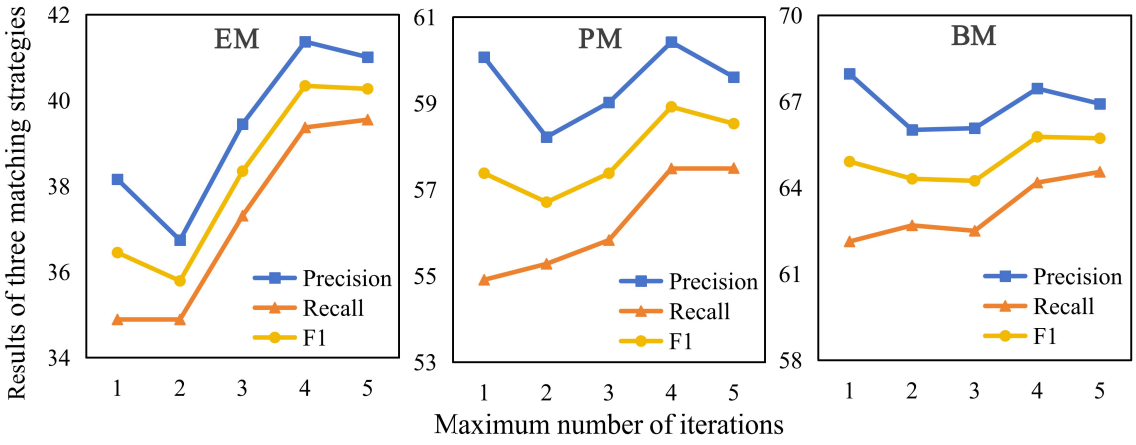

Figure 4: Effect of maximum iteration on three matching strategies for the Car dataset.

ploy GPT-4 to achieve few-shot learning for the COQE task. Subsequently, we use the generated outputs as the initial input for our COF method.

Table 5 shows the test results. The few-shot learning performance of GPT-4 on COQE task on three datasets is significantly lower than the performance of most existing SFT models. This also demonstrates that the improvements achieved by our COF method are not due to the inherent capabilities of GPT-4, but rather to the effectiveness of our approach itself. Furthermore, it can be observed that when the initial input for COF is suboptimal, our adaptive optimization method with dynamic reference demonstration selection is more effective compared to static reference demonstration selection. Besides, our dynamic approach requires no manual intervention and demonstrates greater adaptability.

4.6 Impact of Maximum Number of Iterations

We verify the influence of different iterations on the final optimization results on the Car dataset, aiming to gain a deeper understanding of the optimization process. The experimental results are shown in Figure 4. It can be observed that the best performance is achieved for three matching strategies when the maximum number of iterations is 4.

When the number of iterations is less than or equal to 2, the performance is lower than the initial baseline model. As the number of iterations increases (more than 2), the performance gradually improves and reaches the optimal level. Subsequently, with further increases in iterations, the performance improvement becomes less significant. Therefore, selecting an appropriate number of iterations is crucial for achieving optimal performance of the COQE task on the Car dataset, as both too few and too many iterations hinder reaching the best optimization results.

4.7 Efficient Fine-tuning Directly Based on Various LLMs

To verify the performance of large language models on the COQE task, we conduct a series of rigorous experiments. Specially, we choose LLaMA2 (Touvron et al., 2023), LLaMA3 [3] and Qwen [4] as

[3]https://llama.meta.com/llama3/
[4]https://tongyi.aliyun.com/

Dataset	Model	*EM*			*PM*			*BM*		
		P	***R***	***F1***	***P***	***R***	***F1***	***P***	***R***	***F1***
Camera	LLaMA2-7B	23.33	20.78	21.98	33.88	30.19	31.93	35.80	31.89	33.73
	LLaMA3-8B	28.19	28.13	28.16	39.89	39.81	39.85	41.77	41.68	41.73
Car	LLaMA2-7B	34.49	31.29	32.81	50.51	45.83	48.05	53.76	48.78	51.15
	Qwen-7B	41.25	35.84	38.35	57.88	50.29	53.82	65.79	57.17	61.18
Ele	LLaMA2-7B	34.05	31.09	32.50	51.15	46.71	48.83	54.05	49.35	51.59
	Qwen-7B	35.39	36.09	35.74	53.62	54.67	54.14	61.41	62.61	62.00

Table 6: The performance of directly fine-tuning various LLMs on three datasets.

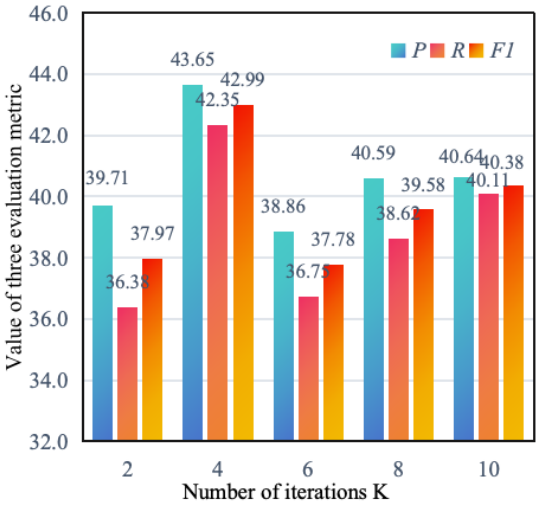

Figure 5: Influence of different error candidate sets on final COQE experiment results.

Sentence Gold	Realistically there will be no difference in your photos. (photos, photos, unknown, no difference, equal)
Few-shot Few-shot*	(unknown, unknown, photos, no difference, equal) (photos, photos, unknown, no difference, equal)
T5-based T5-based*	(unknown, unknown, unknown, unknown, unknown) (photos, photos, unknown, no difference, equal)

Table 7: Case study for Camera dataset. The mark '*' represents the corresponding COF-enhanced COQE result for each initial model.

benchmark models for various datasets. We utilize LORA [5] (Hu et al., 2021) for fine-tuning and prediction. For their typical configurations, we uniformly set the rank value to 8, the alpha value to 16, and the dropout ratio is 0.05. We report the experimental results in Table 6.

As depicted in Table 6, when examining the experimental performance of two different baselines (LLaMA2-7B and LLaMA3-8B) for the same English dataset, we can observe that: with the same model architecture, an increase in model parameters (from 7B to 8B) directly leads to a significant improvement. This strongly suggests that a larger model capacity tends to yield better comprehension and processing capabilities. Further analysis of the Chinese datasets indicates that models with the same amount of parameters do much better on Chinese tasks when trained on more Chinese data, compared to English data. This shows how important it is to use language-specific data to improve performance and how training should be tailored to the unique features of each language.

4.8 Effect of Error Candidate Set Size

To explore how the size of the error candidate set ultimately impacts the experimental results of COQE, we conduct an analysis experiment utilizing the Chinese Car dataset. The results of the experiment are shown in Figure 5.

As illustrated in Figure 5, the optimal value of COQE is achieved when the iteration number K is set to 4. If the iteration number is too low, there are not enough reference demonstrations for the refine module to learn from, leading to poor performance. Conversely, even though it might seem that increasing the iteration count could further enhance performance, in reality, exceeding a certain threshold does not bring about positive effects. Furthermore, an unnecessary high iteration count wastes valuable computational resources.

4.9 Case Study

Table 7 presents a case study for the Camera dataset's COQE results. The sentence serves as the input for the COQE models. The "Gold" indicates the expected COQE output for the given input. As shown in Table 7, few-shot learning only correctly identify comparative opinion and preference, while the T5-based model fails to identify any entities correctly, outputting "unknown" for all

[5]Implemented with LLaMA-Facotry: https://github.com/hiyouga/LLaMA-Factory/

elements. After applying the COF method, both models' outputs improved significantly, matching the ground truth exactly. This case study demonstrates that the COF method effectively enhances the COQE results for both few-shot and T5-based models, aligning the outputs with the ground truth.

5 Conclusion

We propose an adaptive chain of feedback method for the COQE task, which can be applied to any baseline model. Our method includes a dynamic example selection technique that outperforms static methods requiring excessive manual intervention. Additionally, we devise a continuously self-iterative adaptive optimization method. We select two distinct baseline models and validate them on the test sets of three datasets. The experimental results demonstrate the effectiveness and generalization of our approach.

Limitations

The COF method proposed in this paper can effectively improve the performance of COQE tasks, but it still has certain limitations. First, in this paper, we only explore the refinement capability of using GPT4. The ability of our refine module to perform on other LLMs remains to be further explored. Second, although refining with the LLM does not require additional training steps, this process inevitably introduces higher computational resource consumption. To address the limitations, a deeper exploration is warranted.

Acknowledgements

This work was supported by the National Natural Science Foundation of China, specifically under Grant Numbers 62076174, 62376182 and 62106039. We express our gratitude to anonymous reviewers for providing valuable and insightful comments that significantly enhance and refine the overall quality of this paper.

References

Jatin Arora, Sumit Agrawal, Pawan Goyal, and Sayan Pathak. 2017. Extracting entities of interest from comparative product reviews. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 1975–1978.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Proceedings of the 34th Conference on Neural Information Processing Systems*, pages 1–25.

Hu Cao, Jingye Li, Fangfang Su, Fei Li, Hao Fei, Shengqiong Wu, Bobo Li, Liang Zhao, and Donghong Ji. 2022. OneEE: A one-stage framework for fast overlapping and nested event extraction. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 1953–1964. International Committee on Computational Linguistics.

Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, Wei Ye, Yue Zhang, Yi Chang, Philip. S Yu, Qiang Yang, and Xie Xing. 2024. A survey on evaluation of large language models. *ACM Transactions on Intelligent Systems and Technology*, 15(3):1–45.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina. Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, page 4171–4186.

Hao Fei, Yafeng Ren, and Donghong Ji. 2020. Boundaries and edges rethinking: An end-to-end neural model for overlapping entity relation extraction. *Information Processing & Management*, 57(6):102311.

Muhammad Usman Hadi, Rizwan Qureshi, Abbas Shah, Muhammad Irfan, Anas Zafar, Muhammad Bilal Shaikh, Naveed Akhtar, Jia Wu, and Seyedali Mirjalili. 2023. A survey on large language models: Applications, challenges, limitations, and practical usage. *Authorea Preprints*.

Ridong Han, Tao Peng, Chaohao Yang, Benyou Wang, Lu Liu, and Xiang Wan. 2023. Is information extraction solved by chatgpt? an analysis of performance, evaluation criteria, robustness and errors. *arXiv preprint arXiv:2305.14450*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Nitin Jindal and Bing Liu. 2006. Mining comparative sentences and relations. In *Proceedings of the twenty AAAI Conference on Artificial intelligence*, pages 1331–1336. American Association for Artificial Intelligence.

Wiltrud Kessler and Jonas Kuhn. 2014. A corpus of comparisons in product reviews. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation*, pages 2242–2248.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in Neural Information Processing Systems*, 35:22199–22213.

Akshi Kumar, Kathiravan Srinivasan, Wen-Huang Cheng, and Albert Y Zomaya. 2020. Hybrid context enriched deep learning model for fine-grained sentiment analysis in textual and visual semiotic modality social data. *Information Processing & Management*, 57(1):102141.

Bing Liu. 2012. Sentiment analysis and opinion mining. *Synthesis Lectures on Human Language Technologies*, 5(1):1–167.

Fan Liu, Tianshu Zhang, Wenwen Dai, Chuanyi Zhang, Wenwen Cai, Xiaocong Zhou, and Delong Chen. 2024. Few-shot adaptation of multi-modal foundation models: A survey. *Artificial Intelligence Review*, 57(10):268.

Ziheng Liu, Rui Xia, and Jianfei Yu. 2021. Comparative opinion quintuple extraction from product reviews. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3955–3965.

Nianzu Ma, Sahisnu Mazumder, Hao Wang, and Bing Liu. 2020. Entity-aware dependency-based deep graph attention network for comparative preference classification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics,*, pages 5782–5788. Association for Computational Linguistics.

Yubo Ma, Yixin Cao, Yong Hong, and Aixin Sun. 2023. Large language model is not a good few-shot information extractor, but a good reranker for hard samples! In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10572–10601.

Alexander Panchenko, Alexander Bondarenko, Mirco Franzek, Matthias Hagen, and Chris Biemann. 2019. Categorizing comparative sentences. In *Proceedings of the 6th Workshop on Argument Mining*, pages 136–145.

Dae Hoon Park and Catherine Blake. 2012. Identifying comparative claim sentences in full-text scientific articles. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, pages 1–9. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Kim Schouten and Flavius Frasincar. 2015. Survey on aspect-level sentiment analysis. *IEEE Transactions on Knowledge and Data Engineering*, 28(3):813–830.

Mike Schuster and Kuldip K Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681.

Yisheng Song, Ting Wang, Puyu Cai, Subrota K Mondal, and Jyoti Prakash Sahoo. 2023. A comprehensive survey of few-shot learning: Evolution, applications, challenges, and opportunities. *ACM Computing Surveys*, 55(13s):1–40.

Songbo Tan, Kang Liu, Suge Wang, and Xiangwen Liao. 2013. Overview of chinese opinion analysis evaluation 2013. In *Proceedings of the Fifth Chinese Opinion Analysis Evaluation*, pages 1–29.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Zengzhi Wang, Qiming Xie, Yi Feng, Zixiang Ding, Zinong Yang, and Rui Xia. 2023. Is chatgpt a good sentiment analyzer? a preliminary study. *arXiv preprint arXiv:2304.04339*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

Jialiang Wu, Yi Shen, Ziheng Zhang, and Longjun Cai. 2024. Enhancing large language model with decomposed reasoning for emotion cause pair extraction. *arXiv preprint arXiv:2401.17716*.

Qingting Xu, Yu Hong, Fubang Zhao, Kaisong Song, Jiaxiang Chen, Yangyang Kang, and Guodong Zhou. 2023a. Gcn-based end-to-end model for comparative opinion quintuple extraction. In *2023 International Joint Conference on Neural Networks (IJCNN)*, pages 1–6. IEEE.

Qingting Xu, Yu Hong, Fubang Zhao, Kaisong Song, Yangyang Kang, Jiaxiang Chen, and Guodong Zhou. 2023b. Low-resource comparative opinion quintuple extraction by data augmentation with prompting. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 3892–3897.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498.

Zinong Yang, Feng Xu, Jianfei Yu, and Rui Xia. 2023. Unicoqe: Unified comparative opinion quintuple extraction as a set. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 12229–12240.

Chenhan Yuan, Qianqian Xie, and Sophia Ananiadou. 2023. Zero-shot temporal relation extraction with chatgpt. In *The 61st Annual Meeting Of The Association For Computational Linguistics*.

Wenxuan Zhang, Xin Li, Yang Deng, Lidong Bing, and Wai Lam. 2022a. A survey on aspect-based sentiment analysis: tasks, methods, and challenges. *IEEE Transactions on Knowledge and Data Engineering*, 35(11):11019–11038.

Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. 2022b. Automatic chain of thought prompting in large language models. In *The Eleventh International Conference on Learning Representations*, pages 1–32.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jianyun Nie, and Jirong Wen. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*.

Chuyue Zhou, Wangjie You, Juntao Li, Jing Ye, Kehai Chen, and Min Zhang. 2023. Inform: Information entropy based multi-step reasoning for large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3565–3576.

A Dynamic Example Selection Prompt

"Scoring Guideline for Comparative Opinion Quintuple Extraction Task
Task Overview: Extract all (subject, object, aspect, opinion, preference) quintuples from the given text and score each predicted quintuples based on the gold answers.
Scoring Rules:
1. Score each quintuple item by item: Compare the predicted quintuples with gold quintuples one by one.
- If the number of predicted quintuples does not match the gold, the missing or extra elements will be scored as 0 point, with the reason noted.
- Score each element in each quintuple, strictly based on the gold quintuple.
2. Scoring Criteria for Each Element:
Subject:
- If the subject is explicitly mentioned in the given text and the predicted subject matches the gold subject, score 1 point.
- If the subject is not explicitly mentioned in the given text, and both predicted and gold subject are 'unknown', score 1 point.
- If the subject is not explicitly mentioned in the given text, but the predicted subject is wrongly extracted from the given text, score 0 point.
- If the subject is explicitly mentioned in the given text, but the predicted subject does not match the gold subject, score 0 point.
Object:
- If the object is explicitly mentioned in the given text and the predicted object matches the gold object, score 1 point.
- If the object is not explicitly mentioned in the given text, and both predicted and gold object are 'unknown', score 1 point.
- If the object is not explicitly mentioned in the given text, but the predicted object is wrongly extracted from the given text, score 0 point.
- If the object is explicitly mentioned in the given text, but the predicted object does not match the gold object, score 0 point.
Aspect:
- If the aspect is explicitly mentioned in the given text and the predicted aspect matches the gold aspect, score 1 point.
- If the aspect is not explicitly mentioned in the given text, and both predicted and gold aspect are 'unknown', score 1 point.
- If the aspect is not explicitly mentioned in the given text, but the predicted aspect is wrongly extracted from the given text, score 0 point.
- If the aspect is explicitly mentioned in the given text, but the predicted aspect does not match the gold aspect, score 0 point.
Opinion:
- Must be extracted from the given text, cannot use 'unknown'. If the prediction matches the gold, score 1 point.
- If the predicted opinion is from the given text, but does not match the gold one, score 0 point.
Preference:
- The predicted preference must accurately match one of the predefined categories ('better', 'worse', 'equal', 'different') and be fully consistent with the gold, score 1 point.
- If the predicted preference is not among the predefined categories, score 0 point.
- If the predicted preference is not consistent with the gold one, score 0 point.
Next, please directly give the score and reason of each predicted quintuple based on the given input text and gold quintuples.
When scoring each quintuple, please provide the score and reason in the following format:
Quintuple X: Subject: score, reason (with reference to the gold quintuple)
Object: score, reason (with reference to the gold quintuple)
Aspect: score, reason (with reference to the gold quintuple)
Opinion: score, reason (with reference to the gold quintuple)
Preference: score, reason (with reference to the gold quintuple)
Quintuple X total score: Total score of Quintuple X
...(repeat until all quintuples are scored)
Total score sum: Sum of scores for all quintuples"