ECAI 2024
U. Endriss et al. (Eds.)

doi:10.3233/FAIA240959

Iteratively Calibrating Prompts for Unsupervised Diverse Opinion Summarization

Jian Wang[a], Yuqing Sun[a,*], Yanjie Liang[a], Xin Li[a,] and Bin Gong[a]**

[a]School of Software, Shandong University

Abstract. Diverse opinion summarization aims to generate a summary that captures multiple opinions in texts. Although large language models (LLMs) have become the main choice for this task, the performance is highly depend on prompts. In this paper, we propose a self-evaluation based prompt calibration framework to stimulate LLM for generating high quality summary. It adopts the reinforcement learning mechanism to calibrate prompts for maximizing the reward of summary. The framework contains three parts. In the prompt construction part, we design the prompt that contains topic, task instruction and key opinion reference. The topic indicates the main focus of documents, the instruction describes the task with natural language and the key opinion reference is the explicit constraint on the expected opinions. In the reward part, for each summary, its coverage score and diversity score are used to represent the semantic coverage to the source documents and the inter opinion differences, respectively. The prompt calibration part selects the sentences in generated summaries to calibrate the prompts for the next iteration. With this framework, we use a LLM with 7B parameters to generate summaries, which outperforms large GPT-4 and multiple strong baselines. The ablation studies indicate the effectiveness of the iterative calibration process. We analyze the opinion difference in terms of the tendencies of sentences in summaries and use the Natural Language Inference (NLI)-based method to evaluate the faithfulness of summaries. Experiment results show that our method generates summaries with high opinion difference and faithfulness.

1 Introduction

The diverse opinion summarization task aims to generate summaries that contain diverse opinions for the given documents. These diverse opinions contain the opposing opinions or different aspects of the same opinion. Due to the subjectivity, judgment, and open-to-debate nature of opinions, it is challenging and time-consuming to construct large-scale annotated data. Since large language models (LLMs) show good performance in multiple zero-shot tasks, they have become the main choice for the task. The most common way of using LLM is to construct prompts as shown in Fig 1. However, we can see that the quality of summaries is sensitive to different prompts. Compared to prompt1, the generated summary by prompt2 contains more opinions and is more similar to the reference summary, since prompt2 contains some fine-grained guidance information. These findings inspire us to find and add some guidance information into prompts so as to optimize the generated results of LLM.

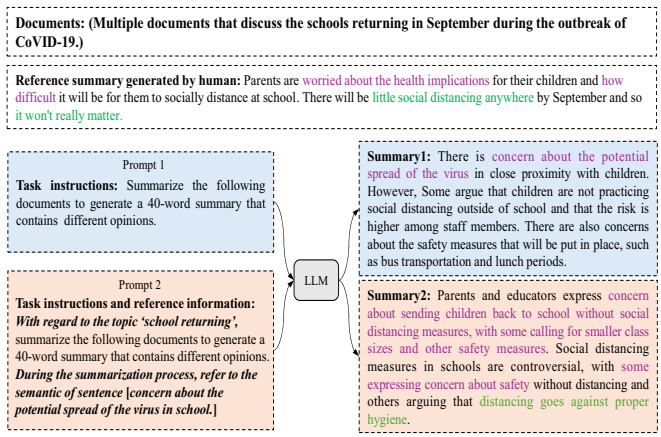

Figure 1. Different prompts lead to different results. We add the opinion reference in prompt 2. Texts marked with the same color have similar semantics to the reference summary.

Recently, multiple methods have adopted different prompts for the summarization task. For example, the Topic-Chunking-Generation (TCG) method refines the summarization task into the topic extraction, clustering and summary generation tasks, then designs prompts for each sub-task [4]. The Web-enhanced General Language Model (WebGLM) adopts context-learning by searching some similar instances on the web [18]. The Chain-of-Thought (COT) method extracts valuable elements such as events and times from documents to construct a prompt in the form of chain of thought [26]. These methods are difficult to form the *opinion diversity* focused prompts.

In order to construct good prompts without supervision, we provide a self-evaluation based prompt calibration framework that iteratively calibrates prompts to maximize the reward of summary. The framework includes the prompt construction part, the reward calculation part and the prompt calibration part. In the first part, we construct prompts containing the topic, task instruction and key opinion reference. The topic is the main focus discussed in multiple documents, which is obtained by counting the frequencies of noun phrases and verb phrases in documents. The task instruction is the natural language description of the diverse opinion summarization task. The key opinion reference is the diverse semantic constraints of opinions, which can be calibrated by the third part. Then summaries are generated based on the constructed prompts. In the second part, we use LLM to check whether each document supports the sentences in

* Corresponding Author. Email: sun_yuqing@sdu.edu.cn
** Corresponding Author. Email: lixincas@126.com

summary. Based on the check results, we construct the support document set for sentences in summary and compute the coverage score and diversity score for the summary. The coverage score is computed by the proportion of support document set relative to the source documents, and the diversity score is computed by the distance between the support document set of sentences. In the prompt calibration part, we iteratively select some sentences from the generated summaries to update the key opinion reference in prompt, enabling LLM to generate summary with multiple opinions and is faithful to the source documents.

2 Related Works

2.1 LLM for opinion summarization

Large models such as Vicuna [8], GPT-3 [7], and GPT-4 have demonstrated good capabilities for the zero-shot and few-shot summarization task. Based on LLMs, users construct different guidance information for generating summaries. For example, the Topic-Chunking-Generation (TCG) method divides the summarization task into three stages that include topic classification, chunk summarization, and summary generation. Each stage is completed with a specific prompt [4]. The Chain-of-Thought (CoT) method extracts important elements such as the time and event from the original documents and constructs a chain of thought prompt to guide LLMs for the summarization task [26]. The above methods construct prompts based on human experience or the source documents, they lack the diversity-focused guidance to constrain the summarization process. The Chain of Density (COD) method adopts an iterative generation strategy for summarization. It first generates a summary through an initial prompt and finds some entities that appear in the original documents but not in the generated summary [1]. These entities are then added to prompt for generating a more informative summary at the next generation process. However, the entities cannot reflect the semantic differences of opinions.

2.2 Constructing pseudo-summaries for opinion summarization

Another category of methods constructs pseudo samples by selecting a document from a document set as a pseudo-summary, and selecting documents that are semantically related to the pseudo-summary as the source documents[13, 21, 20]. Then they train the summarization models through supervised learning. The key difference among these methods lies in the way of constructing pseudo-summaries. For example, the Consistency-based Opinion Summarization method (ConsistSum) uses the distance between the semantic, sentiment, and aspect distributions to select the center document as a pseudo-summary [13]. DenoiseSum randomly selects a document from the document set as a pseudo-summary and adds some noise to the pseudo-summary to form the source documents[2]. Recently, a strong method OPINESUM obtains propositions by slicing the documents at conjunctions, periods and commas. The pseudo-summary is then constructed by concatenating the propositions that are entailed by a large number of documents [20]. The above methods show good performance in the unsupervised scenarios. However, a pseudo-summary typically contains only consensus and does not reflect the differences in opinions, making it difficult for the model to learn the diverse opinions.

3 Our Method

3.1 Problem and framework

Given a document set $\mathbb{D} = \{d_1, d_2, \cdots, d_{|\mathbb{D}|}\}$, the diverse opinion summarization task aims to generate a summary $\boldsymbol{s}$ that contains multiple opinions in $\mathbb{D}$. The key point of this task is to evaluate whether the sentences in $\boldsymbol{s}$ have cover the main opinions in $\mathbb{D}$ and the contained opinions are diverse. Since the efficient way to control LLM is the prompts, we design the iterative prompt calibration framework for evaluating the generated summary and adjusting the prompt (**CPSum** for short). The details of this framework are given below, also shown in Fig.2.

In this framework, the opinion summarization is formalized as the *Contextual Markov Decision Process* (CMDP) with the observable context, denoted as $(\mathcal{S}, \mathcal{A}, \mathcal{C}, \mathcal{R}, \pi)$ [12], where the state space $\mathcal{S}$ is defined over the entire summary space, the action space $\mathcal{A}$ is defined over the prompt space, the context $\mathcal{C}$ refers to the document set $\mathbb{D}$, $\mathcal{R}$ is the reward function and π is the policy. To describe clearly, we use the subscript t to represent the variable at the t-th iteration, such as a_t and $\boldsymbol{s}_t$ denote the prompt and the generated summary at the t-th iteration, respectively. Then the summarization process can be formalized as:

- **action**. An action is to construct a prompt $\boldsymbol{a} \in \mathcal{A}$ containing three parts: the topic $\boldsymbol{F}$ of the source documents, the key opinion reference $\boldsymbol{k}$ and the natural language instruction of the summarization task $\boldsymbol{I}$, formally $\boldsymbol{a} = [\boldsymbol{F}, \boldsymbol{k}, \boldsymbol{I}]$.
- **reward**. The reward of a state $\boldsymbol{s}$, i.e, the generated summary, is computed against the current prompt $\boldsymbol{a}$ and the context $\mathcal{C}$, formally $\mathcal{R}(\boldsymbol{s}|\boldsymbol{a}, \mathcal{C})$.
- **policy**. A policy π is a function for choosing a new action $\boldsymbol{a}_{t+1}$ given the current state $\boldsymbol{s}_t$, current action $\boldsymbol{a}_t$, and the context $\mathcal{C}$, formally $\pi(\boldsymbol{a}_{t+1}|\boldsymbol{s}_t, \boldsymbol{a}_t, \mathcal{C})$.

3.2 Prompt construction

To help LLM focus on a specific topic for summarization, we extract the topic discussed in the source documents $\mathbb{D}$. According to linguistics [9, 17], noun phrases(NP) and verb phrases (VP) play crucial roles in the expression of opinions. We extract these phrases from the document set and select the phrase with the highest frequency as topic $\boldsymbol{F}$.

We also introduce constraints on the expected opinions, which are empty at the first iteration and are updated during the iteration process. These constraints are called the key opinion reference, denoted by $\boldsymbol{k}$. In addition to the topic $\boldsymbol{F}$ and the key opinion reference $\boldsymbol{k}$, the natural language instruction $\boldsymbol{I}$ is responsible for organizing the task, namely, combining all the elements of prompt. We use different instructions for different iterations, as shown in Table 1. Then we adopt LLM with the prompt to generate the summary $\boldsymbol{s}$, formally:

$$\boldsymbol{s} = LLM(\boldsymbol{a}, \mathbb{D}) \tag{1}$$

3.3 Reward calculation

To calculate the reward of the current state, namely the generated summary $\boldsymbol{s}$, we introduce the notions of coverage score and diversity score.

The coverage score quantifies how much the summary $\boldsymbol{s}$ covers the opinions in source documents $\mathbb{D}$, denoted by $Cov(\boldsymbol{s}, \mathbb{D})$. For each sentence $s \in \boldsymbol{s}$ in the generated summary, we compute its

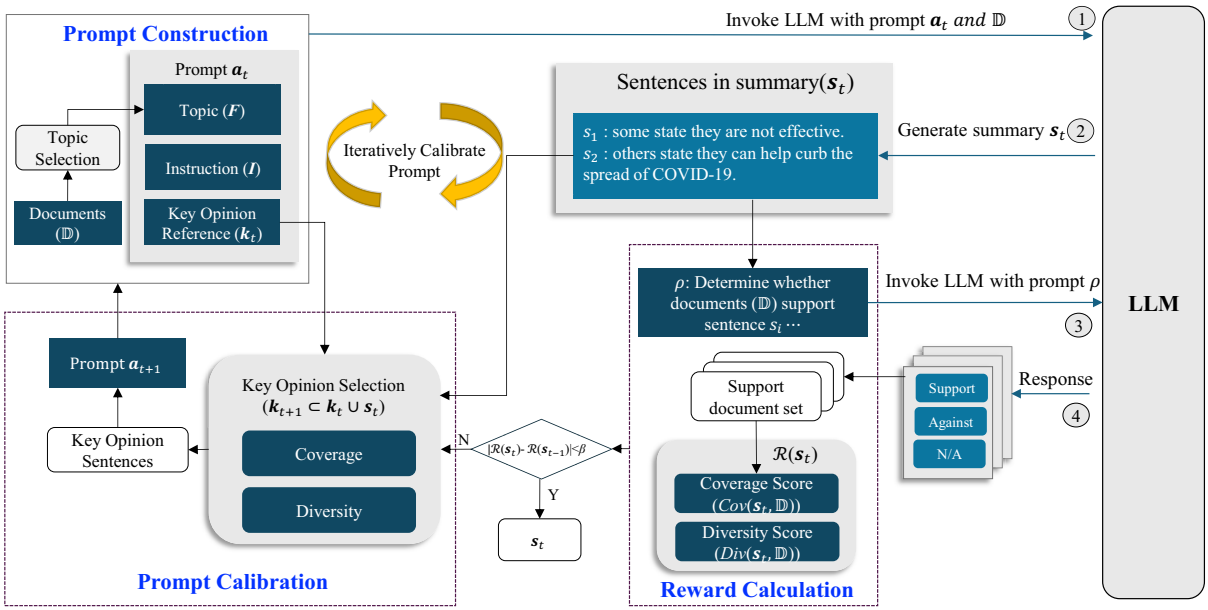

Figure 2. Self-evaluation based prompt calibration framework.

Table 1. Instructions for different iterations.

Iteration stage	The content of instructions.
The first iteration	With regard to the topic $\boldsymbol{F}$, summarize the following documents to generate a 30-words summary that contains different opinions.
Other iterations	With regard to the topic $\boldsymbol{F}$, summarize the following documents to generate a 30-words summary that contains different opinions. During the summarization process, refer to the sentences in $\boldsymbol{k}$ and explore new opinions.

support document set. That is to say, we check how much documents in $\mathbb{D}$ support the sentence s. Since LLMs store a vast amount of knowledge, they have high abilities on reasoning and are now used as evaluation tools for multiple tasks [19], we verify the support relation between document $d \in \mathbb{D}$ and sentence s by LLM with the prompt ρ= *"Determine whether '{d}' supports '{s}', using the word below: 'support', 'against' or 'not clearly' to answer."*. Let $\mathbb{I}(d, s) = 1$ denote LLM returning a positive answer, namely $LLM(\rho, d, s) =' support'$. Otherwise, $\mathbb{I}(d, s) = 0$. Then the coverage score $Cov(\boldsymbol{s}, \mathbb{D})$ is the percentage of the support document set with respect to the document set $\mathbb{D}$:

$$Cov(\boldsymbol{s}, \mathbb{D}) = \frac{|\cup_{s \in \boldsymbol{s}} \{d \in \mathbb{D} | \mathbb{I}(d, s) = 1\}|}{|\mathbb{D}|} \tag{2}$$

The diversity score quantifies the differences between the opinions contained in a generated summary $\boldsymbol{s}$. To compute the differences between two sentences $s_i, s_j \in \boldsymbol{s}$, we adopt the *Jaccard distance* on their support document sets so as to reflect the difference against $\mathbb{D}$. Formally,

$$\delta(s_i, s_j | \mathbb{D}) = 1 - \frac{|\{d \in \mathbb{D} | \mathbb{I}(d, s_i) = 1 \wedge \mathbb{I}(d, s_j) = 1\}|}{|\{d \in \mathbb{D} | \mathbb{I}(d, s_i) = 1 \vee \mathbb{I}(d, s_j) = 1\}|} \tag{3}$$

The diversity score of the generated summary $\boldsymbol{s}$ is computed as the expectation, i.e., the mean, of the distance between all the sentence in $\boldsymbol{s}$:

$$Div(\boldsymbol{s}, \mathbb{D}) = \mathbb{E}_{s_i, s_j \in \boldsymbol{s}}(\delta(s_i, s_j | \mathbb{D})) \tag{4}$$

Then the reward of the summary is computed by the following Equation 5, where function $f()$ is used to combine the coverage score and diversity score, here we choose the sum operation. Based on the reward, we can make a decision to either calibrate the prompt or to stop for accepting the current summary.

$$\mathcal{R}(\boldsymbol{s} | \boldsymbol{a}, \mathbb{D}) = f(Cov(\boldsymbol{s}, \mathbb{D}), Div(\boldsymbol{s}, \mathbb{D})) \tag{5}$$

3.4 Prompt calibration

Due to the blackbox of the LLM generation process, the semantics of generated summaries may drift away from the documents $\mathbb{D}$. To address this issue, we use an iterative strategy to calibrate the prompts for achieving the strong constraints. In particular, according to our empirical studies, we found that some opinions in the generated summary is the very important information source to guide LLM. So we construct a new prompt $\boldsymbol{a}_{t+1} = [\boldsymbol{F}, \boldsymbol{k}_{t+1}, \boldsymbol{I}]$ by choosing the key opinion sentences from the generated summary $\boldsymbol{s}_t$ and the key opinion reference $\boldsymbol{k}_t$ in prompt $\boldsymbol{a}_t$. Namely, we construct new $\boldsymbol{k}_{t+1}$ by selecting a subset of $\boldsymbol{Z}_t = \boldsymbol{k}_t \cup \boldsymbol{s}_t$. We design the following selection strategy in terms of the content coverage and sentence diversity:

1. For the candidate sentence set $\boldsymbol{Z}_t$, initialize an empty set $\boldsymbol{k}_{t+1}$.
2. Select a sentence z in $\boldsymbol{Z}_t$ with the largest support document set, i.e., the set $\{d \in \mathbb{D} | \mathbb{I}(d, z) = 1\}$. Then move the sentence from $\boldsymbol{Z}_t$ to $\boldsymbol{k}_{t+1}$.
3. Select a sentence z in $\boldsymbol{Z}_t$ that has the highest diversity score $\mathbb{E}_{h \in \boldsymbol{k}_{t+1}}(\delta(z, h | \mathbb{D}))$ with the sentences in $\boldsymbol{k}_{t+1}$. Then move the sentence from $\boldsymbol{Z}_t$ to $\boldsymbol{k}_{t+1}$. Repeat this step until the diversity score is less than a threshold α.

Setting the threshold α is important here for controlling LLM. Namely, when users focus on the highly differentiated opinions such as the opposing opinions, a large α is set, and a small α is set when users focus more on the diversity of opinions. The new prompt will be used for the next iteration. The iteration process stops when the reward difference between two generated summaries is less than threshold β or the predefined maximum number of iterations is reached.

4 Experiments

4.1 Datasets and metrics

We use the benchmark dataset Microblog Opinion Summarization (MOS) [5] for experiments. The MOS dataset contains tweets cover two years and has two subsets: *UK Election Opinionated Dataset* (**EO** for short) and *COVID-19 Opinionated Dataset* (**CO** for short). The **EO** contains 681 samples that have opinions about the topics on UK Elections. The **CO** contains 561 samples on the COVID-19 topics. Every sample includes multiple tweets and the corresponding summary. A summary is produced by experts and contains a majority opinion and multiple minority opinions. The statistical information of the two datasets is shown in Table 2. Due to the unsupervised setting, we only use the testing set.

Table 2. Dataset statistics

	EO		CO	
	training	testing	training	testing
Samples	631	50	511	50
Documents/Sample	30	31	32	34
Words/Summary	32	48	31	42

We use the *gram*-based evaluation metric ROUGE and the semantic metric BLEURT [22] to evaluate the similarity between the generated summaries and the reference summaries. ROUGE evaluates summaries by comparing their gram co-occurrences to the references. We compute the F1 of ROUGE-1, ROUGE-2, ROUGE-L and ROUGE-SU4 by *pyrouge*[1] with the following setup: ROUGE-1.5.5.pl-fA-a-c95-m-n2-24-u-p0.5. BLEURT adopts neural network to learn the semantic differences and is robust to different linguistic expressions. We also introduce a new metric that measures the opinion difference of summary in terms of the tendencies of sentences. Besides, we use SummaCzs [15] to evaluate the semantic consistency between the source documents and the summaries.

4.2 Implementation details and comparison methods

4.2.1 Implementation details

In the prompt construction process, we use NLTK[2] to extract noun phrases and verb phrases. We choose Vicuna-7B [8] as the base summarization and evaluation model. A generated summary may contains multiple different or even opposing opinions in a sentence, such as the summary *'Some argue that masks are not effective in preventing the spread of COVID-19, while others believe that they can help curb the spread'*. This makes it challenging to analyze and evaluate the opinions. To address this issue, we split the summary into multiple small sentences according to the conjunctions, transitions, and periods. We set the maximum number of iterations to 4, the diversity score threshold α to 0.6 and the reward coverage threshold β to 0.02. In order to reduce the computational overhead, for the candidate sentences in Z_t, we first filter out the sentences with coverage score below 0.3 before selecting the key opinion reference. To avoid experimental bias, we repeat the experiment 3 times for each setting and use the mean as the final results. [3]

4.2.2 Comparison methods

We compare our method with some representative methods: **LexRank** [10] constructs a weighed connectivity graph, where the sentences are represented as nodes, and the similarity between sentences serves as the edge weights. Then the PageRank algorithm is used to identify important sentences. **QT** [3] adopts Vector-Quantized Variational Autoencoders [23] to obtain the quantized sentence vectors, clusters similar sentences together, quantifies the popularity of the clustering results, and then extracts representative sentences from the most popular ones. **SummPip** [27] constructs a sentence graph based on both the lexical and the semantic relations between sentences, and uses graph clustering to get some sub-graphs. Then the summary is obtained by compressing the sub-graphs into multi-sentences. **Copycat** [6] uses Variational Autoencoder [14] to model the process of generating new opinions from multiple related opinions. **OPINESUM** [20] constructs pseudo-summaries by the application of textual entailment. We use LongT5 [11] as the base model of OPINESUM. **Vicuna-7B** [8] is a moderate language model trained by fine-tuning LLaMA on the user-shared conversations collected from ShareGPT [25]. **GPT-3.5** [7] and **GPT-4** [4] are the large language models, capable of understanding and generating human-like text across a wide range of tasks. **BART** [16] is the widely recognized pre-trained model that has been proven to perform well on multiple summarization tasks. We fine tune BART separately on the 50% and 100% training data in MOS.

4.3 Main results

The comparison results are shown in Table 3. The R-1, R-2, R-L and R-SU4 stand for the F1 of ROUGE-1, ROUGE-2, ROUGE-L and ROUGE-SU4, respectively. Compared with the unsupervised methods, including large-scale language models like GPT-4 and GPT-3.5, our method achieves the best results across all metrics, especially on the EO dataset. Compared to Vicuna, which is the base model of our method, we achieve almost **10%** improvement on the model performance solely relying on the prompt calibration, which shows the strong effectiveness of the framework. We also see that all methods get weaker results on the CO dataset than on the EO dataset. This is because the opinions in the CO dataset tend to employ humor or irony, such as the opinion *'coronovirus: "hold my beer!"'*, which makes it difficult for the model to understand these opinions. Unsurprisingly, the supervised methods outperform the unsupervised methods, indicating the importance of the supervised data. However, the model performance does not continue to improve as the amount of training data increases. Compared to the BART trained on 50% of the training data, BART trained on the entire training dataset achieves a slight advantage on the EO dataset, but performs weaker on the CO dataset. These results are consistent with the findings in paper [5]. We think this is due to the inconsistency between the distribution of the additional added training data and the testing data.

[1] https://github.com/bheinzerling/pyrouge
[2] https://github.com/nltk/nltk
[3] Code is available at https://github.com/wangjian026/CPSum
[4] https://openai.com/gpt-4

Table 3. Model comparison results. BART(50%) and BART(100%) denote the size of the training set used to train BART, respectively.

Model type	Method	Election Opinionated Data(EO)					CoVID-19 Opinionated Data(CO)				
		R-1	R-2	R-L	R-SU4	BLEURT	R-1	R-2	R-L	R-SU4	BLEURT
Unsupervised Extractive	LexRank*	14.27	1.15	9.62	–	-.418	16.41	1.48	10.89	–	-.560
	QT*	14.78	1.08	9.45	–	-.468	14.23	1.03	9.55	–	-.621
Unsupervised Abstractive	SummPip*	13.05	1.15	8.90	–	-.409	12.96	1.37	9.32	–	-.488
	Copycat*	14.05	1.56	10.25	–	-0.503	12.47	1.31	9.41	–	-.552
	OPINESUM	31.58	4.58	23.79	9.03	0.338	27.88	4.42	21.45	7.81	0.306
	Vicuna	32.05	9.54	23.51	12	0.448	27.74	7.56	21.01	10.03	0.432
	GPT-3.5-turbo	31.82	8.59	25.56	10.78	0.262	29.17	7.80	23.64	10.15	0.297
	GPT-4	32.28	9.16	25.17	11.49	0.465	28.39	7.94	22.92	9.93	0.427
	CPSum	**33.56**	**10.82**	**27.78**	**13.00**	**0.467**	**29.81**	**9.67**	**24.57**	**11.47**	**0.441**
Supervised Abstractive	BART(50%)	38.14	12.54	29.37	15.19	0.487	34.68	12.29	28.42	13.98	0.439
	BART(100%)	38.33	12.48	29.49	15.18	0.477	33.88	10.73	27.22	13.06	0.434

The results marked with '*' are taken from the paper [5].

4.4 Opinion difference evaluation

A good summary should contains multiple opinions that reflect the concerns of different groups, such as the summary *'The coronavirus is real and can be deadly. Some people believe it is controlled or a hoax.'*, where the first sentence tends to express the majority opinion, while the second tends to a minority opinion. This provides us with an alternative perspective for measuring the opinion difference.

According to the annotation in MOS dataset, we divide the sentences in a reference summary into two sentence sets $\boldsymbol{O}_{maj}$ and $\boldsymbol{O}_{min}$, where $\boldsymbol{O}_{maj}$ contains the sentences that express the majority opinions and $\boldsymbol{O}_{min}$ contains the sentences that express the minority opinions. For each sentence s in the generated summary $\boldsymbol{s}$, we choose a sentence r in the reference summary that is semantically closest to s, denoted as $r = argmax_{r \in \boldsymbol{O}_{maj} \cup \boldsymbol{O}_{min}} Sim(s, r)$, where $Sim()$ is the cosine similarity between the embedding vectors of two sentences. We use the pretrained all-MiniLM-L6-v2 model[5] to obtain the sentence embeddings. Let $\mathbb{I}^{'}(r)$ denote a function to check whether r is in $\boldsymbol{O}_{maj}$ or $\boldsymbol{O}_{min}$, if $r \in \boldsymbol{O}_{maj}$, then $\mathbb{I}^{'}(r) = 1$, otherwise $\mathbb{I}^{'}(r) = -1$. Then we define the tendency $w(s)$ of a sentence s as follows:

$$w(s) = \mathbb{I}^{'}(r) * Sim(s, r) \tag{6}$$

A positive $w(s)$ indicates that the sentence tends toward the majority opinions, while a negative one means the sentence tends toward the minority opinions. It should be noted that a sentence in a summary often contains multiple opposing opinions or different aspects, which affects the similarity calculation. Therefore, we split the summary into multiple shorter sentences according to the conjunctions, transitions, and periods. In a summary $\boldsymbol{s}$, the greater the difference of tendencies between these sentences, the greater the difference in opinions. We calculate the significance difference $Dif_{sig}(\boldsymbol{s})$ and the group difference $Dif_{gro}(\boldsymbol{s})$ to demonstrate the differences between opinions within a summary:

$$Dif_{sig}(\boldsymbol{s}) = Max_{s \in \boldsymbol{s}}\{w(s)\} - Min_{s \in \boldsymbol{s}}\{w(s)\} \tag{7}$$

$$Dif_{gro}(\boldsymbol{s}) = Mean_{s \in \boldsymbol{s}, w(s)>0}\{w(s)\} - Mean_{s \in \boldsymbol{s}, w(s)<0}\{w(s)\} \tag{8}$$

where, Max, Min and $Mean$ represent the maximum, minimum, and mean values, respectively.

We select all samples in MOS that contain both the majority and minority opinions for testing. The results in Table 4 show that the summaries generated by CPSum have higher opinion difference than those generated by other methods.

[5] https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2

Table 4. Evaluation results on opinion difference and faithfulness

Method	Opinion Difference[0,2]		Faithfulness
	Significance difference	Group difference	[0,1]
OPINESUM	0.84	0.66	0.30
Vicuna	0.92	0.74	0.38
GPT-4	0.90	0.76	0.39
CPSum	**0.96**	**0.78**	**0.42**

To explore the details of the sentence differences, we show the tendency of each sentence in summaries through the scatters in Fig 3, where the x-axis denotes the summary ID, and the y-axis denotes the value of tendency. Each point in the scatter represents a sentence in summaries. The Golden method refers to the reference summaries, which serves as the golden performance of the generated results. By comparing the tendency differences of sentences in the same summary, we find that compared to GPT-4, Vicuna, and OPINESUM, the sentences generated by CPSum are scattered across the upper and lower positions of the scatter plot, indicating that they are more distinctiveness and closer to the reference summaries.

4.5 Faithfulness evaluation based on NLI model.

In our summarization framework, we use LLM to verify whether a document supports a sentence in summary. This can reflect the faithfulness of the generated summaries. In addition to using LLM, we also introduce a supervised model SummaCzs [15] to verify the faithfulness of the generated summaries to the source documents. SummaCzs evaluates faithfulness by computing the entailment scores between the documents and each sentence in the summary. The higher the score, the greater the consistency between the generated summary and the source documents. We perform experiments on the testing data in MOS. Results in Table 4 show that our method achieves the highest faithfulness.

4.6 Human evaluation

There are 100 testing samples in the MOS dataset, we randomly select 30 for human evaluation. We provide the source documents, the reference summaries, and the generated summaries to 5 annotators who are highly educated. Annotators are asked to rank each summary from the highest to the lowest in the following dimensions:

- Sentential Coherence(SC) -Sentences in summary should be semantically related to each other and not contain grammatical errors.
- Non-redundancy(NR) -The summaries should not contain duplicate contents.

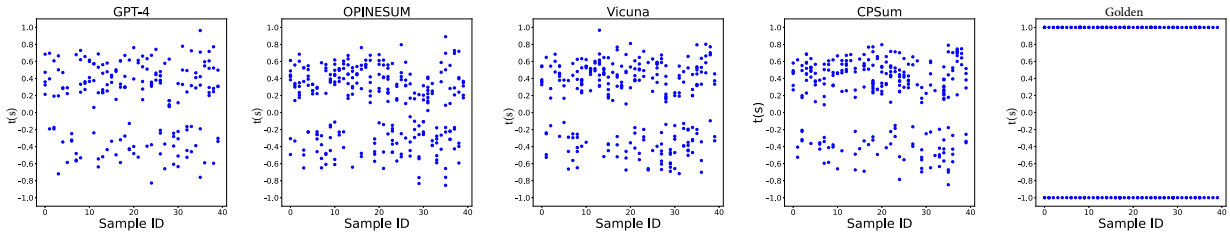

Figure 3. Method comparison on the sentence tendencies in summaries.

Table 5. Human evaluation results

Method	Model Size	SC	NR	OS	DO
OPINESUM	770M	0.03	0.03	0.04	0.05
Vicuna	7B	0.27	0.19	0.27	0.30
GPT-4	>175B	0.32	**0.45**	**0.35**	0.32
CPSum	7B	**0.38**	0.33	0.34	**0.33**

- Opinion similarity(OS) -The summaries should be similar to the reference summaries in opinions.
- Diverse opinion(DO) -The summaries should contain diverse opinions.

We use *Krippendorff's alpha coefficient* to measure the inter-annotator agreements. The coefficient values are 0.116 for sentential coherence, 0.559 for non-redundancy, 0.294 for opinion similarity, 0.151 for diverse opinion, which correspond to the *'slight'*, *'moderate'*,*'fair'*,*'slight'* agreement, respectively[24]. The slight agreement on the DO dimension reflects the inherent subjectivity of judgments about diverse opinions. We analyze the generated summaries, and find that: a) Summaries with opposing opinions lead to high consistency between annotators, while summaries that contain different aspects of the same opinion result in low consistency. b) Summaries that include inflections such as *"some people ..... and others..... "* are considered to have high diversity, whereas other statements conveying the same meaning reduce the annotators' confidence.

For each method, we calculate the percentage of the summaries ranked at the top. The results in Table 5 show that our method exhibits the best results in terms of the sentential coherence and opinion diversity. For the opinion similarity, our method with small parameters (7B) is slightly lower than GPT-4 (higher than 175B). Additionally, by analyzing the generated summaries by OPINESUM, we find some semantically incoherent sentences. These sentences affect the human evaluation results across multiple dimensions.

5 Model analysis

5.1 The ablation studies

We evaluate the effects of the topic selection and the iterative prompt calibration on the model performance. The results in Table 6 indicate that all components play important roles. The calibration component, in particular, has the strongest impact, especially on the ROUGE-L. Besides, we find that the combination of topic and prompt calibration benefits model performance on most metrics, although this combination slightly affects the model's performance on some specific metrics, such as the ROUGE-L on the CO dataset.

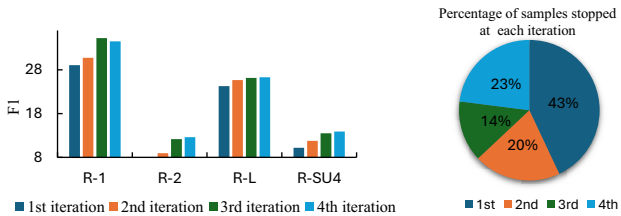

Figure 4. The model performance of each iteration.

5.2 Analysis of the prompt calibration

To understand the importance of the prompt calibration process at each iteration in detail, we analyze the generated summaries at each iteration from both the statistical and instance perspectives. Fig 4 shows the ROUGE scores corresponding to different numbers of iterations, while the pie chart is the proportion of samples that reach the stop iteration condition at each iteration. We find that the quality of the generated summaries gradually improves as the iteration increases, indicating the effectiveness of the iteration process. We also provide an instance to visualize the effects of the prompt calibration. We show the key opinion references and the generated summaries in Table 7. We find that LLM maintains its own main contents while taking useful information from the key opinion reference. For example, the sentence *'This contradicts claims that mask-wearing increases the risk of contracting COVID-19'* in the key opinion reference inspires LLM to focus on the *'contradicts'* and to generate opposing opinions *'study does not conclude that all masks are ineffective'*. Besides, compared to the first iteration, the final summary contains more diverse opinions such as the opinion *'wearing surgical and cloth masks can increase the risk of getting sick'*.

5.3 Hyper-parameter analysis

We check the hyper-parameter settings. α is used to balance the consensus and diversity of opinions in the generated summary. Large α indicates that we are focus on a small number of opinions with high differences. As shown in Fig 5, the model performance increases gradually with α and then shows a decrease. Additionally, we find that the performance of our model with different values of α consistently outperforms the comparison methods across all metrics.

5.4 The correlation between the key opinion reference and the generated summary

Since the key opinion reference serves as the primary information for guiding LLM, to validate the correlation between the key opinion references and the generated summaries, we measure their quality using

Table 6. Ablation experiment results

Method	Election Opinionated Data(EO)					CoVID-19 Opinionated Data(CO)				
	R-1	R-2	R-L	R-SU4	BLEURT	R-1	R-2	R-L	R-SU4	BLEURT
CPSum	33.56	10.82	27.78	13.00	0.456	29.81	9.67	24.57	11.47	0.439
W/O calibration	-1.25	-1.32	-3.04	-1.15	-0.011	-1.72	-1.75	-2.61	-1.28	-0.009
W/O topic	-0.57	-0.49	-1.18	-0.87	-0.01	-0.08	-0.35	-0.045	-0.09	-0.005
W/O topic and calibration	-1.51	-1.27	-4.25	-1.08	-0.016	-2.0	-2.07	-2.10	-3.56	-0.007

Table 7. Case study. Text marked with the same color indicates similar semantics.

Reference summary			
CDC study finds that the overwhelming majority of people contracting Covid-19 wore masks. The majority believe the study vindicates their argument that people shouldn't be made to wear masks. A minority believe the CDC's statements are being misinterpreted by anti-maskers.			
Round	k_t	**Generated summary based on k_t**	**Stop**
1	$\emptyset$	The CDC conducted a study on people who tested positive for COVID-19 in July, finding that 70% reported always wearing a mask for the preceding 14 days. However, this study has been widely interpreted differently, with some suggesting that masks are ineffective and others arguing that they are essential in preventing the spread of the virus. The CDC itself has recommended wearing masks to prevent the spread of the virus.	N
2	['The CDC itself has recommended wearing masks to prevent the spread of the virus.']	'New study by 11 medical institutions analyzed people who tested positive for COVID-19 in July and found that 70% of them "always" wore a mask for the preceding 14 days, contradicting earlier claims that masks are ineffective in preventing the spread of the virus. The CDC has recommended wearing masks to prevent the spread of the virus.	N
3	['The CDC has recommended wearing masks to prevent the spread of the virus.']	'A new study by eleven medical institutions found that 70% of people who tested positive for COVID-19 in July reported always wearing a mask for the preceding 14 days. This contradicts claims that mask-wearing increases the risk of contracting COVID-19. The CDC originally recommended against wearing masks, but later reversed its stance and now recommends masks to prevent the spread of the virus.'	N
4	['The CDC has recommended wearing masks to prevent the spread of the virus.', 'This contradicts claims that mask-wearing increases the risk of contracting COVID-19']	'A new study conducted by 11 medical institutions analyzed a group of people who tested positive for COVID-19 in July. 70% of them reported that they had "always" worn a mask for the preceding 14 days. However, some sources claim that the CDC study shows that mask wearers are more likely to catch COVID-19. Others point out that the study does not conclude that all masks are ineffective. China Joe has a higher chance of catching the virus wearing a mask than not wearing one. Some sources suggest that wearing surgical and cloth masks can increase the risk of getting sick. The CDC has recommended wearing masks to prevent the spread of the virus.'	Y

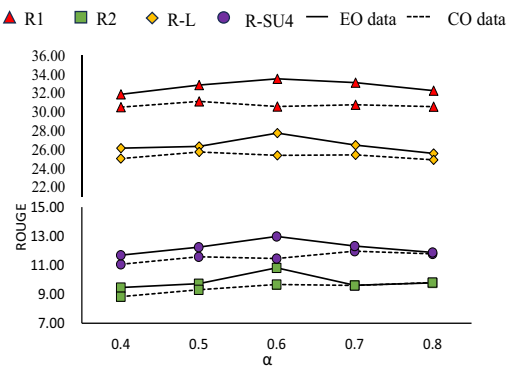

Figure 5. Effects of α on the model performance

equation 5, and analyze their correlation using *Spearman coefficient* and *Kendall's tau rank correlation* coefficients.

We select all key opinion reference and their corresponding summaries throughout the entire iterative process for experimentation. The results show that the values of *Spearman coefficient* and *Kendall's tau rank correlation* are 0.363 and 0.258 with p-value < 0.005. This indicates a positive correlation between them, which further demonstrates the usefulness of prompt calibration for the generating results.

6 Conclusions

In order to guide LLM for generating summaries with diverse opinions while constraining the semantic drift of summaries, we propose the iterative prompt calibration framework. We use LLM to generate summary and evaluate whether the source documents support the sentences in summary, then we select sentences by considering their opinion coverage and opinion difference to calibrate the prompts. To validate the effects of our framework, we measure the opinion difference in generated summary based on the tendencies of sentences, we also use multiple metrics to evaluate the semantic similarity between the generated summary and reference summary. Experimental results show that our method achieves state-of-the-art results. For the future work, a lightweight textual entailment model can be incorporated to assist in determining the support relationship between documents and sentences, so as to reduce the computational overhead.

Acknowledgements

This work was supported by the Key R&D Program of Shandong Province (2023CXGC010801), the National Natural Science Foundation of China (62376138) and the Innovative Development Joint Fund Key Projects of Shandong NSF (ZR2022LZH007).

References

[1] G. Adams, A. Fabbri, F. Ladhak, E. Lehman, and N. Elhadad. From sparse to dense: GPT-4 summarization with chain of density prompting. In Y. Dong, W. Xiao, L. Wang, F. Liu, and G. Carenini, editors, *Proceedings of the 4th New Frontiers in Summarization Workshop*, pages 68–74, Singapore, Dec. 2023.

[2] R. K. Amplayo and M. Lapata. Unsupervised opinion summarization with noising and denoising. In D. Jurafsky, J. Chai, N. Schluter, and J. Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1934–1945, Online, July 2020.

[3] S. Angelidis, R. K. Amplayo, Y. Suhara, X. Wang, and M. Lapata. Extractive Opinion Summarization in Quantized Transformer Spaces. *Transactions of the Association for Computational Linguistics*, 9:277–293, 03 2021. ISSN 2307-387X.

[4] A. Bhaskar, A. Fabbri, and G. Durrett. Prompted opinion summarization with GPT-3.5. In A. Rogers, J. Boyd-Graber, and N. Okazaki, editors, *Findings of the Association for Computational Linguistics: ACL 2023*, pages 9282–9300, Toronto, Canada, July 2023.

[5] I. M. Bilal, B. Wang, A. Tsakalidis, D. Nguyen, R. Procter, and M. Liakata. Template-based abstractive microblog opinion summarization. *Transactions of the Association for Computational Linguistics*, 10: 1229–1248, 2022.

[6] A. Bražinskas, M. Lapata, and I. Titov. Unsupervised opinion summarization as copycat-review generation. In D. Jurafsky, J. Chai, N. Schluter, and J. Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5151–5169, Online, July 2020.

[7] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. Language models are few-shot learners, 2020.

[8] W.-L. Chiang, Z. Li, Z. Lin, Y. Sheng, Z. Wu, H. Zhang, L. Zheng, S. Zhuang, Y. Zhuang, J. E. Gonzalez, I. Stoica, and E. P. Xing. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, March 2023.

[9] D. Deutsch, T. Bedrax-Weiss, and D. Roth. Towards Question-Answering as an Automatic Metric for Evaluating the Content Quality of a Summary. *Transactions of the Association for Computational Linguistics*, 9:774–789, 08 2021. ISSN 2307-387X.

[10] G. Erkan and D. R. Radev. Lexrank: graph-based lexical centrality as salience in text summarization. *J. Artif. Int. Res.*, 22(1):457–479, dec 2004. ISSN 1076-9757.

[11] M. Guo, J. Ainslie, D. Uthus, S. Ontanon, J. Ni, Y.-H. Sung, and Y. Yang. LongT5: Efficient text-to-text transformer for long sequences. In M. Carpuat, M.-C. de Marneffe, and I. V. Meza Ruiz, editors, *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 724–736, Seattle, United States, July 2022.

[12] A. Hallak, D. D. Castro, and S. Mannor. Contextual markov decision processes. *ArXiv*, abs/1502.02259, 2015.

[13] W. Ke, J. Gao, H. Shen, and X. Cheng. Consistsum: Unsupervised opinion summarization with the consistency of aspect, sentiment and semantic. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, WSDM '22, page 467–475, New York, NY, USA, 2022.

[14] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *CoRR*, abs/1312.6114, 2013.

[15] P. Laban, T. Schnabel, P. N. Bennett, and M. A. Hearst. Summac: Revisiting nli-based models for inconsistency detection in summarization. *Transactions of the Association for Computational Linguistics*, 10:163–177, 2022.

[16] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In D. Jurafsky, J. Chai, N. Schluter, and J. Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online, July 2020.

[17] T. Liu, Y. Sun, J. Wu, X. Xu, Y. Han, C. Li, and B. Gong. Unsupervised paraphrasing under syntax knowledge. In *AAAI Conference on Artificial Intelligence*, 2023.

[18] X. Liu, H. Lai, H. Yu, Y. Xu, A. Zeng, Z. Du, P. Zhang, Y. Dong, and J. Tang. Webglm: Towards an efficient web-enhanced question answering system with human preferences. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '23, page 4549–4560, New York, NY, USA, 2023.

[19] Y. Liu, A. R. Fabbri, J. Chen, Y. Zhao, S. Han, S. R. Joty, P. Liu, D. R. Radev, C.-S. Wu, and A. Cohan. Benchmarking generation and evaluation capabilities of large language models for instruction controllable summarization. In *Findings of the Association for Computational Linguistics*, pages 4481–4501, 2024.

[20] A. Louis and J. Maynez. OpineSum: Entailment-based self-training for abstractive opinion summarization. In A. Rogers, J. Boyd-Graber, and N. Okazaki, editors, *Findings of the Association for Computational Linguistics: ACL 2023*, pages 10774–10790, Toronto, Canada, July 2023.

[21] M. L. Reinald Kim Amplayo, Stefanos Angelidis. Aspect-controllable opinion summarization. In *Conference on Empirical Methods in Natural Language Processing*, 2021.

[22] T. Sellam, D. Das, and A. Parikh. BLEURT: Learning robust metrics for text generation. In D. Jurafsky, J. Chai, N. Schluter, and J. Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892, Online, July 2020.

[23] A. van den Oord, O. Vinyals, and K. Kavukcuoglu. Neural discrete representation learning. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6309–6318, Red Hook, NY, USA, 2017.

[24] A. Wang, K. Cho, and M. Lewis. Asking and answering questions to evaluate the factual consistency of summaries. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5008–5020, Online, July 2020.

[25] G. Wang, S. Cheng, X. Zhan, X. Li, S. Song, and Y. Liu. Openchat: Advancing open-source language models with mixed-quality data. *arXiv preprint arXiv:2309.11235*, 2023.

[26] Y. Wang, Z. Zhang, and R. Wang. Element-aware summarization with large language models: Expert-aligned evaluation and chain-of-thought method. In A. Rogers, J. Boyd-Graber, and N. Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8640–8665, Toronto, Canada, July 2023.

[27] J. Zhao, M. Liu, L. Gao, Y. Jin, L. Du, H. Zhao, H. Zhang, and G. Haffari. Summpip: Unsupervised multi-document summarization with sentence graph compression. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '20, page 1949–1952, New York, NY, USA, 2020.