Research Article

# LFOSum: Summarizing Long-form Opinions with Large Language Models

**Mir Tafseer Nayeem**[1], **Davood Rafiei**[1]

1. University of Alberta, Canada

**Online reviews play a pivotal role in influencing consumer decisions across various domains, from purchasing products to selecting hotels or restaurants. However, the sheer volume of reviews—often containing repetitive or irrelevant content—leads to information overload, making it challenging for users to extract meaningful insights. Traditional opinion summarization models face challenges in handling long inputs and large volumes of reviews, while newer Large Language Model (LLM) approaches often fail to generate accurate and faithful summaries. To address those challenges, this paper introduces (1) a new dataset of long-form user reviews, each entity comprising over a thousand reviews, (2) two training-free LLM-based summarization approaches that scale to long inputs, and (3) automatic evaluation metrics. Our dataset of user reviews is paired with in-depth and unbiased critical summaries by domain experts, serving as a reference for evaluation. Additionally, our novel reference-free evaluation metrics provide a more granular, context-sensitive assessment of summary faithfulness. We benchmark several open-source and closed-source LLMs using our methods. Our evaluation reveals that LLMs still face challenges in balancing sentiment and format adherence in long-form summaries, though open-source models can narrow the gap when relevant information is retrieved in a focused manner**[1]

**Corresponding authors:** Mir Tafseer Nayeem, mnayeem@ualberta.ca; Davood Rafiei, drafiei@ualberta.ca

## 1. Introduction

Online opinions play a critical role in shaping consumer decisions about what products to buy, where to stay, where to eat, and even which books to read. A recent survey found that approximately 98% of online customers read reviews before making a purchase decision[1]. These reviews reflect user opinions, providing valuable insights that help set realistic expectations and reveal key details about products and

doi.org/10.32388/D1MVB5

services. However, popular products often accumulate hundreds or even thousands of reviews, many of which contain uninformative content, such as irrelevant personal anecdotes, making them overwhelming to sift through. This leads to information overload[2], where the sheer volume of reviews discourages consumers, sometimes disregarding the reviews at all[3]. Market research shows that most customers read fewer than 10 reviews before making a purchase[4], and this can lead to suboptimal decision-making[5]. The sheer volume, variable quality, and limited consumer patience underscore the need for improved review utilization strategies to mitigate information overload and enhance decision-making.

Review summarization has been studied in the literature under the same name[6] and within the broader field of opinion mining and summarization[7][8], with the goal of producing a concise and easy-to-read summaries about target entities (e.g., a product, hotel, restaurant, or service). A well-constructed summary is expected to capture the most common or popular viewpoints while omitting unnecessary or irrelevant information[9][10]. A key challenge is the scarcity of annotated datasets that pair reviews with summaries. Most review platforms do not provide summaries, and creating them would require costly human annotation, unlike news summarization datasets[11][12][13], where summaries are often included in the source documents. To address this, existing studies have leveraged self-supervised approaches, generating synthetic pairs from review corpora[14][15], typically by designating one review as a pseudo-summary of others. However, most of these datasets are limited to a maximum of 10 reviews[16][17][18], with only a few extending to hundreds[19][20], while real-world entities often accumulate thousands of reviews. Our work aims to scale review summarization to accommodate larger volumes of reviews.

An effective opinion summarization model should possess several desirable properties to address the challenges associated with large-scale review summarization[21]. First, it should offer **control mechanisms**[22][23], enabling users to customize the summaries to their specific needs. Second, the model must be **scalable**, capable of processing thousands of user opinions while efficiently extracting essential information[10]. Lastly, the generated summaries must be **faithful** to the input texts, accurately representing their content while minimizing the risk of hallucination[24][25].

In this paper, we explore three control mechanisms for opinion summarization: **(1)** query control, **(2)** sentiment control, and **(3)** length control. With query control, users can specify preferences such as *'ocean view'* or proximity to a *'metro station.'* Sentiment control enables structuring summaries into sections like 'PROS' and 'CONS', while length control allows users to dictate the length of the generated summaries. To handle large volumes of reviews, we examine two scalable approaches: Retrieval-Augmented Generation (RAG) and long-context Large Language Models (LLMs)[26], both of which show promise[27]. Evaluating faithfulness in long-form summarization poses a unique challenge[28], as modern models often suffer from hallucinations[24][25]. Traditional metrics like RAGAs[29] and RAGChecker[30] are typically designed for factual tasks such as question answering or knowledge-based generation, where sentiment and opinions are secondary concerns. To better align generated summaries with input texts, we treat both as sets of triplets and develop a scheme to quantify their alignment. This approach offers a reference-free evaluation metric tailored to sentiment-rich domains, such as product and service reviews, where opinion and sentiment polarity are crucial.

Our main contributions are summarized as follows:

- We introduce a new dataset of long-form user reviews, where each entity contains over a thousand reviews paired with in-depth, unbiased critical summaries provided by domain experts (§2).
- We propose training-free methods that utilize RAG and long-context LLMs to address the challenges of long-form opinion summarization. Our approach enables controllable and scalable summarization, providing fine-grained user controls (§3).
- We develop three novel, reference-free automatic evaluation metrics based on Aspect-Opinion-Sentiment (AOS) triplets. These metrics provide a granular and context-sensitive assessment of the faithfulness of generated summaries, particularly in sentiment-rich domains where opinions and sentiment polarity are crucial (§3.2.4).

| Datasets | Domain | #Entities | #Reviews | #Sents | #Words | #Tokens | Book Length? | Controls | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | **Sentiment** | **Length** |
| **MeanSum**[17] | Businesses | 200 | 8 | 41.1 | 542.76 | 561.01 | ✗ | ✗ | ✗ |
| **CopyCat**[31] | Products | 60 | 8 | 30.38 | 447.63 | 463.62 | ✗ | ✗ | ✗ |
| **FewSum**[18] | Businesses | 60 | 8 | 29.85 | 443.6 | 457.05 | ✗ | ✗ | ✗ |
| **OpoSum+**[22] | Products | 60 | 10 | 71.8 | 1,138.4 | 1,194.0 | ✗ | ✗ | ✗ |
| **SPACE**[19] | Hotels | 50 | 100 | 910.58 | 16,160.74 | 16,770.18 | ✗ | ✗ | ✗ |
| **AmaSum**[20] | Products | 3,166 | 322.31 | 1,057.3 | 15,232.26 | 15,614.71 | ✗ | ✓ | ✗ |
| **LFOSum** (ours) | Hotels | 500 | 1.5K | 10.5K | 196K | 207K | ✓ | ✓ | ✓ |

**Table 1.** Comparison of our LFOSum dataset with existing alternatives, focusing on long-form, book-length inputs (>100K tokens) and control dimensions. #Entities refers to the number of entities per dataset, while #Reviews indicates the average number of reviews per entity. #Sents represents the average number of sentences per entity, and #Words and #Tokens denote the average number of words and tokens (using the GPT-4o tokenizer) per entity.

## 2. Dataset Construction

We introduce the LFOSum dataset, a collection of long-form user reviews centered around hotel experiences shared online. Rich in detailed descriptions and personal opinions, this dataset is well-suited for opinion summarization tasks. Hotel reviews are particularly valuable due to their in-depth, personalized narratives that cover a wide range of user experiences, such as amenities, service quality, and location. Each entity in the dataset contains over a thousand reviews, offering a substantial volume of input texts.

### *Source Reviews*

The reviews were sourced from TripAdvisor[2], a widely-used platform that combines user-generated reviews with online travel booking services. TripAdvisor's reviews, on average, are three times longer than those found on other leading travel platforms[32], making it an ideal resource for exploring the challenges of long-form summarization with book-length inputs (exceeding 100K tokens)[33].

### Reference Summaries

Annotated datasets that pair summaries with long-form reviews are scarce, largely because such summaries are not readily available on most review platforms and require significant human annotation effort. To address this gap, we utilized Oyster[3], a platform specializing in professional hotel reviews. Oyster's reviews are based on first-hand, in-depth evaluations conducted by expert reviewers, making them a reliable and unbiased source for generating gold-standard summaries. Each review on Oyster is carefully crafted, providing critical assessments that are consistent and trustworthy. The summaries are divided into structured sections, highlighting key aspects of the accommodation, with explicit divisions into 'PROS' and 'CONS'.

### Data Pairing and Crawling Process

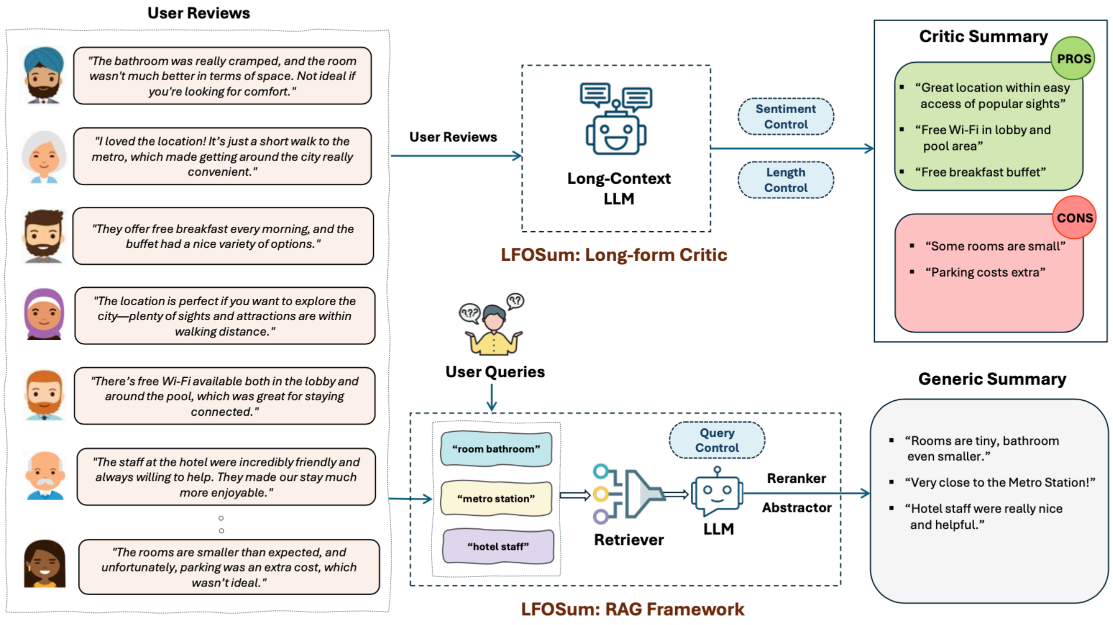
To construct pairs of input reviews and their corresponding summaries, we identified 500 travel destinations from the Oyster platform. For each entity, we collected the overview section from Oyster, which contains the critical summaries structured into 'PROS' and 'CONS'. Next, we searched for the same entities on TripAdvisor. In some cases, multiple entities had the same name; to disambiguate, we used unique identifiers such as the hotel's address and postal code. Once we established the correct entity matches, we crawled the relevant user reviews and corresponding summaries to create the dataset (sample in Appendix [Figure 2]).

### Comparison with Existing Datasets

We compare our proposed LFOSum dataset with existing human-referenced datasets used for evaluating opinion summarization models. As shown in Table 1, our dataset uniquely features book-length input reviews and supports both sentiment and length control. Although AmaSum[20] contains more than three times the number of reviews as SPACE[19], it has fewer tokens overall due to domain differences as hotel reviews tend to be longer and more detailed. Detailed statistics and preprocessing steps can be found in Appendix (Section D & Table 5).

# 3. Methodology

We propose two scalable, training-free methods to handle large volumes of user reviews effectively. First, the **Long-form Critic** method directly utilizes long-context LLMs to generate summaries, allowing users to control aspects such as sentiment and length (§3.1). Second, the **RAG Framework** combines an extractive-generative approach, managing long sequences by incorporating retrieval augmentation (§3.2).



User Reviews
"The bathroom was really cramped, and the room wasn't much better in terms of space. Not ideal if you're looking for comfort."
"I loved the location! It's just a short walk to the metro, which made getting around the city really convenient."
"They offer free breakfast every morning, and the buffet had a nice variety of options."
"The location is perfect if you want to explore the city—plenty of sights and attractions are within walking distance."
"There's free Wi-Fi available both in the lobby and around the pool, which was great for staying connected."
"The staff at the hotel were incredibly friendly and always willing to help. They made our stay much more enjoyable."
"The rooms are smaller than expected, and unfortunately, parking was an extra cost, which wasn't ideal."
User Reviews
Long-Context LLM
Sentiment Control
Length Control
LFOSum: Long-form Critic
Critic Summary
PROS
"Great location within easy access of popular sights"
"Free Wi-Fi in lobby and pool area"
"Free breakfast buffet"
CONS
"Some rooms are small"
"Parking costs extra"
User Queries
"room bathroom"
"metro station"
"hotel staff"
Retriever
Query Control
LLM
Reranker
Abstractor
LFOSum: RAG Framework
Generic Summary
"Rooms are tiny, bathroom even smaller."
"Very close to the Metro Station!"
"Hotel staff were really nice and helpful."

**Figure 1.** Our LFOSum framework includes two methods: (1) Long-form Critic, which uses long-context LLMs to generate critic summaries with user controls for sentiment and length (§3.1), and (2) the RAG Framework, which combines retrieval augmentation with LLMs to handle long-form user reviews and produce summaries (§3.2).

## *3.1. LFOSum: Long-form Critic*

In this approach, we generate critical summaries consisting of 'PROS' and 'CONS' from the full set of user reviews for a specific entity, presented in a long-form setting. To achieve this, long-context LLMs are employed to process the entire review corpus and generate critical summaries. The LLMs are prompted with a detailed task description, all user reviews for the entity, specific constraints, stylistic exemplars, and are instructed to produce the output in a structured JSON format with separate keys for 'PROS' and

‘CONS’ (the prompt presented in Figure 3 of Appendix). In the basic setting, we do not control the length; the model independently determines the optimal number of ‘PROS’ and ‘CONS’ sentences based on the input. The overall process can be formalized as:

$$\text{Critical Summary} = \text{LLM}_{\text{critic}}(R, \mathcal{C}, \mathcal{E}, \mathcal{P}) \tag{1}$$

Where $R$ is the set of user reviews, $\mathcal{C}$ represents task-specific constraints, $\mathcal{E}$ are stylistic exemplars, and $\mathcal{P}$ is the task prompt provided to the LLM.

### *Length Control*

In this setting, we introduce a user-centric control mechanism to specify the desired number of ‘PROS’ and ‘CONS’ sentences for the critical summary. An additional parameter is included in the LLM prompt to guide the generation length. The number of ‘PROS’ and ‘CONS’ is determined based on the ground truth critical summary for the item. By explicitly instructing the LLM with these parameters, we ensure the generated summary aligns with the expected structure and length.

### *Sketch ⇒ Fetch ⇒ Fill (SFF)*

To evaluate sentiment and length-controlled summaries, we parse LLM outputs into structured JSON format. However, LLMs sometimes produce incomplete or malformed outputs (some examples in Appendix B). To address this, we propose the Sketch-Fetch-Fill (SFF) approach for reliable JSON extraction.

1. **Sketch:** We first define the expected JSON structure, specifying key fields (e.g., ‘pros’ and ‘cons’) to guide reconstruction.
2. **Fetch:** Regular expressions are used to extract relevant content from the output, identifying text corresponding to the predefined fields, even with formatting inconsistencies.
3. **Fill:** The extracted data is inserted into the predefined structure, correcting common errors (e.g., missing quotes or misplaced commas) to ensure a valid, parsable JSON.

## *3.2. LFOSum: RAG Framework*

A key component of any RAG framework is the availability of query terms Zhao et al.[34]. In our case, the query terms for an entity are not pre-defined or readily available. To address this, we employ a simple yet effective method to extract query terms from the large input reviews (§3.2.1). These extracted terms are then used to design a combined extractive-generative framework for managing long-form input reviews

doi.org/10.32388/D1MVB5

through retrieval augmentation Lewis et al.[35]. This approach integrates the attributable and scalable properties of extractive methods (§3.2.2) with the coherence and fluency of LLMs (§3.2.3). Another advantage of our RAG framework is that it enables the automatic evaluation of generated summaries in manageable units, allowing for a more fine-grained assessment within long-form context (§3.2.4).

### *3.2.1. Query Term Extraction*

Let $M_e$ denote the language model capturing opinions about an entity $e$, defined as the probability distribution over word sequences. Under the query likelihood model, entity $e$ is considered *relevant* to query term $q$ if q is *likely generated* by $M_e$. Hence more frequent terms in the reviews of an entity may be treated as important query terms, with the exception of common stop words, and the summarization of an entity may be centered around these important terms.

A related task in the literature is aspect extraction, which can be categorized into two types: **(1)** Explicit aspects and **(2)** Implicit aspects[36][37]. Explicit aspects are directly mentioned targets in opinionated sentences, such as *"ocean view"* or *"spa service."* In contrast, implicit aspects are inherently expressed concepts that can generalize explicit examples; for instance, *"ocean view"* may relate to the broader category of *"location,"* while *"spa service"* falls under *"service."* In designing our RAG framework, we focus on explicit aspects (referred to as "***query terms***") due to their repetitive nature in long-form reviews, which facilitates the retrieval of salient sentences covering diverse user concerns. Below, we outline the major components of the query term extraction process:

#### *Candidate Term Extraction & Ranking*

We extract the most frequent unigrams and skip bigrams within a defined window size of 4. This approach captures meaningful multi-word expressions that may not be adjacent but contribute contextually to the overall understanding of the text. To filter out rare or insignificant terms, we apply a frequency threshold, ensuring that only high-frequency, representative terms are retained[4]. The terms are then ranked based on their frequency values to prioritize those most representative of the input reviews.

#### *Top-K Term Refinement*

The extracted query terms are further refined by cross-referencing them with the gold query term list from Pontiki et al.[38] for our domain of interest (i.e., Hotel). This helps eliminate frequent but irrelevant terms, such as stop words. To ensure that the final set of terms is diverse and non-redundant, single terms are removed if both of their constituent words appear within a multi-word query. Ultimately, the top-K most relevant query terms are selected for the retrieval step.

### 3.2.2. Retrieval

We divide user reviews into individual sentences and use the Top-K extracted query terms to retrieve relevant sentences as evidence for each query term, which are then provided as input to the LLMs. This approach offers two key advantages: **(1)** Retrieving sentences based on a diverse set of query terms reduces redundancy in the generated summaries, and **(2)** it increases information coverage from the user reviews[5]. The retrieval process is formalized as follows:

$$S_Q = \text{Top-K}\left(\mathcal{R}(Q, D)\right) \tag{2}$$

Where $Q$ is the set of query terms, $D$ is the collection of review sentences, $\mathcal{R}(Q, D)$ is the retrieval function, and $S_Q$ represents the Top-K retrieved sentences.

### Retrievers

We utilize two types of retrievers: BM25 and Dense retrievers. BM25 is a lexical retriever[39] that scores document relevance based on term frequency Robertson and Zaragoza[39], while Dense retrievers capture deeper contextual meanings through semantic information, ensuring both surface-level lexical matches and nuanced semantic relationships are covered. For the Dense retriever, we employ Sentence Transformers Reimers and Gurevych[40], specifically leveraging the checkpoint[7] due to its superior performance in semantic search across a wide range of benchmarks.

### 3.2.3. LLM as Reranker and Abstractor

We utilize the retrieved sentences for each query term as evidence and instruct LLMs to generate summaries. Two variants of summarization approaches are employed: (1) Extractive and (2) Abstractive. In both cases, LLMs are prompted with the retrieved sentences, and the outputs are aligned in a specified JSON format. The general process for both approaches can be formalized as follows:

$$\text{Summary}(Q) = \text{LLM}(Q, S_Q, \mathcal{C}, \mathcal{P}) \tag{3}$$

Where $Q$ is the query term, $S_Q$ is the set of Top-K retrieved sentences, $\mathcal{C}$ represents the constraints, and $\mathcal{P}$ is the prompt provided to the LLM.

#### *Extractive*

In the extractive approach, LLMs are prompted with a task description, constraints, the query term, and a list of Top-K retrieved sentences. The LLM is instructed to rerank the sentences and select the most relevant one, functioning primarily as a reranker. The complete prompt used for this process is shown in Appendix (Figure 4).

#### *Abstractive*

For the abstractive approach, LLMs are prompted with a task description, constraints, the query term, a list of Top-K retrieved sentences, and stylistic exemplars to guide the output in the desired style. The LLM synthesizes a summary based on the retrieved information, effectively acting as an abstractor. The full prompt used for this task is presented in Appendix (Figure 5).

### *3.2.4. RAG Verification*

To evaluate the ability of LLMs to generate summaries that accurately reflect the input evidence, we build upon the work of Bhaskar et al.[41], who developed desiderata for human evaluation, by introducing automatic evaluation metrics. Our goal is to break down sentences into structured components, allowing for a more granular and fine-grained assessment of factual alignment. We employ Aspect-Opinion-Sentiment (AOS) triplets[42], using a pre-trained model from Scaria et al.[43], which captures both implicit and explicit aspects (as detailed in §3.2.1). Each triplet decomposes the sentence into three core components:

- **Aspect:** The attribute or feature being discussed (e.g., *"room bathroom"*).
- **Opinion:** The expression or judgment about the aspect (e.g., *"clean"*).
- **Sentiment:** The polarity of the opinion (e.g., *negative*, *neutral*, or *positive*).

Given a set of retrieved sentences for each query, and a generated sentence, we evaluate the quality of the generated sentences for the Top-K queries of an entity based on three key metrics:

- **Aspect Relevance (AR):** Measures how well the aspect in the generated sentence aligns with the most important and frequent aspects mentioned in the retrieved evidence. This ensures the summary remains on topic and covers critical aspects.
- **Sentiment Factuality (SF):** Evaluates for a given aspect whether the sentiment in the generated sentence matches the most frequent sentiment found in the retrieved evidence, ensuring that the

sentiment expressed is factually accurate.

- **Opinion Faithfulness (OF):** Assesses for a given aspect and sentiment whether the opinion expressed in the generated sentence is consistent with the opinions found in the retrieved evidence, either through direct matching or semantic similarity.

### *Aspect Relevance (AR)*

For each query, AOS triplets are extracted from both the retrieved and generated sentences. We identify the most frequent aspect from the retrieved evidence and check if it appears in the generated sentence. Aspect Relevance, in this context, is a binary variable, indicating whether the generated sentence remains on-topic by covering the most important aspect. We are interested in the expectation of this variable over generated sentences.

### *Sentiment Factuality (SF)*

For each aspect, sentiments are extracted from AOS triplets of both the retrieved and generated sentences. Neutral sentiments are excluded as they provide limited insight. For each aspect, the most frequent non-neutral sentiment from the retrieved sentences is identified, and the sentiment in the generated sentence is checked for alignment. Similar to AR, SF is a binary variable, indicating whether the generated sentiment is factually correct. Again, we are interested in the expectation of this variable over generated sentences.

| Models | Context Length | Settings | PROS Scores | | | CONS Scores | | | Format Following | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | R1 | RL | BERTScore | R1 | RL | BERTScore | JSON Parsing | SFF (*ours*) |
| | 128K | **GPT-4o-mini** | 29.87 | 17.15 | 65.40 | 15.68 | 9.66 | 57.06 | 500 / 500 | 500 / 500 |
| | | – w/ Length Control [Edit] | **30.99** | **17.75** | **66.24** | 18.39 | 12.03 | 60.31 | 500 / 500 | 500 / 500 |
| AI | 200K | Claude-3-Haiku | 30.67 | 18.08 | 66.74 | 20.19 | 12.91 | 60.85 | 457 / 500 | 500 / 500 |
| | | – w/ Length Control [Edit] | 30.50 | 17.41 | 66.07 | 19.67 | 12.24 | 61.09 | 440 / 500 | 500 / 500 |
| | 1M | Gemini-1.5-Flash | 31.48 | 17.80 | 65.84 | 17.77 | 11.26 | 58.11 | 343 / 500 | 372 / 372 |
| | | – w/ Length Control [Edit] | 30.34 | 17.65 | 65.71 | **18.87** | **12.92** | **61.24** | 489 / 500 | 495 / 495 |

**Table 2.** Evaluation results of our LFOSum: **Long-form Critic** method. PROS refers to positive summaries and CONS refers to negative summaries. Best scores for the Length Control setting are marked in **bold**, while the highest results in the basic setting are underlined. “JSON Parsing” shows the number of samples successfully parsed directly, and “SFF (*ours*)” indicates samples recovered using our SFF method from the valid summaries.

### *Opinion Faithfulness (OF)*

For each aspect and sentiment, opinions are extracted from AOS triplets of both retrieved and generated sentences. A direct opinion match is assigned a score of 1, while indirect matches are evaluated using a semantic similarity function (e.g., cosine similarity), which returns a value between 0 and 1. This allows for semantically similar opinions (e.g., *"beautiful"* and *"stunning"*) to be considered faithful. Therefore, the opinion faithfulness for a given aspect and sentiment is represented as a random variable ranging from 0 to 1, and we report its expected value over generated sentences.

## 4. Evaluation

In this section, we evaluate the performance of our two proposed approaches: **(1)** the Long-form Critic (§4.1) and **(2)** the RAG Framework (§4.2). We assess these methods using a variety of open-source and closed-source models, comparing their performance on standard and newly proposed evaluation metrics. The experimental setup is detailed in the Appendix A, generated summaries in Table 9, Table 10 & Table 11, and the related works are covered in Appendix E.

### *Automatic Evaluation*

We use F1 scores of ROUGE (R1 and RL)[44] and BERTScore[45], following Bhaskar et al.[41]. Although ROUGE scores have been shown to be less reliable for generic opinion summarization tasks[46][47], we report them for consistency with recent studies[41][48][49][10], to benchmark our dataset and methods in long-form settings, and to contribute to discussions on automatic evaluation methods for long-form opinion summarization (§5). Additionally, we use our proposed evaluation metrics to assess the faithfulness of the LLM-generated summaries.

### *4.1. Evaluating Long-form Critic*

We evaluate the ability of several LLMs to generate critical summaries divided into 'PROS' and 'CONS'. For this purpose, we utilized long-context LLMs, providing the full set of user reviews as input. We experimented with several closed-source models, including GPT-4o-mini[8], Claude-3-Haiku[9], and Gemini-1.5-Flash[10], alongside open-source models such as Llama-3.2-3B-Instruct[50] and Phi-3.5-mini-instruct[51], each with varying context lengths.

However, we encountered significant challenges with open-source models. As highlighted in Xia et al.[52], these models frequently failed to adhere to the expected output format, often producing non-parsable JSON outputs, even when employing our SFF method for parsing (§3.1). We directly parsed the expected JSON outputs from the LLMs, and in cases of errors (detailed in the Appendix B), we attempted to automatically recover them using our SFF method (§3.1). If the context length exceeded the model's limit, we truncated the older reviews, prioritizing more recent ones based on posting dates. Summaries were considered valid only if both the "pros" and "cons" sections were not empty, and any invalid summaries were excluded from the evaluation.

### *Results & Analysis*

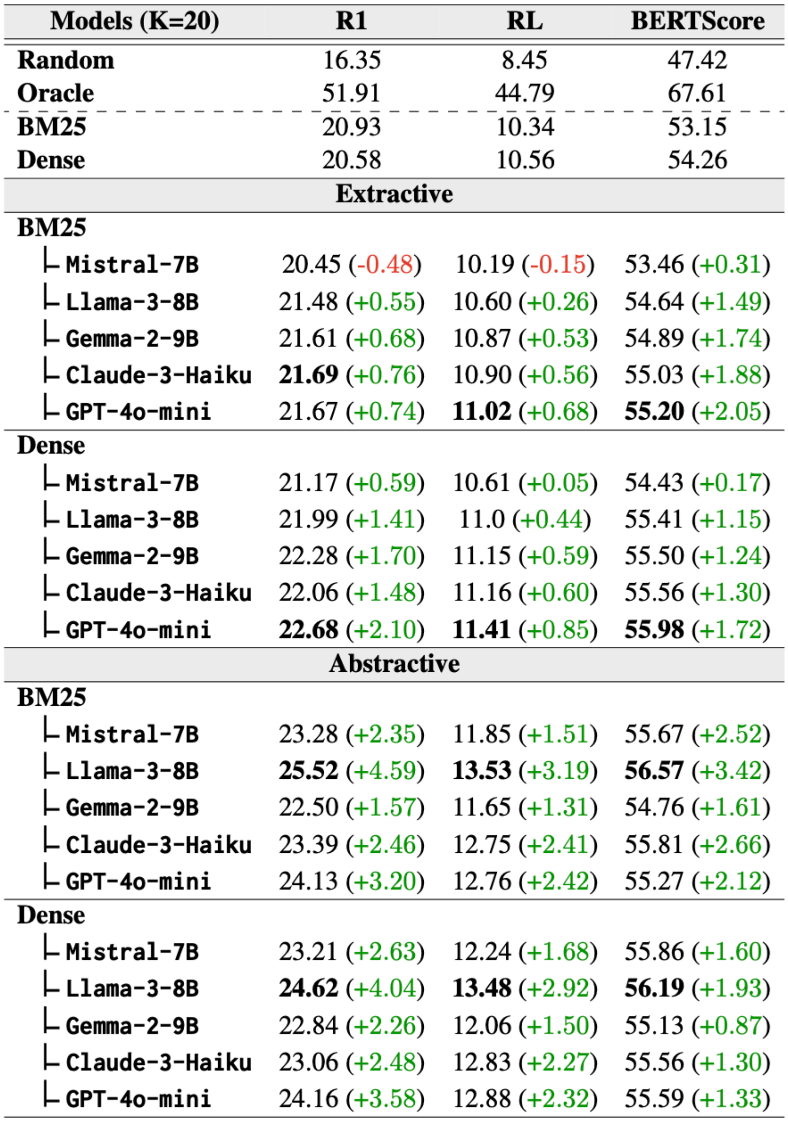
As shown in Table 2, Claude-3-Haiku produces the best summaries in the basic setting for both 'PROS' and 'CONS'. However, across all models, 'CONS' performance is generally weaker, likely because negative reviews are less frequent compared to positive ones[53], making it harder for the models to capture *"needle-in-a-haystack"* information within long-form inputs[54]. In the length control setting, GPT-4o-mini excels in 'PROS', while Gemini-1.5-Flash performs better in 'CONS', likely due to its larger context window. Claude-3-Haiku struggles with length adherence, as noted in Appendix (Table 6). Gemini-1.5-Flash generated 372 out of 500 valid summaries, with the remaining invalid due to empty fields, elaborated more in §5. These results highlight the challenge of balancing sentiment and format adherence in long-form summaries.

## *4.2. Evaluating RAG Framework*

We evaluate our RAG Framework using both open-source and closed-source models. A maximum of 15 top query terms (K=15) are selected for the retrievers, and for each query term, we experiment with retrieving 10 and 20 sentences. For both summary variants—(1) Extractive and (2) Abstractive—the system-generated summary is created by merging the sentences for each query term, as detailed in §3.2.3. The 'PROS' and 'CONS' from the gold summaries are merged to form a generic reference summary, following the standard opinion summarization evaluation protocol without sentiment control[41].

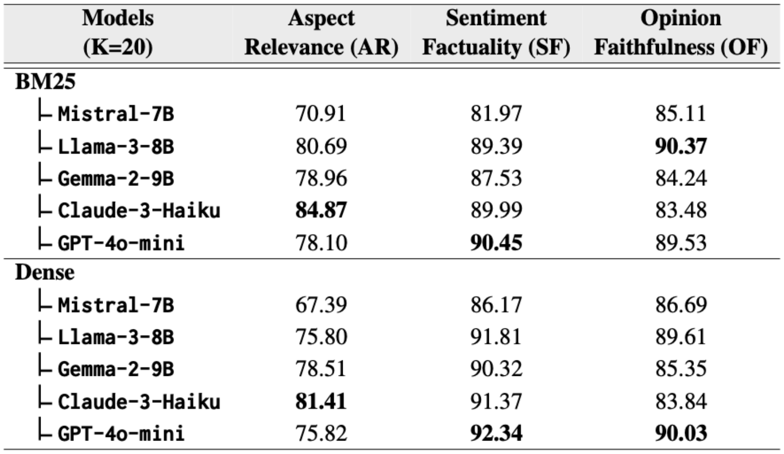
### *Baselines*

For the BM25 and Dense baselines, we select the top sentence retrieved for each of the K query terms to form the summary. For the random baseline, K sentences are randomly selected from the input reviews for each entity. As an upper-bound baseline, the Oracle selects the sentence with the highest ROUGE-L (RL) score for each gold summary sentence, providing an approximate upper limit for performance.



| Models (K=20) | R1 | RL | BERTScore |
|---|---|---|---|
| **Random** | 16.35 | 8.45 | 47.42 |
| **Oracle** | 51.91 | 44.79 | 67.61 |
| **BM25** | 20.93 | 10.34 | 53.15 |
| **Dense** | 20.58 | 10.56 | 54.26 |
| **Extractive** | | | |
| **BM25** | | | |
| ├ Mistral-7B | 20.45 (-0.48) | 10.19 (-0.15) | 53.46 (+0.31) |
| ├ Llama-3-8B | 21.48 (+0.55) | 10.60 (+0.26) | 54.64 (+1.49) |
| ├ Gemma-2-9B | 21.61 (+0.68) | 10.87 (+0.53) | 54.89 (+1.74) |
| ├ Claude-3-Haiku | **21.69** (+0.76) | 10.90 (+0.56) | 55.03 (+1.88) |
| ├ GPT-4o-mini | 21.67 (+0.74) | **11.02** (+0.68) | **55.20** (+2.05) |
| **Dense** | | | |
| ├ Mistral-7B | 21.17 (+0.59) | 10.61 (+0.05) | 54.43 (+0.17) |
| ├ Llama-3-8B | 21.99 (+1.41) | 11.0 (+0.44) | 55.41 (+1.15) |
| ├ Gemma-2-9B | 22.28 (+1.70) | 11.15 (+0.59) | 55.50 (+1.24) |
| ├ Claude-3-Haiku | 22.06 (+1.48) | 11.16 (+0.60) | 55.56 (+1.30) |
| ├ GPT-4o-mini | **22.68** (+2.10) | **11.41** (+0.85) | **55.98** (+1.72) |
| **Abstractive** | | | |
| **BM25** | | | |
| ├ Mistral-7B | 23.28 (+2.35) | 11.85 (+1.51) | 55.67 (+2.52) |
| ├ Llama-3-8B | **25.52** (+4.59) | **13.53** (+3.19) | **56.57** (+3.42) |
| ├ Gemma-2-9B | 22.50 (+1.57) | 11.65 (+1.31) | 54.76 (+1.61) |
| ├ Claude-3-Haiku | 23.39 (+2.46) | 12.75 (+2.41) | 55.81 (+2.66) |
| ├ GPT-4o-mini | 24.13 (+3.20) | 12.76 (+2.42) | 55.27 (+2.12) |
| **Dense** | | | |
| ├ Mistral-7B | 23.21 (+2.63) | 12.24 (+1.68) | 55.86 (+1.60) |
| ├ Llama-3-8B | **24.62** (+4.04) | **13.48** (+2.92) | **56.19** (+1.93) |
| ├ Gemma-2-9B | 22.84 (+2.26) | 12.06 (+1.50) | 55.13 (+0.87) |
| ├ Claude-3-Haiku | 23.06 (+2.48) | 12.83 (+2.27) | 55.56 (+1.30) |
| ├ GPT-4o-mini | 24.16 (+3.58) | 12.88 (+2.32) | 55.59 (+1.33) |

**Table 3.** Evaluation results of our LFOSum: RAG Framework with K=20, where K is the number of retrieved sentences. The best results compared to their respective baseline models are marked in **bold**, and Δ gains are shown in round brackets and highlighted in green for improvements and red for declines.



| Models (K=20) | Aspect Relevance (AR) | Sentiment Factuality (SF) | Opinion Faithfulness (OF) |
|---|---|---|---|
| **BM25** | | | |
| ├ Mistral-7B | 70.91 | 81.97 | 85.11 |
| ├ Llama-3-8B | 80.69 | 89.39 | **90.37** |
| ├ Gemma-2-9B | 78.96 | 87.53 | 84.24 |
| ├ Claude-3-Haiku | **84.87** | 89.99 | 83.48 |
| ├ GPT-4o-mini | 78.10 | **90.45** | 89.53 |
| **Dense** | | | |
| ├ Mistral-7B | 67.39 | 86.17 | 86.69 |
| ├ Llama-3-8B | 75.80 | 91.81 | 89.61 |
| ├ Gemma-2-9B | 78.51 | 90.32 | 85.35 |
| ├ Claude-3-Haiku | **81.41** | 91.37 | 83.84 |
| ├ GPT-4o-mini | 75.82 | **92.34** | **90.03** |

**Table 4.** RAG verification results on the Abstractive summary variant with K=20, where K is the number of retrieved sentences. Scores are multiplied by 100 for better readability. The best results are marked in **bold**.

### *Results & Analysis*

As presented in Table 3, for the Extractive summary variant, the closed-source models (Claude-3-Haiku and GPT-4o-mini) generally outperform the open-source models across all metrics. However, in the Abstractive variant, the performance of open-source models, particularly Llama-3-8B, improves significantly. This suggests that in settings requiring more abstraction and synthesis, open-source models can effectively narrow the gap between themselves and their closed-source counterparts, especially when relevant information is retrieved in a focused manner. In both extractive and abstractive settings, summaries driven by the most important query terms directly impact overall performance. The Oracle baseline further shows that there is still considerable room for improvement, highlighting the inherent challenges in long-form summarization. For RAG verification (Table 4), closed-source models outperform open-source models across key metrics. Claude-3-Haiku excels in AR and SF, demonstrating its ability to stay focused on relevant aspects while maintaining factually aligned sentiment. GPT-4o-mini shows strong performance in SF and leads in OF, ensuring that the sentiments and opinions expressed in the generated summaries are consistent with the retrieved evidence. Similar trends are observed with K=10, as presented in Appendix Tables 7 and 8, which reinforce the results seen with K=20.

## 5. Discussion and Future Directions

### *Moderation Issues in User Reviews*

In the basic setting, Gemini-1.5-Flash generated several invalid summaries due to sensitive or inappropriate content, such as *"Manager is an African middle-aged man who was irresponsible and harsh"* and *"Want more offers?? Call me +1 111 222 ******,"* triggering its safety mechanism[11]. Even after disabling safety filters, the issue persisted, highlighting the difficulty of handling long-form user reviews. However, in the length-controlled setting, the model produced fewer invalid summaries by prioritizing safer content. Other models did not face similar issues, possibly due to different content moderation filters. Addressing these challenges presents an important area for future work.

### *Evaluation*

Evaluating opinion summarization for long-form user reviews is especially challenging, whether through automatic or human assessments. Human evaluation metrics such as Fluency, Coherence, and Non-Redundancy[18][19] are often less applicable when designing systems based on LLMs[55]. Moreover, most existing LLM-based evaluators are tailored to short input reviews[28]. Our dataset, with its explicit 'PROS' and 'CONS' paired with long-form reviews, offers opportunities to develop more suitable LLM-based evaluation metrics.

## 6. Conclusion

In this paper, we addressed key challenges in long-form opinion summarization by introducing a new dataset of over a thousand user reviews per entity, paired with in-depth critical summaries from domain experts. We proposed two training-free summarization methods utilizing RAG and long-context LLMs, designed for scalable and controllable summarization. Additionally, we developed novel reference-free evaluation metrics that offer a fine-grained, context-sensitive assessment of summary faithfulness. Furthermore, based on our insights, we offer suggestions for future research.

## Limitations

In this work, we evaluated our proposed methods using a selection of both open-source and closed-source LLMs. We intentionally focused on cost-effective yet efficient closed-source models and open-source models that can be deployed on consumer-grade hardware, given the constraints of *academic settings*. The performance of more powerful, large-scale models remains unexplored, but we encourage the broader research community to benchmark these models using our dataset and methods.

While we experimented with different retrievers (BM25 and Dense) for both summary variants using Top-K values of 10 and 20, other retriever configurations might yield better performance. Optimizing for additional retriever options is beyond the scope of this study, but we acknowledge that further exploration in this area could lead to improvements.

Although we proposed novel automatic evaluation metrics built on top of the RAG framework with retrieved evidence, their applicability may be limited in full long-form settings where complete retrieval is not feasible. This remains a potential avenue for future research.

Finally, our research and the development of LFOSum are exclusively centered on the English language. This means its use and effectiveness might not be the same for other languages.

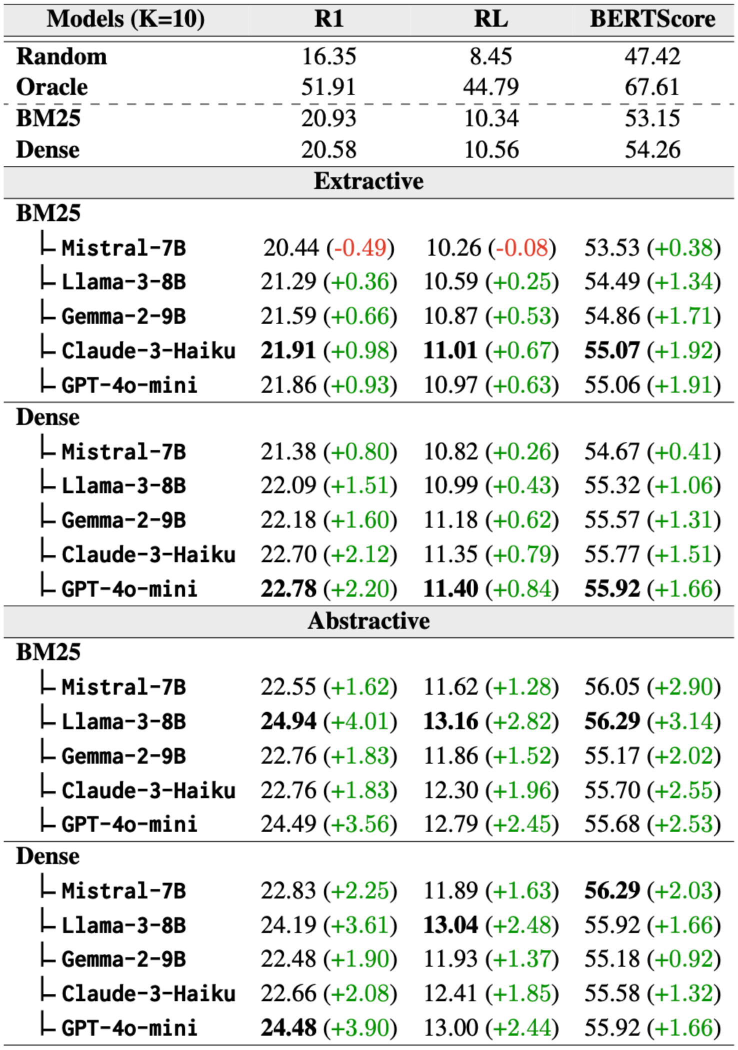
## Supplementary Material: Appendices

| **TripAdvisor [Source Reviews]** | | | **Oyster [Reference Summary]** | |
|---|---|---|---|---|
| #Entities | 500 | | #Entities | 500 |
| Avg. #Reviews | ≈1.5K | | Avg. #Sents | 12.29 |
| Avg. #Sents | ≈10.5K | | Avg. #PROS | 8.45 |
| Avg. #Words | ≈196K | | Avg. #CONS | 3.84 |
| Avg. #Tokens | ≈207K | | Avg. #Tokens | 105.98 |

**Table 5.** Statistics of the LFOSum evaluation dataset. ‘#Entities‘ denotes the total number of entities. For Source Reviews, the averages include the number of user reviews (‘Avg. #Reviews‘), sentences (‘Avg. #Sents‘), words (‘Avg. #Words‘), and tokens (‘Avg. #Tokens‘, computed using the GPT-4o tokenizer) per entity. For Reference Summary, the averages represent the number of sentences (‘Avg. #Sents‘), positive sentences (‘Avg. #PROS‘), negative sentences (‘Avg. #CONS‘), and tokens (‘Avg. #Tokens‘) per entity.

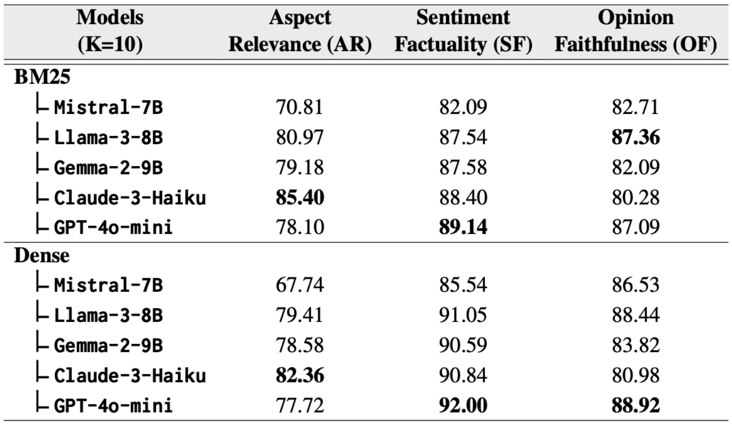
| Models | PROS Length | CONS Length | Overall |
|---|---|---|---|
| **GPT-4o-mini** | 489 / 500 | 495 / 500 | 484 / 500 |
| **Claude-3-Haiku** | 480 / 500 | 455 / 500 | 450 / 500 |
| **Gemini-1.5-Flash** | 493 / 495 | 490 / 495 | 490 / 495 |

**Table 6.** Length control evaluation results of our Long-form Critic method. “PROS Length” refers to the number of generated summaries that adhered to the expected length for positive summaries, while “CONS Length” indicates adherence to the length for negative summaries. “Overall” represents the total number of summaries where both lengths were followed correctly.



| Models (K=10) | R1 | RL | BERTScore |
|---|---|---|---|
| **Random** | 16.35 | 8.45 | 47.42 |
| **Oracle** | 51.91 | 44.79 | 67.61 |
| **BM25** | 20.93 | 10.34 | 53.15 |
| **Dense** | 20.58 | 10.56 | 54.26 |
| **Extractive** | | | |
| **BM25** | | | |
| └ **Mistral-7B** | 20.44 (-0.49) | 10.26 (-0.08) | 53.53 (+0.38) |
| └ **Llama-3-8B** | 21.29 (+0.36) | 10.59 (+0.25) | 54.49 (+1.34) |
| └ **Gemma-2-9B** | 21.59 (+0.66) | 10.87 (+0.53) | 54.86 (+1.71) |
| └ **Claude-3-Haiku** | **21.91** (+0.98) | **11.01** (+0.67) | **55.07** (+1.92) |
| └ **GPT-4o-mini** | 21.86 (+0.93) | 10.97 (+0.63) | 55.06 (+1.91) |
| **Dense** | | | |
| └ **Mistral-7B** | 21.38 (+0.80) | 10.82 (+0.26) | 54.67 (+0.41) |
| └ **Llama-3-8B** | 22.09 (+1.51) | 10.99 (+0.43) | 55.32 (+1.06) |
| └ **Gemma-2-9B** | 22.18 (+1.60) | 11.18 (+0.62) | 55.57 (+1.31) |
| └ **Claude-3-Haiku** | 22.70 (+2.12) | 11.35 (+0.79) | 55.77 (+1.51) |
| └ **GPT-4o-mini** | **22.78** (+2.20) | **11.40** (+0.84) | **55.92** (+1.66) |
| **Abstractive** | | | |
| **BM25** | | | |
| └ **Mistral-7B** | 22.55 (+1.62) | 11.62 (+1.28) | 56.05 (+2.90) |
| └ **Llama-3-8B** | **24.94** (+4.01) | **13.16** (+2.82) | **56.29** (+3.14) |
| └ **Gemma-2-9B** | 22.76 (+1.83) | 11.86 (+1.52) | 55.17 (+2.02) |
| └ **Claude-3-Haiku** | 22.76 (+1.83) | 12.30 (+1.96) | 55.70 (+2.55) |
| └ **GPT-4o-mini** | 24.49 (+3.56) | 12.79 (+2.45) | 55.68 (+2.53) |
| **Dense** | | | |
| └ **Mistral-7B** | 22.83 (+2.25) | 11.89 (+1.63) | **56.29** (+2.03) |
| └ **Llama-3-8B** | 24.19 (+3.61) | **13.04** (+2.48) | 55.92 (+1.66) |
| └ **Gemma-2-9B** | 22.48 (+1.90) | 11.93 (+1.37) | 55.18 (+0.92) |
| └ **Claude-3-Haiku** | 22.66 (+2.08) | 12.41 (+1.85) | 55.58 (+1.32) |
| └ **GPT-4o-mini** | **24.48** (+3.90) | 13.00 (+2.44) | 55.92 (+1.66) |

**Table 7.** Evaluation results of our LFOSum: RAG Framework with K=10, where K is the number of retrieved sentences. The best results compared to their respective baseline models are marked in bold, and Δ gains are shown in round brackets and highlighted in green for improvements and red for declines.



| Models<br>(K=10) | Aspect<br>Relevance (AR) | Sentiment<br>Factuality (SF) | Opinion<br>Faithfulness (OF) |
|---|---|---|---|
| **BM25** | | | |
| └ Mistral-7B | 70.81 | 82.09 | 82.71 |
| └ Llama-3-8B | 80.97 | 87.54 | **87.36** |
| └ Gemma-2-9B | 79.18 | 87.58 | 82.09 |
| └ Claude-3-Haiku | **85.40** | 88.40 | 80.28 |
| └ GPT-4o-mini | 78.10 | **89.14** | 87.09 |
| **Dense** | | | |
| └ Mistral-7B | 67.74 | 85.54 | 86.53 |
| └ Llama-3-8B | 79.41 | 91.05 | 88.44 |
| └ Gemma-2-9B | 78.58 | 90.59 | 83.82 |
| └ Claude-3-Haiku | **82.36** | 90.84 | 80.98 |
| └ GPT-4o-mini | 77.72 | **92.00** | **88.92** |

**Table 8.** RAG verification results on the Abstractive summary variant with K=10, where K is the number of retrieved sentences. Scores are multiplied by 100 for better readability. The best results are marked in bold.

## *Appendix A. Experimental Setup*

### *A.1. Model Configuration*

For our RAG Framework, we utilize both open-source models (Mistral-7B, Llama-3-8B, Gemma-2-9B) and closed-source models (Claude-3-Haiku, GPT-4o-mini). Across all models, we set consistent hyperparameters for both the Extractive and Abstractive summarization variants: max_new_tokens=256, temperature=0.7, and top_p=0.9.

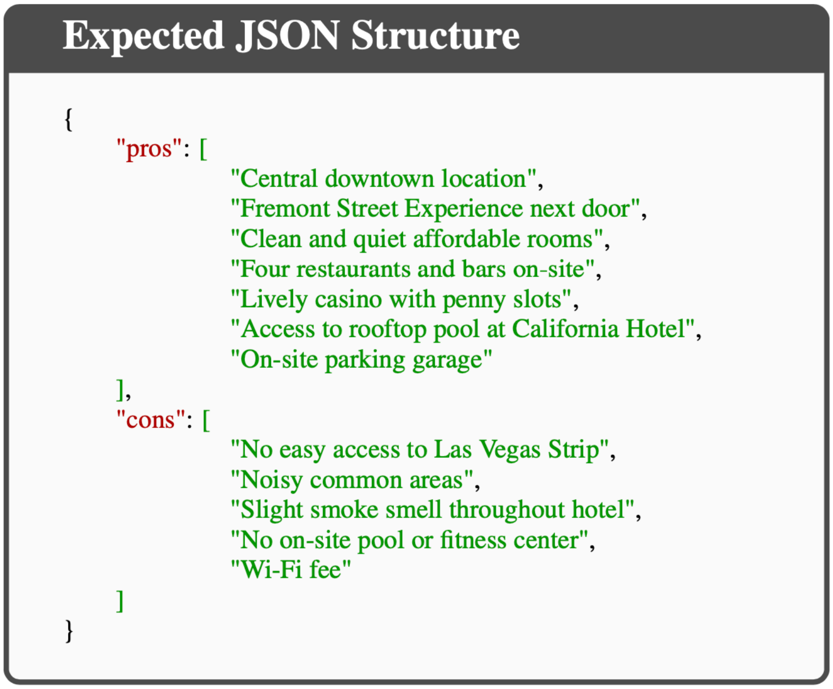
For the Long-form Critic, we retain the default parameters of the long-context LLMs (GPT-4o-mini, Claude-3-Haiku, Gemini-1.5-Flash), with the exception of max_tokens=512, as this value ensures the model can generate comprehensive critic summaries for long-form user reviews.

### *A.2. JSON Format Adherence*

To ensure that the LLMs output in a structured JSON format, we employ several strategies. These include explicitly stating the requirement for JSON output in the prompts, providing a sample JSON structure, and incorporating in-context examples with the desired format. For models such as those from OpenAI[12], released on August 6th, 2024. (GPT-4o-mini), we specify formatting instructions by configuring the necessary fields and descriptions (e.g., response_format="type": "json_object"). Similarly, for Gemini models, we use field descriptions (e.g., generation_config=“response_mime_type”: “application/json”) to enforce JSON outputs, ensuring reliable evaluation.

## *Appendix B. Common JSON Parsing Errors*

One of the key challenges when working with LLMs to generate sentiment and length-controlled summaries is ensuring that the outputs conform to a structured format, such as JSON. While the desired output is a well-formed JSON dictionary, LLMs sometimes produce outputs that are incomplete, malformed, or improperly structured, making them difficult or impossible to parse directly. Below, we outline the expected JSON format and common types of issues encountered when generating JSON from LLMs:

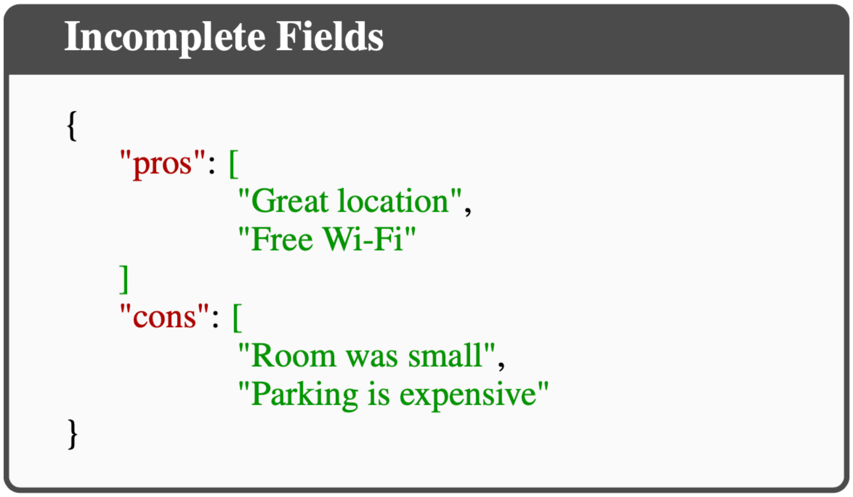


Expected JSON Structure

```
{
    "pros": [
            "Central downtown location",
            "Fremont Street Experience next door",
            "Clean and quiet affordable rooms",
            "Four restaurants and bars on-site",
            "Lively casino with penny slots",
            "Access to rooftop pool at California Hotel",
            "On-site parking garage"
    ],
    "cons": [
            "No easy access to Las Vegas Strip",
            "Noisy common areas",
            "Slight smoke smell throughout hotel",
            "No on-site pool or fitness center",
            "Wi-Fi fee"
    ]
}
```

**Incomplete Fields:**

LLMs generate partial outputs where entire fields, such as ’pros’ or ’cons’, are missing, incomplete, or malformed.



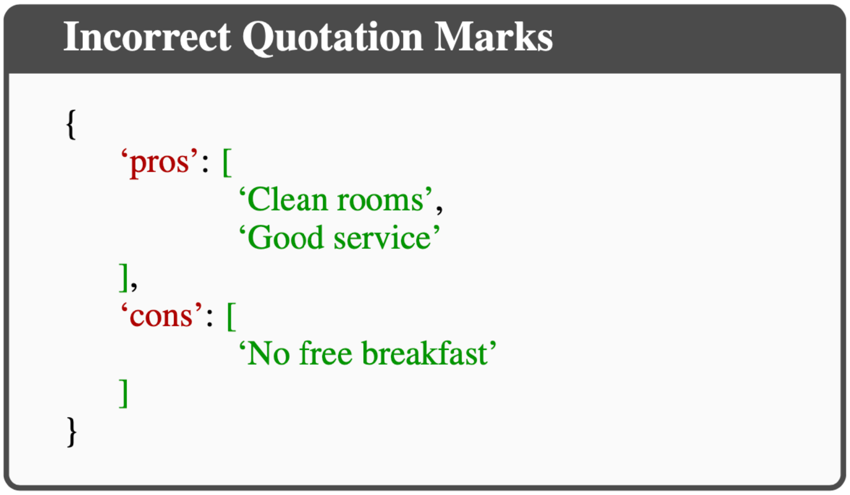
Incomplete Fields

```
{
    "pros": [
            "Great location",
            "Free Wi-Fi"
    ]
    "cons": [
            "Room was small",
            "Parking is expensive"
}
```

In this case, the missing comma after the "pros" list and the unclosed string in the "cons" list render this output invalid for parsing.

**Incorrect Quotation Marks:**

Inconsistent use of single (‘) and double (") quotes is a common issue, as JSON requires strict adherence to double quotes for both keys and values.



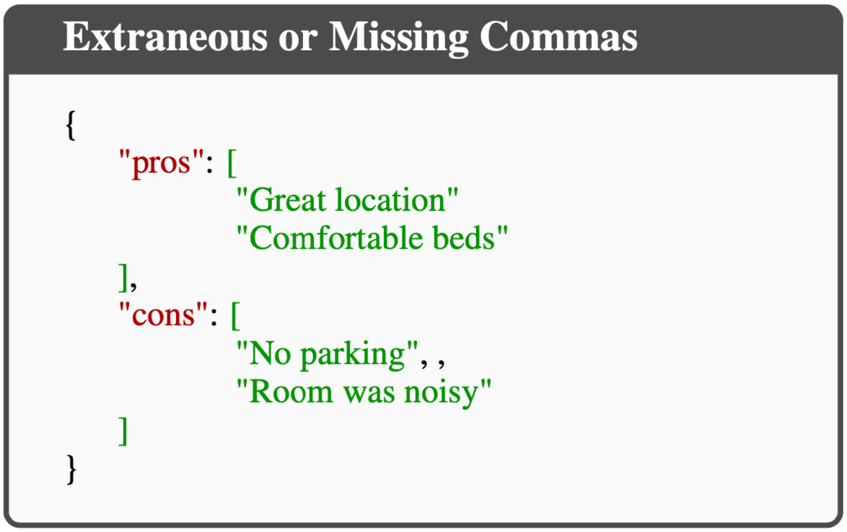
Incorrect Quotation Marks

```
{
    ‘pros’: [
            ‘Clean rooms’,
            ‘Good service’
    ],
    ‘cons’: [
            ‘No free breakfast’
    ]
}
```

This output uses single quotes, making it incompatible with standard JSON parsers.

**Extraneous or Missing Commas:**

LLMs often omit or misplace commas between key-value pairs or list elements, which breaks the JSON structure.



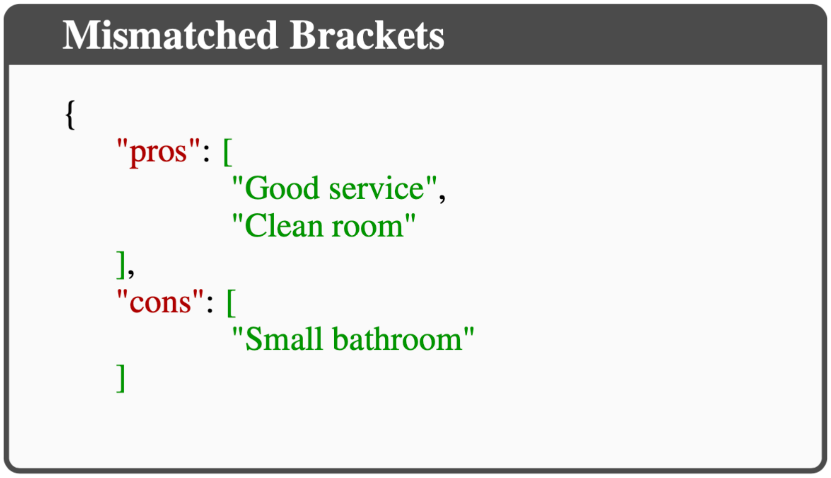
**Extraneous or Missing Commas**

```
{
    "pros": [
            "Great location"
            "Comfortable beds"
    ],
    "cons": [
            "No parking", ,
            "Room was noisy"
    ]
}
```

The missing comma between “Great location” and “Comfortable beds” and invalid comma between “No parking” and “Room was noisy”, render this JSON invalid.

**Mismatched Brackets:**

Unbalanced or missing curly braces ({}) and square brackets ([]) are frequent, especially when generating long lists or deeply nested structures.



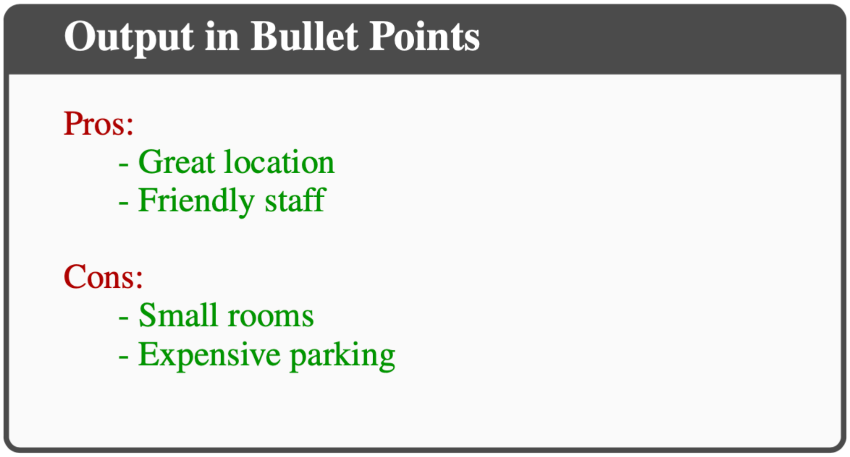
**Mismatched Brackets**

```
{
    "pros": [
            "Good service",
            "Clean room"
    ],
    "cons": [
            "Small bathroom"
    ]
```

In this case, the closing curly brace is missing, leading to a syntax error.

**Output in Bullet Points:**

LLM outputs are sometimes structured informally (e.g., using bullet points to list pros and cons), a common format in user-generated content. This structure cannot be directly parsed, as shown in the following example:



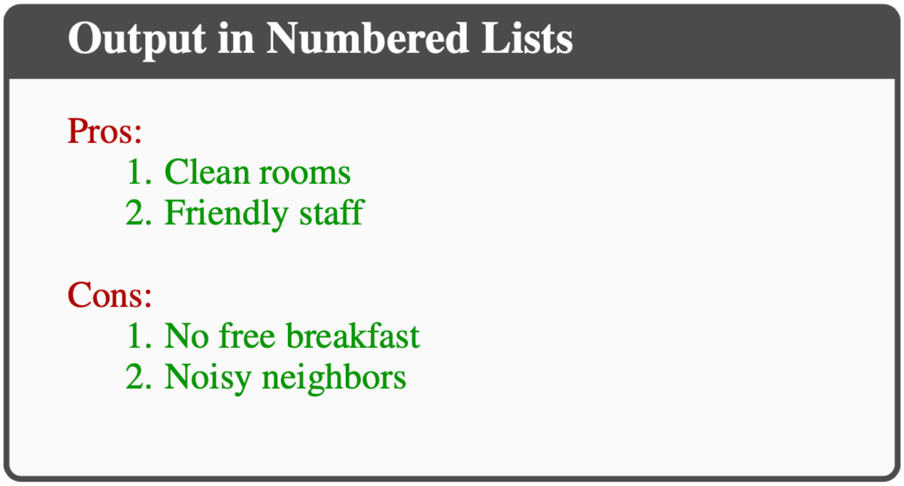
**Output in Bullet Points**

Pros:
- Great location
- Friendly staff

Cons:
- Small rooms
- Expensive parking

**Output in Numbered Lists:**

Outputs can also appear in a numbered list format. Due to formatting inconsistencies, these cannot be parsed directly. This issue was particularly observed during our experiments with length-controlled summary generation, as many user reviews present pros and cons in this format.



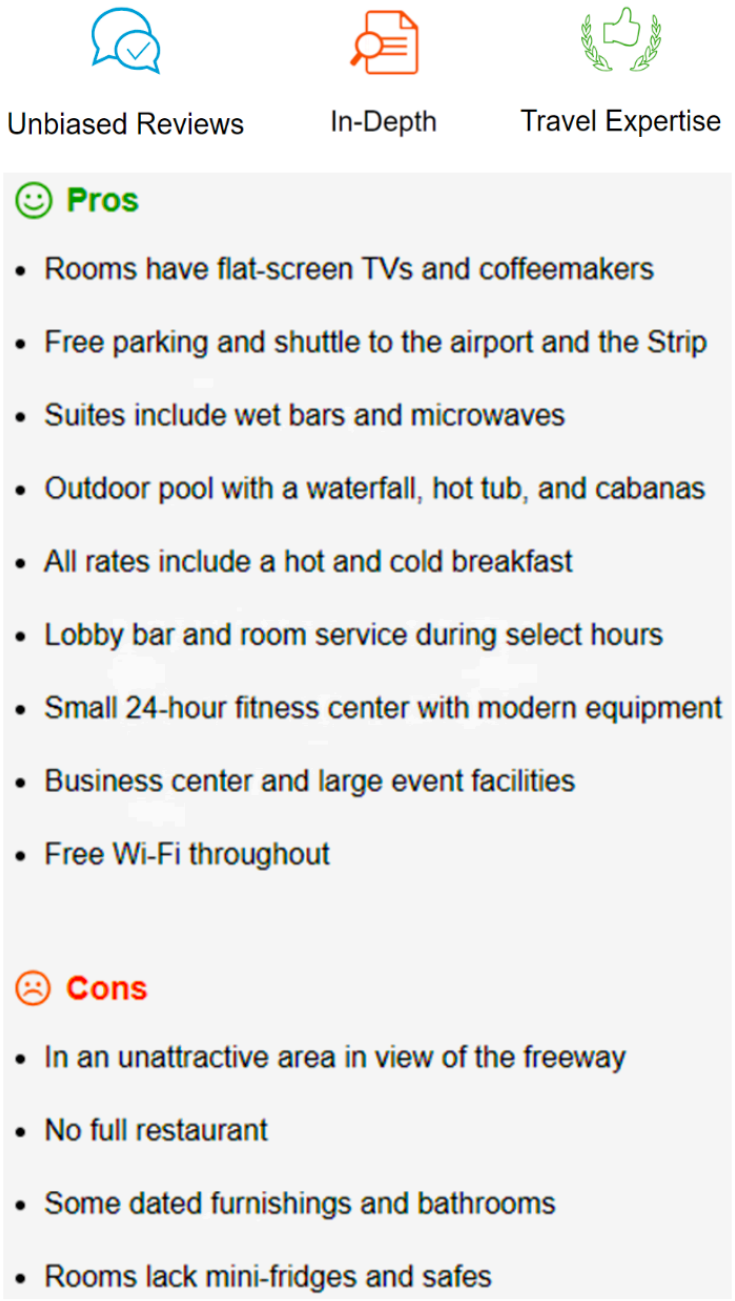
**Output in Numbered Lists**

Pros:
1. Clean rooms
2. Friendly staff

Cons:
1. No free breakfast
2. Noisy neighbors

**Minimal Structure:**

In some cases, LLM outputs can include lists of pros and cons presented as comma-separated strings within a sentence-like format. This structure often deviates from standard JSON formatting, making it difficult to parse directly, as shown in the following example.



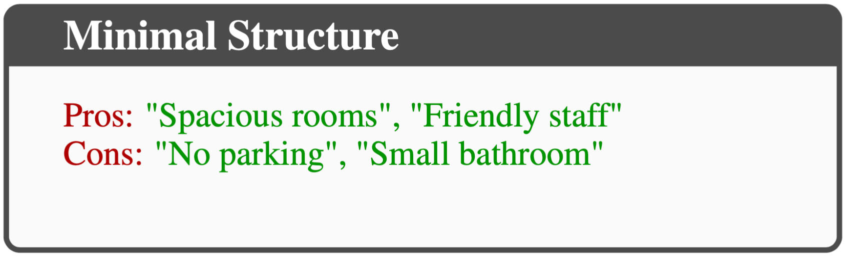
☺ **Pros**

- Rooms have flat-screen TVs and coffeemakers
- Free parking and shuttle to the airport and the Strip
- Suites include wet bars and microwaves
- Outdoor pool with a waterfall, hot tub, and cabanas
- All rates include a hot and cold breakfast
- Lobby bar and room service during select hours
- Small 24-hour fitness center with modern equipment
- Business center and large event facilities
- Free Wi-Fi throughout

☹ **Cons**

- In an unattractive area in view of the freeway
- No full restaurant
- Some dated furnishings and bathrooms
- Rooms lack mini-fridges and safes

**Figure 2.** A sample example from our dataset. Hampton Inn Tropicana (https://www.oyster.com/las-vegas/hotels/hampton-inn-tropicana/)



Minimal Structure

Pros: "Spacious rooms", "Friendly staff"
Cons: "No parking", "Small bathroom"

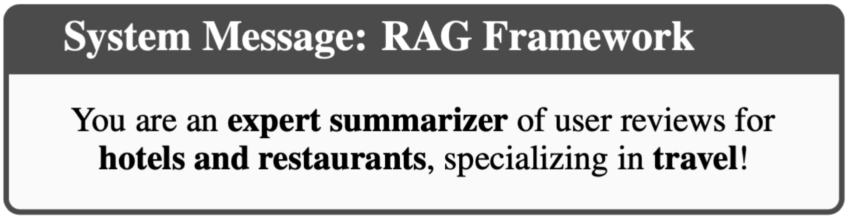
*Appendix C. System Message Design*

To guide the LLM for opinion summarization, we developed a system message specifying the model's role and constraints. The message defines the LLM as an "expert summarizer of user reviews" within the domain of "hotels and restaurants," with a specialization in "travel." These elements were designed with several key considerations:

**Role and Task:** Defining the LLM as an expert ensures focused, high-quality outputs. It helps the model capture relevant sentiments and aspects while minimizing irrelevant details.

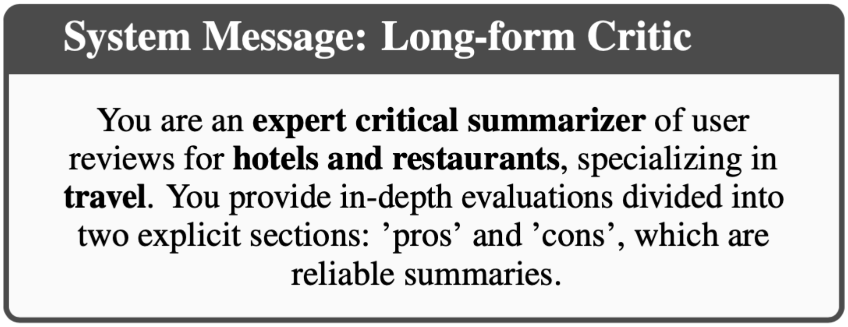
**Domain:** Narrowing the scope to hotels and restaurants ensures the model prioritizes key factors such as service quality, location, and amenities—critical in user-generated travel reviews.

**Specialization:** Adding a travel specialization refines the model's focus on aspects unique to travelers, such as proximity to attractions and comfort during stays.



System Message: RAG Framework

You are an **expert summarizer** of user reviews for **hotels and restaurants**, specializing in **travel**!



System Message: Long-form Critic

You are an **expert critical summarizer** of user reviews for **hotels and restaurants**, specializing in **travel**. You provide in-depth evaluations divided into two explicit sections: 'pros' and 'cons', which are reliable summaries.

This system messages are crafted to align the model's outputs with user needs, ensuring summaries remain concise, relevant, and actionable for travel-related decisions.

## *Appendix D. Data Preprocessing*

In our preprocessing pipeline, we focused on filtering the review text based on language without making any explicit modifications to the content of the reviews themselves. We retained only sentences written in English, removing those written in other languages to ensure consistency in the dataset, similar to previous approaches[56]. For language identification, we employed the spacy-langdetect[13] module, which allowed us to efficiently detect and filter out non-English content, following practices outlined in recent work[57].

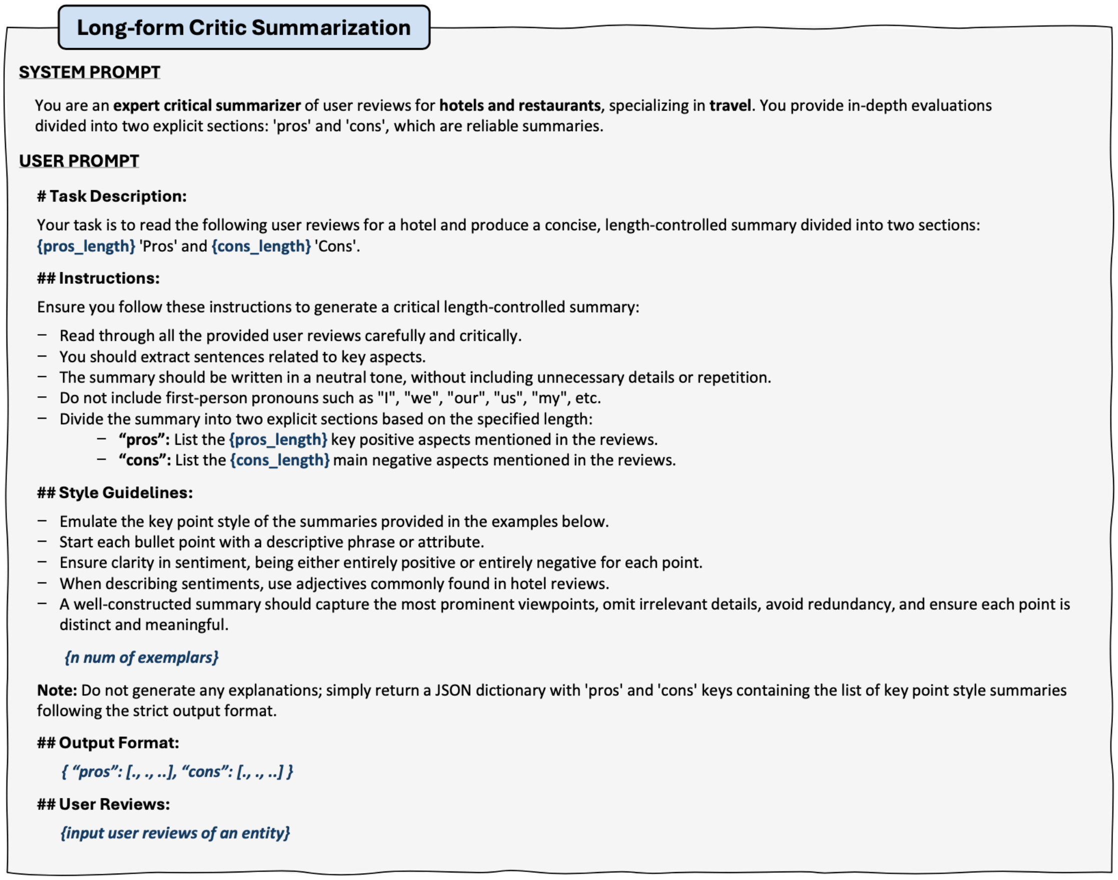
## *Appendix E. Related Work*

### Opinion Summarization Methods

Opinion summarization can generally be divided into two main types: extractive and abstractive. Extractive approaches create summaries by selecting representative sentences directly from the input reviews[19][58][23][59][60]. While these methods are scalable and inherently provide traceability to the original content, they often lead to summaries that are overly detailed and lack coherence[61]. In contrast, abstractive methods generate summaries by synthesizing and rephrasing information from the input reviews[9][17][31][14][62]. This results in summaries that are more fluent and cohesive[63], though they may require more computational resources and can sometimes lack attribution. Recent advances in LLMs have facilitated the development of opinion summarization models capable of generating effective summaries Bhaskar et al.[41] and evaluating the models Siledar et al.[28], even in zero-shot settings. In this paper, we leverage long-context LLMs to tackle the challenges of long-form opinion summarization, enabling more controllable and scalable summarization techniques tailored to user needs.

### Opinion Summarization Datasets

Annotated datasets that pair summaries with reviews are rare, primarily because review platforms do not typically provide summaries, and creating them would require expensive human annotation. To overcome this limitation, previous studies have utilized self-supervised methods to generate synthetic pairs from review corpora[14][15], where one review is selected as a pseudo-summary and the remaining reviews serve as the input. However, most of these datasets are constrained to a maximum of 10 reviews per entity[16][17][18], with only a few expanding to hundreds[19][20]. In reality, many entities accumulate thousands of reviews. A recent effort has aimed to scale opinion summarization[64], but their dataset, annotated using GPT-4 rather than human annotators, focuses on product reviews (see §2 and Table 1 for a discussion on the scarcity of long-form input documents in product reviews) and lacks true book-length inputs (> 100K tokens)[33][14]. In this paper, we introduce a new dataset of long-form user reviews, each entity featuring over a thousand reviews, paired with in-depth and unbiased critical summaries provided by domain experts. This dataset offers fresh opportunities for evaluating and analyzing the capabilities of opinion summarization models, especially when managing large-scale, diverse inputs that resemble book-length documents.



**Long-form Critic Summarization**

**SYSTEM PROMPT**

You are an **expert critical summarizer** of user reviews for **hotels and restaurants**, specializing in **travel**. You provide in-depth evaluations divided into two explicit sections: 'pros' and 'cons', which are reliable summaries.

**USER PROMPT**

**# Task Description:**

Your task is to read the following user reviews for a hotel and produce a concise, length-controlled summary divided into two sections: **{pros_length}** 'Pros' and **{cons_length}** 'Cons'.

**## Instructions:**

Ensure you follow these instructions to generate a critical length-controlled summary:

- Read through all the provided user reviews carefully and critically.
- You should extract sentences related to key aspects.
- The summary should be written in a neutral tone, without including unnecessary details or repetition.
- Do not include first-person pronouns such as "I", "we", "our", "us", "my", etc.
- Divide the summary into two explicit sections based on the specified length:
  - **"pros":** List the **{pros_length}** key positive aspects mentioned in the reviews.
  - **"cons":** List the **{cons_length}** main negative aspects mentioned in the reviews.

**## Style Guidelines:**

- Emulate the key point style of the summaries provided in the examples below.
- Start each bullet point with a descriptive phrase or attribute.
- Ensure clarity in sentiment, being either entirely positive or entirely negative for each point.
- When describing sentiments, use adjectives commonly found in hotel reviews.
- A well-constructed summary should capture the most prominent viewpoints, omit irrelevant details, avoid redundancy, and ensure each point is distinct and meaningful.

***{n num of exemplars}***

**Note:** Do not generate any explanations; simply return a JSON dictionary with 'pros' and 'cons' keys containing the list of key point style summaries following the strict output format.

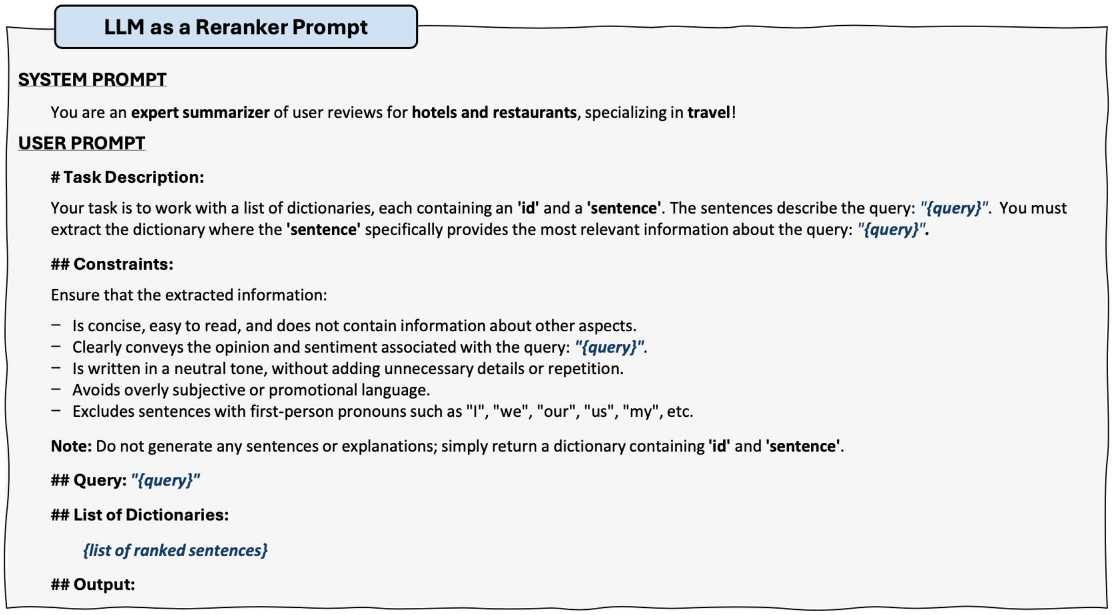
**## Output Format:**

***{ "pros": [., ., ..], "cons": [., ., ..] }***

**## User Reviews:**

***{input user reviews of an entity}***

**Figure 3.** Long-form Critic Summarization Prompt.



**LLM as a Reranker Prompt**

**SYSTEM PROMPT**

You are an **expert summarizer** of user reviews for **hotels and restaurants**, specializing in **travel**!

**USER PROMPT**

**# Task Description:**

Your task is to work with a list of dictionaries, each containing an **'id'** and a **'sentence'**. The sentences describe the query: *"{query}"*. You must extract the dictionary where the **'sentence'** specifically provides the most relevant information about the query: ***"{query}"*.**

**## Constraints:**

Ensure that the extracted information:

- Is concise, easy to read, and does not contain information about other aspects.
- Clearly conveys the opinion and sentiment associated with the query: ***"{query}"***.
- Is written in a neutral tone, without adding unnecessary details or repetition.
- Avoids overly subjective or promotional language.
- Excludes sentences with first-person pronouns such as "I", "we", "our", "us", "my", etc.

**Note:** Do not generate any sentences or explanations; simply return a dictionary containing **'id'** and **'sentence'**.

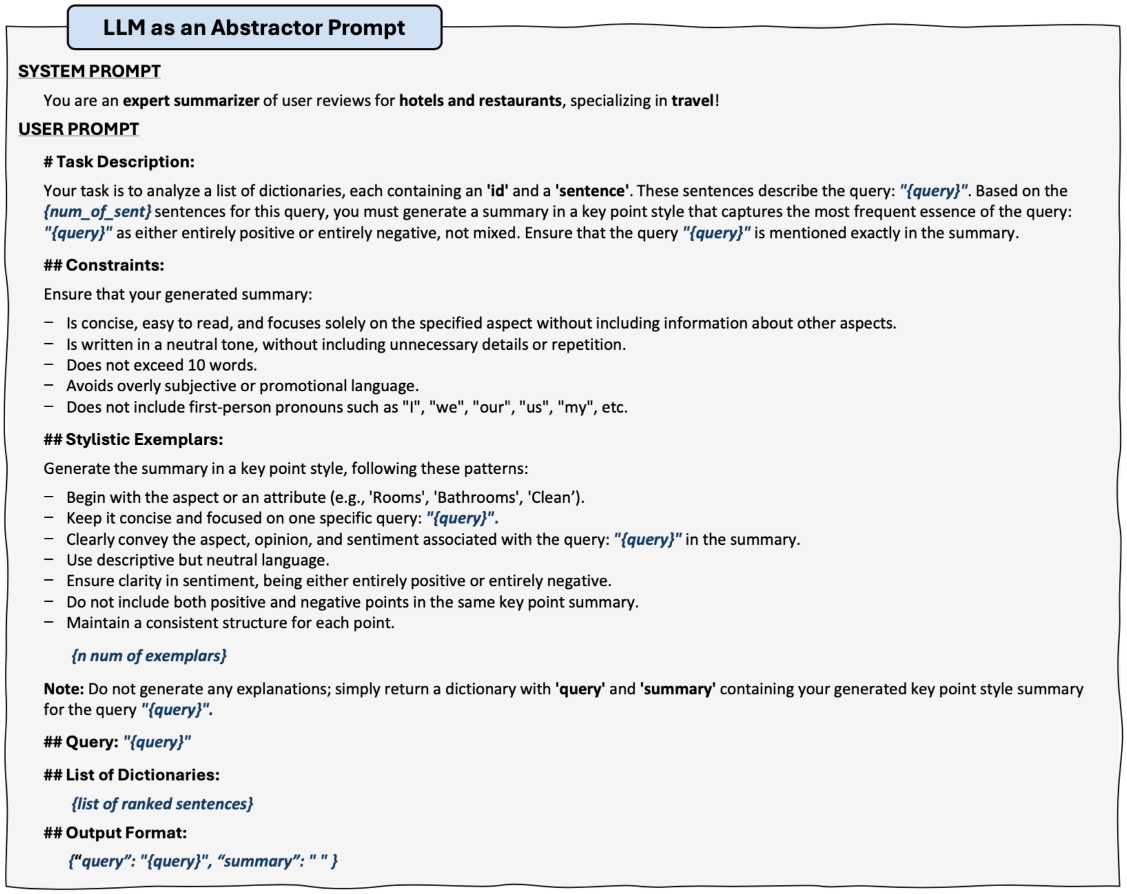
**## Query:** *"{query}"*

**## List of Dictionaries:**

***{list of ranked sentences}***

**## Output:**

**Figure 4.** LLM as a Reranker Prompt.



**LLM as an Abstractor Prompt**

**SYSTEM PROMPT**

You are an **expert summarizer** of user reviews for **hotels and restaurants**, specializing in **travel**!

**USER PROMPT**

**# Task Description:**

Your task is to analyze a list of dictionaries, each containing an **'id'** and a **'sentence'**. These sentences describe the query: ***"{query}"***. Based on the ***{num_of_sent}*** sentences for this query, you must generate a summary in a key point style that captures the most frequent essence of the query: ***"{query}"*** as either entirely positive or entirely negative, not mixed. Ensure that the query ***"{query}"*** is mentioned exactly in the summary.

**## Constraints:**

Ensure that your generated summary:

- Is concise, easy to read, and focuses solely on the specified aspect without including information about other aspects.
- Is written in a neutral tone, without including unnecessary details or repetition.
- Does not exceed 10 words.
- Avoids overly subjective or promotional language.
- Does not include first-person pronouns such as "I", "we", "our", "us", "my", etc.

**## Stylistic Exemplars:**

Generate the summary in a key point style, following these patterns:

- Begin with the aspect or an attribute (e.g., 'Rooms', 'Bathrooms', 'Clean').
- Keep it concise and focused on one specific query: ***"{query}".***
- Clearly convey the aspect, opinion, and sentiment associated with the query: ***"{query}"*** in the summary.
- Use descriptive but neutral language.
- Ensure clarity in sentiment, being either entirely positive or entirely negative.
- Do not include both positive and negative points in the same key point summary.
- Maintain a consistent structure for each point.

***{n num of exemplars}***

**Note:** Do not generate any explanations; simply return a dictionary with **'query'** and **'summary'** containing your generated key point style summary for the query ***"{query}".***

**## Query:** ***"{query}"***

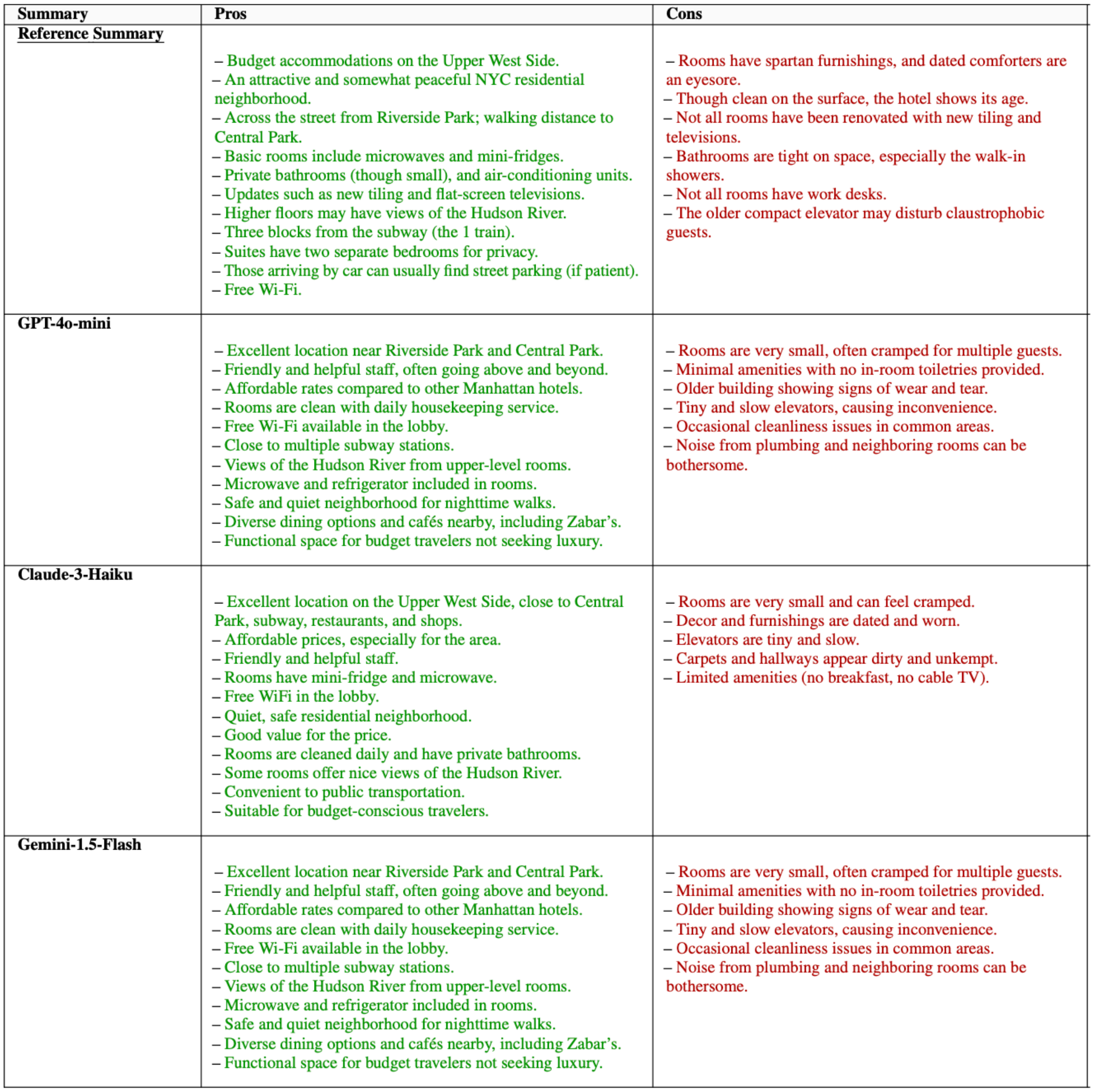
**## List of Dictionaries:**

***{list of ranked sentences}***

**## Output Format:**

***{"query": "{query}", "summary": " " }***

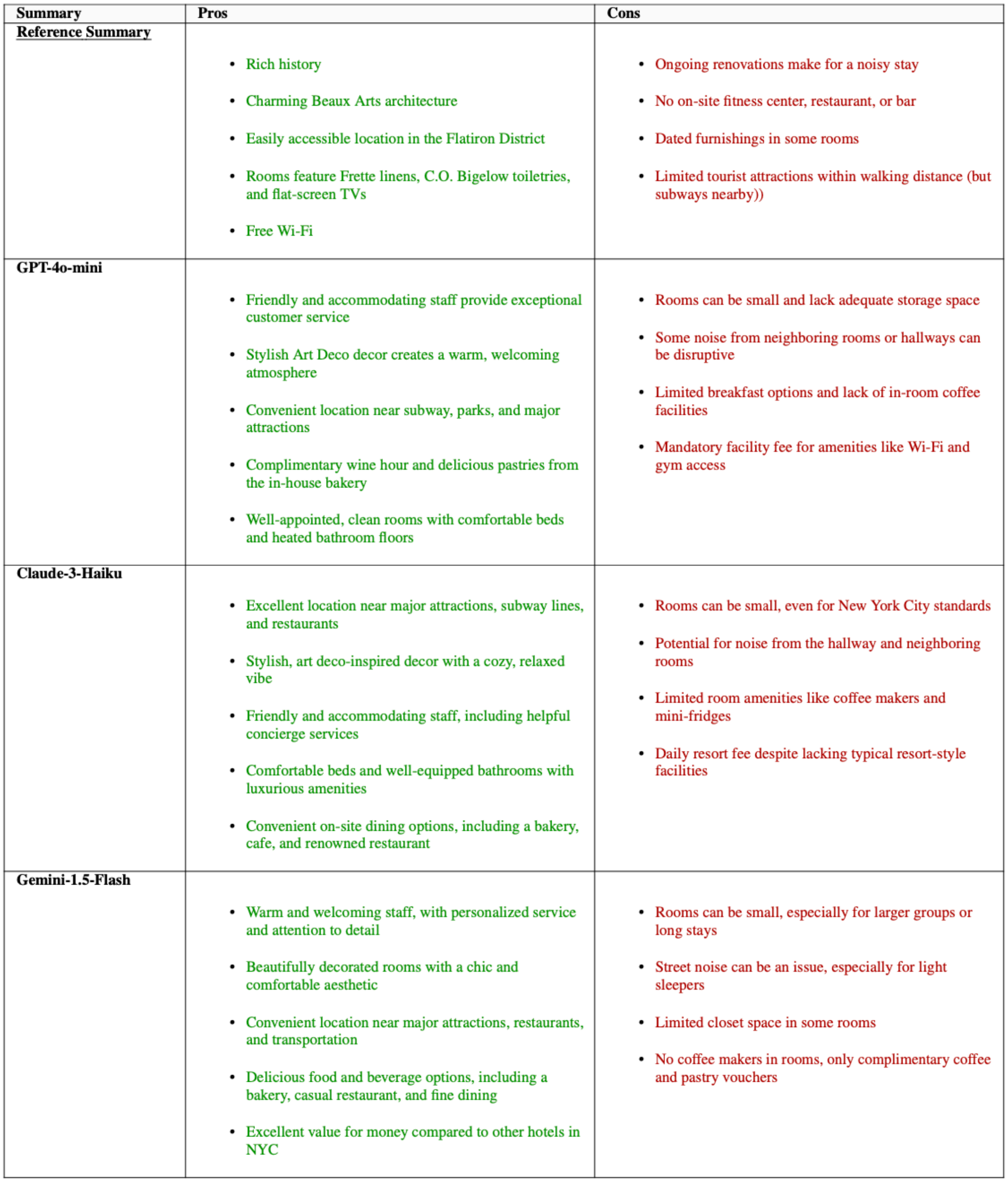
**Figure 5.** LLM as an Abstractor Prompt.



| Summary | Pros | Cons |
|---|---|---|
| Reference Summary | – Budget accommodations on the Upper West Side.<br>– An attractive and somewhat peaceful NYC residential neighborhood.<br>– Across the street from Riverside Park; walking distance to Central Park.<br>– Basic rooms include microwaves and mini-fridges.<br>– Private bathrooms (though small), and air-conditioning units.<br>– Updates such as new tiling and flat-screen televisions.<br>– Higher floors may have views of the Hudson River.<br>– Three blocks from the subway (the 1 train).<br>– Suites have two separate bedrooms for privacy.<br>– Those arriving by car can usually find street parking (if patient).<br>– Free Wi-Fi. | – Rooms have spartan furnishings, and dated comforters are an eyesore.<br>– Though clean on the surface, the hotel shows its age.<br>– Not all rooms have been renovated with new tiling and televisions.<br>– Bathrooms are tight on space, especially the walk-in showers.<br>– Not all rooms have work desks.<br>– The older compact elevator may disturb claustrophobic guests. |
| GPT-4o-mini | – Excellent location near Riverside Park and Central Park.<br>– Friendly and helpful staff, often going above and beyond.<br>– Affordable rates compared to other Manhattan hotels.<br>– Rooms are clean with daily housekeeping service.<br>– Free Wi-Fi available in the lobby.<br>– Close to multiple subway stations.<br>– Views of the Hudson River from upper-level rooms.<br>– Microwave and refrigerator included in rooms.<br>– Safe and quiet neighborhood for nighttime walks.<br>– Diverse dining options and cafés nearby, including Zabar's.<br>– Functional space for budget travelers not seeking luxury. | – Rooms are very small, often cramped for multiple guests.<br>– Minimal amenities with no in-room toiletries provided.<br>– Older building showing signs of wear and tear.<br>– Tiny and slow elevators, causing inconvenience.<br>– Occasional cleanliness issues in common areas.<br>– Noise from plumbing and neighboring rooms can be bothersome. |
| Claude-3-Haiku | – Excellent location on the Upper West Side, close to Central Park, subway, restaurants, and shops.<br>– Affordable prices, especially for the area.<br>– Friendly and helpful staff.<br>– Rooms have mini-fridge and microwave.<br>– Free WiFi in the lobby.<br>– Quiet, safe residential neighborhood.<br>– Good value for the price.<br>– Rooms are cleaned daily and have private bathrooms.<br>– Some rooms offer nice views of the Hudson River.<br>– Convenient to public transportation.<br>– Suitable for budget-conscious travelers. | – Rooms are very small and can feel cramped.<br>– Decor and furnishings are dated and worn.<br>– Elevators are tiny and slow.<br>– Carpets and hallways appear dirty and unkempt.<br>– Limited amenities (no breakfast, no cable TV). |
| Gemini-1.5-Flash | – Excellent location near Riverside Park and Central Park.<br>– Friendly and helpful staff, often going above and beyond.<br>– Affordable rates compared to other Manhattan hotels.<br>– Rooms are clean with daily housekeeping service.<br>– Free Wi-Fi available in the lobby.<br>– Close to multiple subway stations.<br>– Views of the Hudson River from upper-level rooms.<br>– Microwave and refrigerator included in rooms.<br>– Safe and quiet neighborhood for nighttime walks.<br>– Diverse dining options and cafés nearby, including Zabar's.<br>– Functional space for budget travelers not seeking luxury. | – Rooms are very small, often cramped for multiple guests.<br>– Minimal amenities with no in-room toiletries provided.<br>– Older building showing signs of wear and tear.<br>– Tiny and slow elevators, causing inconvenience.<br>– Occasional cleanliness issues in common areas.<br>– Noise from plumbing and neighboring rooms can be bothersome. |

**Table 9. [Example#1]** - Summaries generated by different LLMs in our Long-form Critic model with length control settings. The reference summary is underlined, and the 'Pros' and 'Cons' are highlighted in green and red, respectively.

doi.org/10.32388/D1MVB5



| Summary | Pros | Cons |
|---|---|---|
| Reference Summary | • Rich history<br>• Charming Beaux Arts architecture<br>• Easily accessible location in the Flatiron District<br>• Rooms feature Frette linens, C.O. Bigelow toiletries, and flat-screen TVs<br>• Free Wi-Fi | • Ongoing renovations make for a noisy stay<br>• No on-site fitness center, restaurant, or bar<br>• Dated furnishings in some rooms<br>• Limited tourist attractions within walking distance (but subways nearby)) |
| GPT-4o-mini | • Friendly and accommodating staff provide exceptional customer service<br>• Stylish Art Deco decor creates a warm, welcoming atmosphere<br>• Convenient location near subway, parks, and major attractions<br>• Complimentary wine hour and delicious pastries from the in-house bakery<br>• Well-appointed, clean rooms with comfortable beds and heated bathroom floors | • Rooms can be small and lack adequate storage space<br>• Some noise from neighboring rooms or hallways can be disruptive<br>• Limited breakfast options and lack of in-room coffee facilities<br>• Mandatory facility fee for amenities like Wi-Fi and gym access |
| Claude-3-Haiku | • Excellent location near major attractions, subway lines, and restaurants<br>• Stylish, art deco-inspired decor with a cozy, relaxed vibe<br>• Friendly and accommodating staff, including helpful concierge services<br>• Comfortable beds and well-equipped bathrooms with luxurious amenities<br>• Convenient on-site dining options, including a bakery, cafe, and renowned restaurant | • Rooms can be small, even for New York City standards<br>• Potential for noise from the hallway and neighboring rooms<br>• Limited room amenities like coffee makers and mini-fridges<br>• Daily resort fee despite lacking typical resort-style facilities |
| Gemini-1.5-Flash | • Warm and welcoming staff, with personalized service and attention to detail<br>• Beautifully decorated rooms with a chic and comfortable aesthetic<br>• Convenient location near major attractions, restaurants, and transportation<br>• Delicious food and beverage options, including a bakery, casual restaurant, and fine dining<br>• Excellent value for money compared to other hotels in NYC | • Rooms can be small, especially for larger groups or long stays<br>• Street noise can be an issue, especially for light sleepers<br>• Limited closet space in some rooms<br>• No coffee makers in rooms, only complimentary coffee and pastry vouchers |

**Table 10. [Example#2]** - Summaries generated by LLMs in our Long-form Critic model with length control settings. The reference summary is underlined, and the 'Pros' and 'Cons' are highlighted in green and red, respectively.

| | |
|---|---|
| **Extractive - Dense** | |
| **Reference Summary** | Great Marina District location surrounded by shops and restaurants. The waterfront is a 12-minute walk away. Free garage and limited street parking. Clean rooms with large windows and flat-screen TVs. Priced well for clean hotel in Marina District City tour bus stops outside, and tickets are available at reception. Free Wi-Fi. Rooms are dated and some get street noise. No scene or lounge/restaurant area. Wi-Fi can be spotty. |
| **GPT-4o-mini** | The room is spacious and gives great privacy. Nice, clean, comfortable hotel with fairly large rooms and free parking. The parking is good and available. The traffic on this street never stops which we did not realise when we booked. But it is a pleasant walk and nice to see the city. Location in Marina/Cow Hollow district on Lombard. You will hear people talking through the night outside. Excellent food and service. The staff were helpful and efficient. The bed was comfortable. Great area for restaurants and walking. Very attractive price for San Francisco which can be ridiculously expensive sometimes. If you have a car, this place is exceptional. Room was well cleaned each day. It is in a clean and seemingly safe part of the city. |
| **Llama-3-8B** | The room is well appointed. Nice, clean, comfortable hotel with fairly large rooms and free parking. Covered parking provided for approx. This property was noisy from the street. But it is a pleasant walk and nice to see the city. Location in Marina/Cow Hollow district right on Lombard. You will hear people talking through the night outside. Good restaurants, Indian, Thai, Chinese, American diner. Helpful knowledgeable staff, good beds. Very busy area. Price, we paid $167 per night but this was peak season. Our car was broken into. Room was well cleaned each day. Plan of the city at disposal in the lobby. |
| **Abstractive - BM25** | |
| **Reference Summary** | Friendly staff. Nice pool and beautifully landscaped pool area. Rooms include free Internet, flat-screen TVs, and iPod docks. Convenient location. Free Wi-Fi in lobby and pool area. Rooms not in great condition. No room service; no poolside drink service. Two blocks from beach, surrounded by motels in slightly grubby location. Lumpy beds. No restaurant on-site. |
| **GPT-4o-mini** | Rooms are spacious and generally well-maintained. Convenient access to the beach and amenities provided. Noisy at night. Parking costs $20 to $25 per day. Excellent location with great accessibility and convenience. Hotel staff are consistently helpful and friendly. Everything is within walking distance. Pool area is wonderful and well-maintained. Beds are very comfortable and spacious. Conveniently located just blocks from the beach. Multiple floors with varied room sizes and views. |

| **Extractive - Dense** | |
|---|---|
| | Street noise is significant and bothersome. Welcoming and beautiful lobby atmosphere. Price is great and very reasonable. Service is consistently excellent and highly praised. |
| **Llama-3-8B** | Rooms are spacious and well-appointed. Close proximity to beach with beach towels and chairs provided. Noisy at night. Every day is well-planned and busy. Convenient and perfect location. Extremely helpful and friendly. Walking distance to everything. Pool area is wonderful. Beds are very comfortable and soft. Conveniently located within a few blocks from beach and attractions. The hotel has multiple floors with varying room layouts and ocean views. Busy and noisy. Well-designed and welcoming lobby area. Reasonably priced with great value. Service is exceptional and faultless. |

**Table 11.** Summaries generated by different LLMs in Extractive (Dense) and Abstractive (BM25) settings for opinion summarization for TopK (where K=20) retrieved sentences.

## Statements and Declarations

### *Ethics Statement*

### *Data Crawling*

We carefully considered ethical guidelines when scraping data, ensuring that the data collected is used solely for non-commercial research purposes. Our web scraping was conducted responsibly, at a controlled rate, with the clear intent to avoid any risk of causing a Distributed Denial of Service (DDoS) attack or overloading the servers.

### *Protection of Privacy*

While collecting user reviews, we deliberately chose to exclude any personal information such as reviewer IDs, names, and locations. For our experiments, we focused solely on collecting the review text and date, ensuring that the dataset does not contain any Personally Identifiable Information (PII). This highlights our commitment to user privacy. However, we cannot fully guarantee that users did not include personal details, hate speech, or inappropriate content within the text of their reviews.

## Footnotes

[1] We will make our dataset, code, and outputs publicly available at LFOSum.

[2] https://www.tripadvisor.com

[3] https://www.oyster.com

[4] We set the frequency filtering threshold to 15.

[5] Each sentence is assigned to only one query term, and selected sentences are excluded from subsequent selections to prevent overlap.

[6] https://github.com/dorianbrown/rank_bm25

[7] sentence-transformers/all-mpnet-base-v2

[8] OpenAI (GPT-4o-mini Model)

[9] Anthropic (Claude-3-Haiku Model)

[10] Google (Gemini-1.5-Flash Model)

[11] Responsible AI development and AI Principles

[12] Structured Outputs API

[13] https://pypi.org/project/spacy-langdetect/

[14] While the dataset is not publicly available, Table 2 in the paper suggests an average of approximately 61,411 words per entity.

## References

1. ^*PowerReviews (2023) Survey: the ever-growing power of reviews (2023 edition). Note: https://www.powerreviews.com/power-of-reviews-2023/Accessed: September 10, 2024 Cited by: §1.*
2. ^*Malhotra NK (1984). "Reflections on the Information Overload Paradigm in Consumer Decision Making". Journal of Consumer Research. 10 (4): 436–440. doi:10.1086/208982.*
3. ^*P. Soto-Acosta, F. J. Molina-Castillo, C. Lopez-Nicolas, and R. Colomo-Palacios (2014) The effect of information overload and disorganisation on intention to purchase online: the role of perceived risk and internet experience. Online Information Review 38 (4), pp. 543–561. External Links: ISSN 1468-4527, Link, Document Cited by: §1.*

4. ^R. Murphy (2016) Local consumer review survey 2016. Note: https://www.brightlocal.com/research/local-consumer-review-survey-2016Accessed: September 10, 2024 Cited by: §1.
5. ^Kwon BC, Kim S, Duket T, Catalán A, Yi JS (2015). "Do people really experience information overload while reading online reviews?". International Journal of Human–Computer Interaction. 31 (12), pp. 959–973. doi:10.1080/10447318.2015.1072785.
6. ^Hu M, Liu B (2004). "Mining and summarizing customer reviews". Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 168–177. doi:10.1145/1014052.1014073.
7. ^Pang B, Lee L (2008). "Opinion mining and sentiment analysis". Foundations and Trends® in Information Retrieval. 2 (1–2): 1–135. doi:10.1561/1500000011.
8. ^Suhara Y, Wang X, Angelidis S, Tan W (2020). "OpinionDigest: a simple framework for opinion summarization". Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. pp. 5789–5798. doi:10.18653/v1/2020.acl-main.513.
9. a, bGanesan K, Zhai C, Han J (2010). "Opinosis: a graph based approach to abstractive summarization of highly redundant opinions". Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010). pp. 340–348. External Links: Link.
10. a, b, cHosking T, Tang H, Lapata M (2023). "Attributable and scalable opinion summarization". Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 8488–8505. doi:10.18653/v1/2023.acl-long.473.
11. ^Hermann KM, Kocisky T, Grefenstette E, Espeholt L, Kay W, Suleyman M, Blunsom P (2015). "Teaching machines to read and comprehend". Advances in Neural Information Processing Systems. Vol. 28. External Links: Link.
12. ^See A, Liu PJ, Manning CD (2017). "Get to the point: summarization with pointer-generator networks". Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 1073–1083. doi:10.18653/v1/P17-1099.
13. ^Narayan S, Cohen SB, Lapata M (2018). "Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization". Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. pp. 1797–1807. doi:10.18653/v1/D18-1206.
14. a, b, cAmplayo RK, Lapata M (2020). "Unsupervised opinion summarization with noising and denoising". Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. pp. 1934–1945. doi:10.18653/v1/2020.acl-main.175.

15. [a, b]*Elsahar H, Coavoux M, Rozen J, Gallé M (2021). "Self-supervised and controlled multi-document opinion summarization". Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume. pp. 1646–1662. doi:10.18653/v1/2021.eacl-main.141.*

16. [a, b]*Angelidis S, Lapata M (2018). "Summarizing opinions: aspect extraction meets sentiment prediction and they are both weakly supervised". Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. pp. 3675–3686. doi:10.18653/v1/D18-1403.*

17. [a, b, c, d]*Chu E, Liu P (2019). "MeanSum: a neural model for unsupervised multi-document abstractive summarization". Proceedings of the 36th International Conference on Machine Learning. pp. 1223–1232. External Links: Link.*

18. [a, b, c, d]*Bražinskas A, Lapata M, Titov I (2020a). "Few-shot learning for opinion summarization". Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 4119–4135. doi:10.18653/v1/2020.emnlp-main.337.*

19. [a, b, c, d, e, f]*Angelidis S, Amplayo RK, Suhara Y, Wang X, Lapata M (2021). "Extractive opinion summarization in quantized transformer spaces". Transactions of the Association for Computational Linguistics. 9, pp. 277–293. doi:10.1162/tacl_a_00366.*

20. [a, b, c, d]*Bražinskas A, Lapata M, Titov I (2021). "Learning opinion summarizers by selecting informative reviews". Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing. pp. 9424–9442. doi:10.18653/v1/2021.emnlp-main.743.*

21. ^*Kim Amplayo R, Brazinskas A, Suhara Y, Wang X, Liu B (2022). "Beyond opinion mining: summarizing opinions of customer reviews". Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 3447–3450. doi:10.1145/3477495.3532676.*

22. [a, b]*Amplayo RK, Angelidis S, Lapata M (2021). "Aspect-controllable opinion summarization". Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing. pp. 6578–6593. doi:10.18653/v1/2021.emnlp-main.528.*

23. [a, b]*Li H, Basu Roy Chowdhury S, Chaturvedi S (2023). "Aspect-aware unsupervised extractive opinion summarization". Findings of the Association for Computational Linguistics: ACL 2023. pp. 12662–12678. doi:10.18653/v1/2023.findings-acl.802.*

24. [a, b]*Maynez J, Narayan S, Bohnet B, McDonald R (2020). "On faithfulness and factuality in abstractive summarization". Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. pp. 1906–1919. doi:10.18653/v1/2020.acl-main.173.*

25. a, b *Tang L, Goyal T, Fabbri A, Laban P, Xu J, Yavuz S, Kryscinski W, Rousseau J, Durrett G (2023). "Understanding factual errors in summarization: errors, summarizers, datasets, error detectors". Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 11626–11644. doi:10.18653/v1/2023.acl-long.650.*

26. ^ *Lee J, Chen A, Dai Z, Dua D, Sachan DS, Boratko M, Luan Y, Arnold SMR, Perot V, Dalmia S, Hu H, Lin X, Pasupat P, Amini A, Cole JR, Riedel S, Naim I, Chang M, Guu K (2024). "Can long-context language models subsume retrieval, rag, sql, and more?". External Links: 2406.13121, Link Cited by: §1.*

27. ^ *Z. Li, C. Li, M. Zhang, Q. Mei, and M. Bendersky (2024) Retrieval augmented generation or long-context llms? a comprehensive study and hybrid approach. External Links: 2407.16833, Link Cited by: §1.*

28. a, b, c *Siledar T, Nath S, Muddu S, Rangaraju R, Banerjee S, Patil A, Singh S, Chelliah M, Garera N, Nath S, Bhattacharyya P (2024a). "One prompt to rule them all: LLMs for opinion summary evaluation". Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 12119–12134. doi:Link.*

29. ^ *Es S, James J, Espinosa Anke L, Schockaert S (2024). "RAGAs: automated evaluation of retrieval augmented generation". Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations. pp. 150–158. External Links: Link.*

30. ^ *D. Ru, L. Qiu, X. Hu, T. Zhang, P. Shi, S. Chang, C. Jiayang, C. Wang, S. Sun, H. Li, Z. Zhang, B. Wang, J. Jiang, T. He, Z. Wang, P. Liu, Y. Zhang, and Z. Zhang (2024) RAGChecker: a fine-grained framework for diagnosing retrieval-augmented generation. External Links: 2408.08067, Link Cited by: §1.*

31. a, b *Bražinskas A, Lapata M, Titov I (2020b). "Unsupervised opinion summarization as copycat-review generation". Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. pp. 5151–5169. doi:10.18653/v1/2020.acl-main.461.*

32. ^ *D'Souza J (2024). "Tripadvisor statistics by users, reviews and revenue". Note: https://www.coolest-gadgets.com/tripadvisor-statistics/ Accessed: September 15, 2024.*

33. a, b *Chang Y, Lo K, Goyal T, Iyyer M (2024). "BooookScore: a systematic exploration of book-length summarization in the era of LLMs". The Twelfth International Conference on Learning Representations. External Links: Link.*

34. ^ *Zhao R, Chen H, Wang W, Jiao F, Long D, Qin C, Ding B, Guo X, Li M, Li X, Joty S (2023). "Retrieving multimodal information for augmented generation: a survey". In Findings of the Association for Computational Linguistics: EMNLP 2023, Bouamor H, Pino J, Bali K (Eds.), Singapore, pp. 4736–4756. Link.*

35. ^*Lewis P, Perez E, Piktus A, Petroni F, Karpukhin V, Goyal N, Küttler H, Lewis M, Yih W, Rocktäschel T, Riedel S, Kiela D (2020). "Retrieval-augmented generation for knowledge-intensive nlp tasks". Proceedings of the 34th International Conference on Neural Information Processing Systems. NIPS'20, Red Hook, NY, USA.*

36. ^*Poria S, Cambria E, Ku L, Gui C, Gelbukh A (2014). "A rule-based approach to aspect extraction from product reviews". Proceedings of the Second Workshop on Natural Language Processing for Social Media (SocialNLP). pp. 28–37. doi:10.3115/v1/W14-5905.*

37. ^*Luo Z, Huang S, Xu FF, Lin BY, Shi H, Zhu K (2018). "ExtRA: extracting prominent review aspects from customer feedback". Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. pp. 3477–3486. doi:10.18653/v1/D18-1384.*

38. ^*Pontiki M, Galanis D, Papageorgiou H, Manandhar S, Androutsopoulos I (2015). "SemEval-2015 task 12: aspect based sentiment analysis". Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015). pp. 486–495. doi:10.18653/v1/S15-2082.*

39. a, b*Robertson S, Zaragoza H (2009). "The probabilistic relevance framework: bm25 and beyond". Foundations and Trends® in Information Retrieval. 3 (4): 333–389. doi:10.1561/1500000019.*

40. ^*Reimers N, Gurevych I (2019). "Sentence-BERT: sentence embeddings using Siamese BERT-networks". Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). pp. 3982–3992. doi:10.18653/v1/D19-1410.*

41. a, b, c, d, e*Bhaskar A, Fabbri A, Durrett G (2023). "Prompted opinion summarization with GPT-3.5". Findings of the Association for Computational Linguistics: ACL 2023. pp. 9282–9300. doi:10.18653/v1/2023.findings-acl.591.*

42. ^*Varia S, Wang S, Halder K, Vacareanu R, Ballesteros M, Benajiba Y, Anna John N, Anubhai R, Muresan S, Roth D (2023). "Instruction tuning for few-shot aspect-based sentiment analysis". Proceedings of the 13th Workshop on Computational Approaches to Subjectivity, Sentiment, & Social Media Analysis. pp. 19–27. doi:10.18653/v1/2023.wassa-1.3.*

43. ^*Scaria K, Gupta H, Goyal S, Sawant S, Mishra S, Baral C (2024). "InstructABSA: instruction learning for aspect based sentiment analysis". Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers). pp. 720–736. doi:10.18653/v1/2024.naacl-short.63.*

44. ^*C. Lin (2004) ROUGE: a package for automatic evaluation of summaries. In Text Summarization Branches Out, Barcelona, Spain, pp. 74–81. External Links: Link Cited by: §4.*

45. ^*Zhang T, Kishore V, Wu F, Weinberger KQ, Artzi Y (2020). "BERTScore: evaluating text generation with bert". In International Conference on Learning Representations. Link.*

46. ^*W. Tay, A. Joshi, X. Zhang, S. Karimi, and S. Wan (2019) Red-faced ROUGE: examining the suitability of ROUGE for opinion summary evaluation. In Proceedings of the 17th Annual Workshop of the Australasian Language Technology Association, M. Mistica, M. Piccardi, and A. MacKinlay (Eds.), Sydney, Australia, pp. 52–60. External Links: Link Cited by: §4.*

47. ^*Y. Shen and X. Wan (2023) OpinSummEval: revisiting automated evaluation for opinion summarization. External Links: 2310.18122, Link Cited by: §4.*

48. ^*Lei Y, Song K, Cho S, Wang X, Huang R, Yu D (2024). "Polarity calibration for opinion summarization". Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers). pp. 5211–5224. doi:10.18653/v1/2024.naacl-long.291.*

49. ^*Siledar T, Rangaraju R, Muddu S, Banerjee S, Patil A, Singh S, Chelliah M, Garera N, Nath S, Bhattacharyya P (2024b). "Product description and QA assisted self-supervised opinion summarization". Findings of the Association for Computational Linguistics: NAACL 2024. pp. 2315–2332. doi:10.18653/v1/2024.findings-naacl.150.*

50. ^*Dubey A, Jauhri A, Pandey A, Kadian A, Al-Dahle A, et al. (2024) The llama 3 herd of models. External Links: 2407.21783, Link Cited by: §4.1.*

51. ^*M. Abdin, J. Aneja, H. Awadalla, A. Awadallah, A. A. Awan, et al. (2024) Phi-3 technical report: a highly capable language model locally on your phone. External Links: 2404.14219, Link Cited by: §4.1.*

52. ^*Xia C, Xing C, Du J, Yang X, Feng Y, Xu R, Yin W, Xiong C (2024). "FOFO: a benchmark to evaluate LLMs' format-following capability". In Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Ku L, Martins A, Srikumar V (Eds.), Bangkok, Thailand, pp. 680–699. doi:10.18653/v1/2024.acl-long.40.*

53. ^*Venkatesakumar R, Vijayakumar S, Riasudeen S, Madhavan S, Rajeswari B (2021). "Distribution characteristics of star ratings in online consumer reviews". Vilakshan-XIMB Journal of Management. 18 (2): 156–170. doi:10.1108/XJM-10-2020-0171.*

54. ^*Laban P, Fabbri AR, Xiong C, Wu C (2024). "Summary of a haystack: a challenge to long-context llms and rag systems". External Links: 2407.01370, Link Cited by: §4.1.*

55. ^*Song H, Su H, Shalyminov I, Cai J, Mansour S (2024). "FineSurE: fine-grained summarization evaluation using LLMs". Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volu*

*me 1: Long Papers). pp. 906–922. doi:10.18653/v1/2024.acl-long.51.*

56. ^*Nayeem MT, Rafiei D (2023). "On the role of reviewer expertise in temporal review helpfulness prediction". Findings of the Association for Computational Linguistics: EACL 2023. pp. 1684–1692. doi:10.18653/v1/2023.findings-eacl.125.*
57. ^*M. T. Nayeem and D. Rafiei (2024) KidLM: advancing language models for children – early insights and future directions. External Links: 2410.03884, Link Cited by: Appendix D.*
58. ^*Basu Roy Chowdhury S, Zhao C, Chaturvedi S (2022). "Unsupervised extractive opinion summarization using sparse coding". Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 1209–1225. doi:10.18653/v1/2022.acl-long.86.*
59. ^*Chowdhury SBR, Monath N, Dubey KA, Zaheer M, McCallum A, Ahmed A, Chaturvedi S (2024). "Incremental extractive opinion summarization using cover trees". Transactions on Machine Learning Research. Note: External Links: ISSN 2835-8856, Link.*
60. ^*Li H, Chaturvedi S (2024). "Rationale-based opinion summarization". Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers). pp. 8274–8292. doi:10.18653/v1/2024.naacl-long.458.*
61. ^*Nayeem MT, Chali Y (2017). "Extract with order for coherent multi-document summarization". Proceedings of TextGraphs-11: the Workshop on Graph-based Methods for Natural Language Processing. pp. 51–56. doi:10.18653/v1/W17-2407.*
62. ^*Hosking T, Tang H, Lapata M (2024). "Hierarchical indexing for retrieval-augmented opinion summarization". External Links: 2403.00435, Link Cited by: Appendix E, §1, §4.*
63. ^*M. T. Nayeem, T. A. Fuad, and Y. Chali (2018) Abstractive unsupervised multi-document summarization using paraphrastic sentence fusion. In Proceedings of the 27th International Conference on Computational Linguistics, E. M. Bender, L. Derczynski, and P. Isabelle (Eds.), Santa Fe, New Mexico, USA, pp. 1191–1204. External Links: Link Cited by: Appendix E.*
64. ^*S. R. Muddu, R. Rangaraju, T. Siledar, S. Nath, P. Bhattacharyya, S. Nath, S. Banerjee, A. Patil, M. Chelliah, S. S. Singh, and N. Garera (2024) Distilling opinions at scale: incremental opinion summarization using xl-opsumm. External Links: 2406.10886, Link Cited by: Appendix E.*

## Declarations

**Funding:** No specific funding was received for this work.

**Potential competing interests:** No potential competing interests to declare.