FineSurE: Fine-grained Summarization Evaluation using LLMs

Hwanjun Song[1]*, **Hang Su**[2,†], **Igor Shalyminov**[2,†], **Jason Cai**[2,†], **Saab Mansour**[2,†]
[1]Korea Advanced Institute of Science and Technology
[2]AWS AI Labs
songhwanjun@kaist.ac.kr
{shawnsu, shalymin, cjinglun, saabm}@amazon.com

Abstract

Automated evaluation is crucial for streamlining text summarization benchmarking and model development, given the costly and time-consuming nature of human evaluation. Traditional methods like ROUGE do not correlate well with human judgment, while recently proposed LLM-based metrics provide only summary-level assessment using Likert-scale scores. This limits deeper model analysis, e.g., we can only assign one hallucination score at the summary level, while at the sentence level, we can count sentences containing hallucinations. To remedy those limitations, we propose FineSurE, a fine-grained evaluator specifically tailored for the summarization task using large language models (LLMs). It also employs completeness and conciseness criteria, in addition to faithfulness, enabling multi-dimensional assessment. We compare various open-source and proprietary LLMs as backbones for FineSurE. In addition, we conduct extensive benchmarking of FineSurE against SOTA methods including NLI-, QA-, and LLM-based methods, showing improved performance especially on the completeness and conciseness dimensions. The code is available at https://github.com/DISL-Lab/FineSurE-ACL24.

1 Introduction

Text summarization stands out as an important task in natural language processing, aiming to generate a condensed summary of a provided text while retaining its essential information (Gupta and Gupta, 2019; Song et al., 2023). Despite the enhanced quality of summaries produced by LLMs, the development of automated methods for evaluation remains a challenge (Kryściński et al., 2020; Maynez et al., 2020). Conventional *reference-based* metrics, such as ROUGE (Lin, 2004), have exhibited a *weak* correlation with actual human judgments (Liu et al., 2023). Consequently, human evaluation remains an essential step for accurately assessing the quality of generated summaries, even considering its inherent costs and time-consuming nature.

Recently, the need for better automatic evaluators has become an important research topic, aiming to streamline evaluation processes and ease manual efforts in model development (Gao et al., 2023). This effort provides valuable insights into whether generated summaries align with predefined quality standards, including aspects like faithfulness. Various approaches have been explored, including approaches based on neural language inference (NLI) (Laban et al., 2022) and question-answering (QA) (Fabbri et al., 2022; Zhong et al., 2022). In addition, LLMs have recently proven their potential to be an *automated* tool for human-like evaluation (Liu et al., 2023; Wang et al., 2023). The latest LLM-based method, G-Eval (Liu et al., 2023), demonstrated a Spearman correlation coefficient of over 0.5 with Likert-scale human judgments on the news domain using GPT-4.

Despite these advancements, we contend that the current LLM-based automated methods still fall short in achieving precise evaluation, primarily attributed to the *coarse-grained* evaluation pipeline and the *ambiguity* in evaluation dimensions. Specifically for coarse-grained evaluation, the evaluation dimensions–namely faithfulness, coherence, and relevance [1]–are frequently assessed at the summary-level, resulting in Likert-scale scores for each summary (Gao and Wan, 2022; Shen and Wan, 2023; Liu et al., 2023; Wang et al., 2023). This Likert-scale scoring method lacks fine-grained information on errors in generated summaries. For instance, it does not provide a breakdown of the number of summary sentences with quality issues or specify

*Corresponding Author.
†This work conducted independently and is not related to the author(s) position at Amazon.

[1]We omit fluency assessment as modern AI models typically generate highly fluent outputs (Liu et al., 2023).

Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 906–922

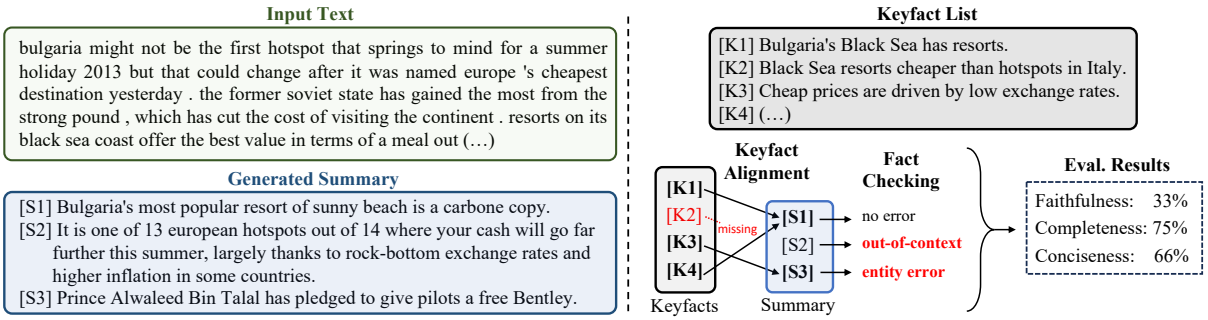

Figure 1: **FineSurE framework:** the given summary is evaluated by conducting the two tasks of fact checking and keyfact alignment. In this specific example, the faithfulness score is 33%, since only one out of the three summary sentences is factually correct; the completeness score is 75%, since three out of the four keyfacts align with the summary; and the conciseness score is 66%, since two out of the three sentences are related to the keyfacts.

the types of mistakes present in each sentence. Furthermore, regarding ambiguity, the evaluation of coherence and relevance is hindered by the lack of clarity in their definition of "the collective quality of all sentences" and "the selection of important content from the source" (Fabbri et al., 2021; Shen et al., 2023). Given that human can encounter challenges in evaluating summaries, it is inappropriate to expect a neural model to provide accurate and objective assessments. Hence, there is a need to develop a more precise evaluation framework that results in a more detailed assessment with clearly defined evaluation dimensions.

In this paper, we present **FineSurE** (**Fine**-grained **Su**mma**r**ization **E**valuation) using LLMs, a novel automated approach designed to evaluate the summarization quality at a *fine-grained* level based on summary sentences or keyfacts[2], as depicted in Figure 1. We aim to evaluate summaries using this framework along three vital criteria: the *faithfulness* of minimizing factuality errors, the *completeness* of encompassing the majority of keyfacts, and the *conciseness* of avoiding unnecessary details. Thus, our framework entails executing two finely grained procedures utilizing LLMs: **(1) fact checking** involves identifying specific factuality errors present in each summary sentence and **(2) keyfact alignment** focuses on aligning each keyfact with all summary sentences from which they are inferred. We leverage the outcomes from both procedures to calculate *precise percentage* scores, offering a more detailed assessment than Likert-scale scoring. This fine-grained approach enables us to analyze the quality issues of generated texts at both the sentence and keyfact-level.

[2]A keyfact refers to a concise sentence conveying a single key information, comprising at most 2-3 entities, also referred to as a semantic content unit (Bhandari et al., 2020). The keyfact list can be generated automatically or by humans.

On top of keyfact alignment, two dimensions (completeness and conciseness) can serve as better replacements for coherence and relevance (Fabbri et al., 2021), as they evaluate two key aspects of a good summary, assessing the comprehensive inclusion and density of key information while excluding irrelevant content.

To summarize, our main contributions are as follows: (1) we argue that LLM-based summarization suffers from hallucination, information emission and verbosity hence requiring revisiting the evaluation dimensions, (2) we suggest three metrics targeting LLM output characteristics and tackling the aforementioned problems including faithfulness, completeness and conciseness, (3) we propose a novel automated evaluation framework - FineSurE, based on keyfact lists and using LLMs to generate the keyfacts, align them to the summary sentences and categorize the errors automatically, (4) we compare various open-source and proprietary LLMs to power FineSurE and analyze their correlation to human judgment at the summary and system levels, (5) we provide comprehensive results comparing our method with similarity-based, NLI-based, QA-based, and LLM-based automated methods and show improved human correlation for FineSurE over state-of-the-art methods.

2 Related Work

Efforts to assess the quality of texts generated by language models have led to numerous initiatives in designing effective automated evaluators across various research directions (Lin, 2004; Fabbri et al., 2022; Zhong et al., 2022; Wang et al., 2023).

Similarity-based Evaluator. The evaluation of generated text can be measured by the n-gram overlap with the reference text, employing met-

rics such as ROUGE (Lin, 2004), BLEU (Papineni et al., 2002), and METEOR (Banerjee and Lavie, 2005). In contrast to relying on exact matches, several evaluators have leveraged token similarity through contextual embeddings like BERTScore (Zhang et al., 2019), MoverScore (Zhao et al., 2019), and BARTScore (Yuan et al., 2021). However, these evaluators lack human correlation and a multi-dimensional assessment of text quality akin to real human evaluation, as they typically produce a single-dimensional score based on text similarity.

To assess text quality, primarily focusing on checking factual consistency, task-specific evaluators utilizing NLI and QA have been explored.

NLI-based Evaluator. This task involves fact-checking and verification by retrieving relevant evidence from the input text to support the claim made in the generated text (Glover et al., 2022; Honovich et al., 2022; Utama et al., 2022). DAE (Goyal and Durrett, 2020) introduced the dependency arc entailment formulation, offering a more fine-grained approach to faithfulness evaluation. SummaC (Laban et al., 2022) presented a lightweight model that facilitates NLI by segmenting input text into sentence units and aggregating scores between pairs of sentences. Despite their enhanced performance, they only focus on assessing faithfulness.

QA-based Evaluator. This involves generating plausible questions from the reference text and then answering these questions considering the generated text (Scialom et al., 2021; Chen et al., 2021). QAGS (Wang et al., 2020) and QAFactEval (Fabbri et al., 2022) enhanced the accuracy of faithfulness evaluation, surpassing other similarity- and NLI-based evaluators in text summarization. UniEval (Zhong et al., 2022) proposed a unified evaluator capable of assessing multi-dimensional evaluation of text generation through the QA task. In text summarization, it evaluates four aspects: faithfulness, coherence, relevance, and fluency. Generally, these methods require training a neural model to generate questions and their corresponding answers.

LLM-based Evaluator. With the emergence of LLMs, there is a move toward utilizing them as reference-free automated evaluators in diverse scenarios (Shi et al., 2023; Lin and Chen, 2023; Chen et al., 2023; Fu et al., 2023). Recently, a few efforts have been made to evaluate faithfulness using edited text (Laban et al., 2023), atomic facts (Min et al., 2023), and external knowledge base (Feng et al., 2023), as well as to assess multi-dimensional aspects (Liu et al., 2023; Shen et al., 2023; Wang et al., 2023). Although LLMs have shown promise as evaluators, they currently lack a fine-grained assessment and primarily focus on addressing faithfulness, without considering other important dimensions for high-quality summaries.

Unlike prior studies, we define crucial aspects for a detailed evaluation using LLMs and introduce a new fine-grained evaluation framework called FineSurE. This framework addresses numerous open questions regarding the capabilities of LLMs, including sentence-level fact-checking, classification of error types, and keyfact-level alignment.

3 FineSurE Framework

3.1 Evaluation Dimensions

LLMs enhance the quality of summarization, but they rather suffer from hallucination, information emission and verbosity (Ji et al., 2023; Saito et al., 2023), requiring revisiting evaluation dimensions. Therefore, we advocate for a thorough assessment of the two evaluation criteria, "completeness" and "conciseness," in addition to "faithfulness." These two dimensions can effectively assess both information emission and verbosity while also complementing each other in evaluating information inclusion and summary succinctness.

- **Faithfulness**: The summarizer does not manipulate the information in the input text (i.e., intrinsic) and add any information not directly inferable from the input text (i.e., extrinsic).
- **Completeness**: The summarizer ensures the inclusion of all keyfacts from the input text in the output summary.
- **Conciseness**: The summarizer refrains from incorporating information outside the keyfacts in the output, maintaining a succinct and focused summary.

Note that, adhering to the precise definition of faithfulness in the recent work (Pagnoni et al., 2021), we categorize error types into a total of seven categories, with "out of context" as an extrinsic error, and "predicate," "entity," "circumstance," "coreference," "discourse link," and "grammatical" as intrinsic errors. See examples in Appendix A.

3.2 Evaluation Pipeline

We discuss the evaluation pipeline implementing the dimensions discussed previously. We employ LLMs as a tool to conduct fact checking and keyfact alignment tasks. Specifically, we design two

prompts tailored for the two tasks, as shown in Figures 3-4 of Appendix B. All prompts are customized to generate outputs in JSON format, enhancing the success ratio of following our instructions and facilitating parsing. The detailed analysis of the success ratio is provided in Section 4.2.

Task 1. Fact Checking. Figure 3 illustrates our prompt and its expected JSON output for fact checking. We convert the problem of fact checking into a *categorization problem* involving nine categories. These include the seven factuality errors, along with an additional category "other error" for errors outside the seven errors, and an additional category "no error" for cases where no error was detected. Therefore, given a pair of input text and model summary, the LLM is expected to output the *error type* classified into one of the nine categories for each sentence along with a concise reason.

Task 2. Keyfact Alignment. Figure 4 shows our prompt and its expected JSON output for keyfact alignment. We address the alignment problem through *keyfact matching*, a process that involves two sequential tasks: verifying if each keyfact is inferred from the summary and, if affirmative, specifying the line numbers for all the corresponding summary sentences. Thus, given a pair of keyfact list[3] and model summary, the output should be the binary label and the list of line numbers of all summary sentences matched for each keyfact.

Parsing and Scoring. The evaluation scores are computed using the results from the two tasks. Given a document D, let $S = \{s_1, \dots s_N\}$ be the generated summary with N sentences. By the fact checking task, we identify a subset of $S_{fact} \subseteq S$, which consists solely of summary sentences marked "no error". Then, the percentage score of faithfulness on S is determined by:

$$Faithfulness(D, S) = |S_{fact}|/|S|. \quad (1)$$

Let $K = \{k_1, \dots, k_M\}$ be the list of keyfacts with a size of M. Through the keyfact alignment, we construct a bipartite graph $M = (K, S, E)$, where the edge set $E = \{(k, s) : k \to s \mid k \in K \wedge s \in S\}$ and $k \to s$ indicates that the keyfact k aligns with the summary sentence s. Then, the percentage scores of completeness and conciseness on S are computed at the summary level by:

$$\begin{aligned} Completeness(K, S) &= \big|\{k|(k, s) \in E\}\big|/|K| \\ Conciseness(K, S) &= \big|\{s|(k, s) \in E\}\big|/|S|, \end{aligned} \quad (2)$$

where the operator $|\cdot|$ returns the number of unique items within the provided set. Intuitively, the two scores represent completeness, indicating the degree to which keyfacts are included in the summary, and conciseness, reflecting the density of relevant sentences aligning with given keyfacts. Moreover, unlike existing LLM-based methods (Liu et al., 2023; Wang et al., 2023; Shen et al., 2023), we provide more detailed information about the error type associated with each sentence and the alignment of each keyfact with summary sentences.

3.3 Prompt Engineering

We explore various prompt engineering strategies (Wei et al., 2022; Yu et al., 2023) to identify the most suitable one for our evaluation pipeline:

- **Basic Prompt**: A default question prompt in plain text, e.g., is the summary sentence supported by the transcript?
- **Instruction**: The prompt is provided using a step-by-step instruction using "Instruction:".
- **Categorization**: The prompt solves a categorization task by providing target categories.
- **Reasoning**: The prompt uses a chain-of-thought approach, incorporating a reasoning step.
- **Evidence Mapping**: The prompt requests an exact quote from the input to confirm the decision made by LLMs.

Combining all the above techniques was not always superior. Evaluation prompts are recommended to use instruction format with categorization and reasoning for faithfulness evaluation, as in Figure 3, and only instruction format for completeness and conciseness evaluation, as in Figure 4. See the detailed ablation in Appendix G.

3.4 Keyfact Extraction

The list of keyfacts is essential for evaluating the completeness and conciseness using FineSurE. Humans are best suited to generate these keyfacts as they understand the priorities in different domains, such as medicine or sales. However, in some cases, obtaining human keyfacts can be challenging. FineSurE works with human keyfacts by default, but for cases where no human keyfacts are provided, it can employ the LLM to extract keyfacts automatically. This process is entirely automated, utilizing prompts tailored for keyfact

[3]The list of keyfacts is provided by humans; if unavailable, it can be automatically derived from the reference summary. See Appendix C for details.

Direction	Method	Sentence-level bAcc (↑)	Summary-level Pearson Corr (↑)	Summary-level Spearman Corr (↑)	System-level Rank Corr (↑)
Similarity-based	ROUGE-1	Not Available	0.324 (0.00)	0.332 (0.00)	0.883 (0.00)
	ROUGE-2	Not Available	0.384 (0.00)	0.315 (0.00)	0.947 (0.00)
	ROUGE-L	Not Available	0.175 (0.00)	0.180 (0.00)	0.667 (0.05)
	BERTScore	Not Available	0.008 (0.69)	0.000 (0.97)	-0.133 (0.73)
	BARTScore	Not Available	0.717 (0.00)	0.736 (0.00)	0.937 (0.00)
NLI-based	SummaC-Conv	Not Available	0.828 (0.00)	0.814 (0.00)	0.883 (0.00)
QA-based	UniEval	Not Available	0.743 (0.00)	0.772 (0.00)	0.983 (0.00)
	QAFactEval	Not Available	0.841 (0.00)	0.813 (0.00)	0.933 (0.00)
LLM-based	G-Eval (GPT-4)	Not Available	**0.841** (0.00)	0.834 (0.00)	**0.950** (0.00)
	FineSurE (GPT-4)	**86.4%**	0.833 (0.00)	**0.839** (0.00)	**0.950** (0.00)

Table 1: Performance of **faithfulness evaluation on FRANK** using ten automated metrics at the sentence-, summary- and system-level. The values in parenthesis represent p-values. The best results are marked in bold.

extraction (see Figure 5). For further details, refer to Appendix C. The impact of employing automatic keyfact extraction on keyfact alignment is discussed in Section 4.1.2.

4 Evaluation

Datasets To evaluate the automated evaluator's performance, we need datasets with human annotations for sentence-level faithfulness and keyfact-level alignment. Since *no* single dataset includes both types of annotations, we opt for two separate datasets. FRANK (Pagnoni et al., 2021) is a benchmark dataset of 2,246 summaries for factuality evaluation metrics. It encompasses summaries of nine summarization systems on CNNDM (Hermann et al., 2015) and XSUM (Narayan et al., 2018), providing fine-grained annotations of sentence-level factuality error types. On the other hand, REALSumm (Bhandari et al., 2020) is another dataset of 2,500 summaries from 25 summarization systems for automated metrics based on CNNDM. It includes a list of human keyfacts, along with corresponding annotations indicating their presence in the summary. FRANK and REALSumm obtain the inter-annotator agreement (IAA) scores of 0.58 (cohen's kappa) and 0.66 (Krippendorff's alpha) for three annotators, respectively.

LLMs as Evaluators We use the GPT-4-turbo (gpt-4-1106-preview) (Achiam et al., 2023) by default in main evaluation, but test with various open-source and proprietary LLMs, including Mixtral-8x7B (Jiang et al., 2024), Phi-2, Llama-2/-3 (Touvron et al., 2023), GPT-3.5-turbo, and GPT-4-omni (gpt-4o-2024-05-13), in Section 4.2. We set the temperature to 0 and clear the history for every evaluation instance, following the literature (Shen et al., 2023). We use HuggingFace models for open-source LLMs and paid APIs for proprietary LLMs.

Baselines We compare FineSurE with five similarity-based methods, ROUGE-1/-2/-L (Lin, 2004), BERTScore (Zhang et al., 2019), and BARTScore (Yuan et al., 2021); a NLI-based method, SummaC-Conv (Laban et al., 2022); two QA-based methods, UniEval (Zhong et al., 2022) and QAFactEval (Fabbri et al., 2022); and the latest LLM-based method, G-Eval (Liu et al., 2023). Note that QAFactEval and SummaC-Conv are only compared for faithfulness evaluation, as they are limited to factuality. We obtain all the results by executing each metric in our experimental setup.

Performance Each automated evaluator's performance is assessed by comparing estimated scores with ground-truth human judgments using *sentence*, *summary*, and *system*-level measurements. This multi-level analysis is crucial, as we seek to understand the agreement of the evaluator on each sentence, each summary, and the average performance of each summarization system.

Balanced accuracy (bAcc) assesses faithfulness in classifying each summary sentence for the presence or absence of factual errors at the sentence-level. This is the average of true positive and true negative rates widely used when the two classes are imbalanced (Brodersen et al., 2010). *Pearson* and *Spearman correlations* assess all three dimensions at the summary-level by comparing percentage scores in Eqs. (1)-(2) derived from predicted and human evaluation results. Lastly, *rank correlation* is a system-level measure assessing the alignment of performance rankings across summarization systems (models) calculated by both our evaluator and humans. The details of the measurements are provided in Appendix D.

Error Category	OutE	EntE	PredE	CirE	GramE	LinkE	CorefE	Mean
Random Guessing	14.3%	14.3%	14.3%	14.3%	14.3%	14.3%	14.3%	14.3%
Bart-Large (Fine-tuned)	**56.7**%	36.9%	14.8%	34.0%	21.4%	0.0%	**40.0**%	29.1%
FineSurE (GPT-4)	50.2%	**63.7**%	**41.9**%	**38.1**%	**44.6**%	**19.4**%	37.8%	**42.2**%

Table 2: Accuracy analysis of **factuality error localization** in assessing faithfulness, with error categories including OutE (out-of-context), EntE (entity error), PredE (predicate error), CirE (circumstance error), GramE (grammatical error), LinkE (discourse link error), and CorefE (coreference error). "Random Guessing" is the performance of randomly selecting from the seven categories, i.e., 1/7=14.3%, while "Bart-Large" is a stronger baseline model fine-tuned on FRANK for error localization.

Dimension		(a) Completeness			(b) Conciseness		
Direction	Method	Summary-level Pearson (↑)	Summary-level Spearman (↑)	System-level Rank (↑)	Summary-level Pearson (↑)	Summary-level Spearman (↑)	System-level Rank (↑)
Similarity-based	ROUGE-1	0.484 (0.00)	0.461 (0.00)	0.516 (0.01)	0.387 (0.00)	0.371 (0.00)	0.332 (0.10)
	ROUGE-2	0.456 (0.00)	0.461 (0.00)	0.463 (0.02)	0.328 (0.00)	0.337 (0.00)	0.290 (0.16)
	ROUGE-L	0.425 (0.00)	0.428 (0.00)	0.238 (0.25)	0.310 (0.00)	0.321 (0.00)	0.083 (0.69)
	BERTScore	0.455 (0.00)	0.443 (0.00)	0.619 (0.00)	0.416 (0.00)	0.405 (0.00)	0.783 (0.00)
	BARTScore	0.216 (0.00)	0.199 (0.00)	0.653 (0.00)	0.241 (0.00)	0.210 (0.00)	0.824 (0.00)
QA-based	UniEval	0.134 (0.00)	0.180 (0.00)	0.346 (0.09)	0.086 (0.00)	0.128 (0.00)	-0.176 (0.39)
LLM-based	G-Eval (GPT4)	0.314 (0.00)	0.295 (0.00)	0.908 (0.00)	0.314 (0.00)	0.277 (0.00)	0.582 (0.00)
	FineSurE (GPT-4)	**0.688** (0.00)	**0.677** (0.00)	**0.949** (0.00)	**0.505** (0.00)	**0.451** (0.00)	0.880 (0.00)
	FineSurE† (GPT-4)	0.571 (0.00)	0.546 (0.00)	0.905 (0.00)	0.438 (0.00)	0.399 (0.00)	**0.911** (0.00)

Table 3: Performance of **completeness and conciseness evaluation on REALSumm** using ten automated evaluation metrics at the summary- and system-level. The values in parenthesis represent p-values. Fine-Eval† utilizes the list of keyfacts automatically derived through LLMs, in contrast to relying on human keyfacts.

Method	Faithfulness	Completeness	Conciseness
G-Eval	0.906	0.799	0.759
FineSurE	**0.921**	**0.853**	**0.908**

Table 4: **Inter-annotator agreement score (IAA)** of GEval and FineSurE across three distinct evaluations.

4.1 Main Results: Evaluators Comparison

4.1.1 Faithfulness

Table 1 summarizes the agreement between automated evaluators and human scores in faithfulness evaluation at three different granularities. FineSurE significantly outperforms similarity-, NLI-, and QA-based evaluators at all levels of evaluation.

It is important to note that none of the existing methods provide sentence-level evaluation results, relying instead on summary-level scoring, such as Likert-scale scores. It is noteworthy that FineSurE has the capability to assess whether each sentence contains a factual error or not, demonstrating remarkable alignment with human sentence-level judgments, with a balanced accuracy of 86.4%.

Given the strong alignment with human judgment, using LLMs as an evaluator holds great promise for enhancing the reliability of evaluation processes for text summarization. However, one open question remains: *Can LLMs identify the type of factuality error?*

Table 2 unveils the capability of LLMs for factuality error localization, demonstrating accuracy as the probability that the predicted error category matches the correct answer given by humans. FineSurE outperforms the strong baseline, Bart-Large[3] fine-tuned on FRANK for error localization, despite not being trained on any error localization data, i.e., zero-shot prediction. Its superiority is primarily stemming from error categories that are uncommon in the training set for Bart-Large, such as PredE (141 cases), CirE (142 cases), and LinkE (41 cases). Nevertheless, LLMs still make numerous mistakes in accurately identifying the exact error type, despite their excellent performance in the binary decision of hallucination.

Therefore, achieving a level of evaluation comparable to human performance in more intricate assessment tasks remains a challenging objective.

4.1.2 Completeness and Conciseness

The agreement between automated evaluators and human scores on completeness and conciseness is summarized in Table 3. In contrast to similarity-based evaluators, which provide a single composite score, UniEval and G-Eval yield four distinct

[3]The pre-trained Bart-Large (Lewis et al., 2020) is fine-tuned on error localization data constructed using FRANK, comprising 3,885 training and 1,007 testing sentences, each paired with their corresponding human error categories.

Type	LLM	Sentence-level bAcc (↑)	Summary-level Pearson Corr (↑)	Summary-level Spearman Corr (↑)	System-level Rank Corr (↑)	Success Ratio
Open-source	Phi-2 (2.7B)	48.1%	-0.108 (0.00)	-0.010 (0.73)	-0.700 (0.04)	50.4%
	Llama2-70B	56.5%	0.133 (0.00)	0.147 (0.00)	0.833 (0.01)	86.2%
	Mixtral-8x7B	50.7%	-0.023 (0.38)	0.036 (0.18)	-0.450 (0.22)	63.1%
	Mixtral-8x7B-Inst.	78.7%	0.708 (0.00)	0.716 (0.00)	0.883 (0.00)	88.9%
	Llama3-70B-Inst.	**92.0**%	0.844 (0.00)	0.841 (0.00)	0.933 (0.01)	**98.3**%
Proprietary	Gemini-1-pro	87.7%	0.733 (0.00)	0.736 (0.00)	0.916 (0.00)	98.0%
	GPT-3.5-turbo	78.8%	0.709 (0.00)	0.709 (0.00)	0.933 (0.00)	93.1%
	GPT-4-turbo	86.4%	0.833 (0.00)	0.839 (0.00)	**0.950** (0.00)	98.1%
	GPT-4-omni	91.8%	**0.855** (0.00)	**0.852** (0.00)	0.883 (0.00)	98.1%

Table 5: Performance of **faithfulness evaluation** using five open-source and four proprietary LLMs. The rightmost column is the success ratio of accurately following the prompt.

scores, evaluating faithfulness, coherence, relevance, and fluency. We use their coherence and relevance scores to calculate the correlation with human scores for completeness and conciseness, as they indicate the inclusion and density of key information, respectively.

Overall, FineSurE using human keyfacts demonstrates a very high agreement with human evaluations for completeness and conciseness, surpassing other evaluators significantly. This is because keyfact alignment is essential to verify the coverage of crucial information in the summary, a task that cannot be accomplished with existing LLM-based method like G-Eval. See the qualitative example in Appendix E. We also assess the performance of FineSurE without employing human keyfacts and, instead, utilizing machine-generated keyfacts, as outlined in Appendix C. The keyfacts are extracted using GPT-4 with a specific prompt. It is noteworthy that, even with machine-generated key facts, FineSurE maintains a higher level of agreement over other automated evaluators.

With an advantage as a fine-grained evaluator, FineSurE also provides evaluation results at the keyfact-level, revealing which keyfacts are omitted in the summary, i.e., keyfact matching. Given a list of keyfacts, it includes binary labels ("Yes" or "No") in the JSON output, as illustrated in Figure 4. Therefore, we assess the agreement for the keyfact matching task by calculating the IAA score between machine and human labels. FineSurE demonstrates a Krippendorff's alpha of 0.65 for keyfact matching. This robust agreement at various levels corroborates that FineSurE has a potential to be an effective fine-grained automatic evaluator.

Furthermore, in Appendix F, we compare FineSurE with two variants of G-Eval, which are tailored for completeness and conciseness evaluation by modifying its prompts to be more suitable for such assessment and integrating them for use with keyfacts. FineSurE maintains its significant dominance even with additional tuning on G-Eval.

4.1.3 Stability in Evaluation Results

Concerns arise about evaluation result stability with LLMs due to their inherent text generation randomness, even at temperature 0. Despite LLM-based methods relying on Likert-scale evaluation, such as G-Eval, showing significant fluctuations in judgment alignment (Shen et al., 2023; Liu et al., 2023), Table 4 demonstrates that FineSurE (GPT-4) maintains much higher agreement in summary-level evaluation scores across three distinct runs. This underscores the benefit of employing fine-grained percentage scores derived from sentence- and keyfact-level assessments.

4.2 LLMs as Evaluators Comparison

It is interesting to observe how the evaluation agreement varies based on the choice of LLMs, given the abundance of open-source and proprietary LLMs.

Success Ratio. The primary limitation of open-source LLMs is their comparatively lower success ratio in following prompts, compared to proprietary LLMs; only Llama3-70B-Inst exhibits a high success ratio comparable to proprietary LLMs. Upon analyzing failure cases, the top three reasons are: (1) the output is either not in JSON format or an incorrect JSON format, (2) the output consists of meaningless text, e.g., python codes or no output at all, and (3) the JSON output includes only a few lines of sentences or keyfacts.

Furthermore, the maximum token length in context for open-source LLMs is notably shorter compared to proprietary LLMs. GPT-4 series can process up to 128K tokens, whereas open-source LLMs generally handle up to 8K input tokens. This results in prompt truncation when handling lengthy

Dimension		(a) Completeness			(b) Conciseness			
Type	Method	Summary-level Pearson (↑)	Summary-level Spearman (↑)	System-level Rank (↑)	Summary-level Pearson (↑)	Summary-level Spearman (↑)	System-level Rank (↑)	Succ. Ratio
Open-source	Phi-2 (2.7B)	0.093 (0.00)	0.104 (0.00)	0.338 (0.10)	0.058 (0.04)	0.069 (0.01)	-0.039 (0.85)	52.1%
	Llama2-70B	0.421 (0.00)	0.401 (0.00)	0.824 (0.00)	0.387 (0.00)	0.371 (0.00)	0.612 (0.00)	53.7%
	Mixtral-8x7B	0.166 (0.00)	0.152 (0.00)	0.431 (0.03)	0.087 (0.00)	0.108 (0.00)	0.264 (0.20)	53.8%
	Mixtral-8x7B-Inst.	0.439 (0.00)	0.437 (0.00)	0.678 (0.00)	0.367 (0.00)	0.361 (0.00)	0.798 (0.00)	87.5%
	Llama3-70B-Inst.	**0.755** (0.00)	**0.747** (0.00)	0.881 (0.00)	0.445 (0.00)	0.444 (0.00)	0.786 (0.00)	92.0%
Proprietary	Gemini-1-pro	0.583 (0.00)	0.567 (0.00)	0.820 (0.00)	0.435 (0.00)	0.402 (0.00)	0.745 (0.00)	99.7%
	GPT-3.5-turbo	0.509 (0.00)	0.493 (0.00)	0.848 (0.00)	0.381 (0.00)	0.372 (0.00)	0.706 (0.00)	74.5%
	GPT-4-turbo	0.688 (0.00)	0.677 (0.00)	0.949 (0.00)	0.505 (0.00)	0.451 (0.00)	0.880 (0.00)	**99.8%**
	GPT-4-omni	0.691 (0.00)	0.686 (0.00)	**0.943** (0.00)	**0.522** (0.00)	**0.467** (0.00)	**0.932** (0.00)	99.6%

Table 6: Performance of **completeness and conciseness** evaluation using five open-source and four proprietary LLMs. The rightmost column is the success ratio of accurately following the prompt.

Type	LLM Models	Factuality Error Type OutE	EntE	PredE	CirE	GramE	LinkE	CorefE	Mean
Baseline (Random Guessing)		14.3%	14.3%	14.3%	14.3%	14.3%	14.3%	14.3%	14.3%
Open-source	Phi-2	8.1%	12.6%	4.5%	5.8%	6.9%	8.3%	6.8%	7.6%
	Llama2-70B	21.6%	31.4%	13.4%	14.3%	19.0%	11.1%	25.0%	19.4%
	Mixtral-8x7b	21.8%	24.1%	16.0%	6.7%	15.6%	7.1%	5.6%	13.8%
	Mixtral-8x7b-Inst.	37.8%	45.4%	26.1%	12.5%	22.2%	25.0%	22.7%	27.4%
	Llama3-70B-Inst.	66.1%	64.8%	41.1%	38.1%	**54.5%**	**43.8%**	37.5%	49.4%
Proprietary	Gemini-1-pro	51.9%	36.0%	23.2%	18.2%	3.8%	0.0%	25.0%	22.6%
	GPT-3.5-turbo	52.4%	42.4%	26.4%	25.9%	52.4%	0.0%	12.9%	30.3%
	GPT-4-turbo	50.2%	63.7%	41.9%	38.1%	44.6%	19.4%	37.8%	42.2%
	GPT-4-omni	**70.6%**	**69.2%**	**45.7%**	**46.6%**	50.0%	42.3%	**44.0%**	**52.6%**

Table 7: Accuracy analysis of **factuality error localization** in assessing faithfulness using five open-source and four proprietary LLMs, where "Baseline" is the performance of random guessing. Top-1 values are marked in bold.

input texts, potentially leading to failures in generating accurate outputs in text summarization.

Agreement with Human Score. Tables 5-6 summarize the correlation of nine different LLMs with human judgment, computed only for the successful cases of adhering to the prompt. Although the recent Llama3-70B-Inst shows strong agreement with humans, in general, there is a noticeable gap between open-source and proprietary LLMs. Regarding open-source LLMs, the agreement with human scores increases with the model size; for example, Llama2-70B exhibits a higher correlation coefficient than Phi-2. Additionally, instruction tuning also plays a role, as observed in Mixtral-8x7b's performance, which improved significantly after instruction tuning. In contrast, all the proprietary LLMs exhibit high correlation coefficient. Particularly, more recent and powerful LLMs exhibit better performance, i.e., GPT-4-turbo > GPT-3.5-turbo, GPT-4-omni > GPT-4-turbo.

It's notable that LLMs with a high success ratio exhibit a strong correlation, suggesting they are not penalized by their high success ratios. Therefore, a more advanced LLM simultaneously achieves higher agreement and success ratios.

Error Localization We provide a detailed factuality error localization analysis using different LLMs in Table 7. GPT-4-omni improves the mean accuracy in error localization by 10% over GPT-4-turbo. The categorization accuracies of open-source LLMs are considerably lower than those of proprietary LLMs in general. However, the latest open-source LLM, Llama3-70B-Inst, outperforms GPT-4-turbo in error localization, achieving an average prediction accuracy of 49.4%, which is 7.2% higher than that of GPT-4-turbo. Additionally, instruction tuning demonstrates a significant accuracy boost in this task, as evidenced by the improvement from Mixtral-8x7b to Mixtral-8x7b-Inst.

4.3 Evaluation using FineSurE

As an actual application of an automated evaluator, we gather summaries generated by four open-source and four proprietary LLMs, and subsequently assess their summarization quality using the FineSurE algorithm (see the prompt for summarization we used in Appendix H). Figure 2 shows the percentage scores of the eight LLMs for faithfulness, completeness, and conciseness. The summaries are generated for 100 examples sourced from CNNDM, all of which are also included in

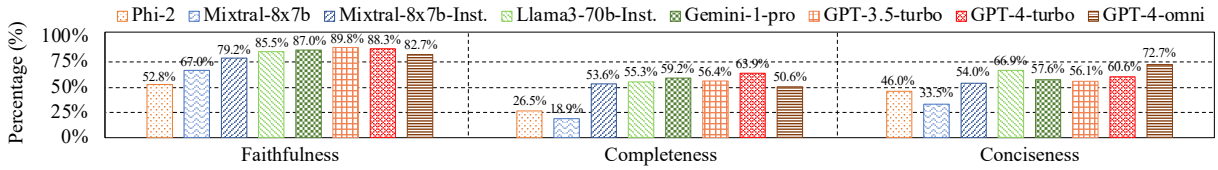

Figure 2: **Evaluation using FineSurE** for eight LLMs in text summarization on CNNDM.

REALSumm, thereby possessing the list of keyfacts extracted by human annotators.

In general, proprietary LLMs, including different versions of GPT, generate high-quality summaries in comparison to open-source counterparts. Interestingly, GPT-4-omni exhibits the highest agreement with humans as an automated evaluator in Tables 5-6, but its faithfulness and completeness scores are significantly worse even than GPT-3.5-turbo. Consequently, GPT-4-omni is likely to include more hallucinations and miss many important keyfacts in summary generation.

The performance ranking of each model changes significantly for each evaluation dimension. GPT-3.5-turbo, GPT-4-turbo, and GPT-4-omni are the best for faithfulness, completeness, and conciseness, respectively. Nevertheless, it is noteworthy that Llama3-70B-Inst, an open-source LLM, exhibits comparable performance to the state-of-the-art proprietary LLMs. In open-source LLMs, instruction tuning significantly enhances summarization quality, as evidenced by the performance increase of Mixtral-8x7b-Inst over Mixtral-8x7b. These findings align with prior observations reported in recent studies on faithfulness (Laban et al., 2023) and instruction tuning (Zhang et al., 2023).

Lastly, while there is no doubt that faithfulness is crucial, achieving both completeness and conciseness simultaneously turns out to be very important and challenging in text summarization, as evident from the low percentage scores even with GPT-4 series. Therefore, it emphasizes the need to put more effort into these aspects for a good summary.

5 Conclusion

We introduce FineSurE, a novel automated evaluator designed for fine-grained and multi-dimensional text summarization evaluation. The evaluation process is broken down into fact checking and keyfact alignment, providing detailed insights, where keyfacts can be either provided by humans or generated by LLMs. Our experiments include a thorough comparison with existing evaluators, exploration of performance using eight opensource or proprietary LLMs, and real quality assessment of recent LLM-generated summaries. The results indicate the potential effectiveness of FineSurE as a text summarization evaluator, showcasing its promising capabilities in advancing automated evaluation for text summarization.

Limitations

Our automated evaluator is primarily tested on the news domain due to the limited availability of benchmark datasets with fine-grained human annotations. We emphasize the critical importance of constructing a high-quality benchmark dataset with high diversity in input domains, length, and types. Also, the prompts for evaluation may need to be tuned if a different summary is expected like the summary from the medical domain. Lastly, other aspects can be considered for text summarization, such as toxicity and social bias. We leave these challenges as future work.

Ethics Statement

This paper focuses on designing an automatic evaluator using LLMs for text summarization. Therefore, we do not anticipate any negative ethical and social impact.

Acknowledgements

The first author, Hwanjun Song, was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. RS-2024-00334343) and Artificial Intelligence industrial convergence cluster development project funded by the Ministry of Science and ICT (MSIT, Korea) & Gwangju Metropolitan City (No. BA00000772).

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for mt evaluation with improved correlation with human judgments. In *ACLW*.

Manik Bhandari, Pranav Narayan Gour, Atabak Ashfaq, Pengfei Liu, and Graham Neubig. 2020. Re-evaluating evaluation in text summarization. In *EMNLP*.

Kay Henning Brodersen, Cheng Soon Ong, Klaas Enno Stephan, and Joachim M Buhmann. 2010. The balanced accuracy and its posterior distribution. In *ICPR*.

Yi Chen, Rui Wang, Haiyun Jiang, Shuming Shi, and Ruifeng Xu. 2023. Exploring the use of large language models for reference-free text quality evaluation: A preliminary empirical study. *arXiv preprint arXiv:2304.00723*.

Yiran Chen, Pengfei Liu, and Xipeng Qiu. 2021. Are factuality checkers reliable? adversarial meta-evaluation of factuality in summarization. In *EMNLP*.

Alexander R Fabbri, Wojciech Kryściński, Bryan McCann, Caiming Xiong, Richard Socher, and Dragomir Radev. 2021. Summeval: Re-evaluating summarization evaluation. *Transactions of the Association for Computational Linguistics*, 9:391–409.

Alexander Richard Fabbri, Chien-Sheng Wu, Wenhao Liu, and Caiming Xiong. 2022. QAFactEval: Improved qa-based factual consistency evaluation for summarization. In *NAACL*.

Shangbin Feng, Vidhisha Balachandran, Yuyang Bai, and Yulia Tsvetkov. 2023. FactKB: Generalizable factuality evaluation using language models enhanced with factual knowledge. In *EMNLP*.

Jinlan Fu, See-Kiong Ng, Zhengbao Jiang, and Pengfei Liu. 2023. GPTscore: Evaluate as you desire. *arXiv preprint arXiv:2302.04166*.

Xiyan Fu and Anette Frank. 2023. SETI: Systematicity evaluation of textual inference. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 4101–4114, Toronto, Canada. Association for Computational Linguistics.

Mingqi Gao, Jie Ruan, Renliang Sun, Xunjian Yin, Shiping Yang, and Xiaojun Wan. 2023. Human-like summarization evaluation with chatgpt. *arXiv preprint arXiv:2304.02554*.

Mingqi Gao and Xiaojun Wan. 2022. Dialsummeval: Revisiting summarization evaluation for dialogues. In *NAACL*.

John Glover, Federico Fancellu, Vasudevan Jagannathan, Matthew R Gormley, and Thomas Schaaf. 2022. Revisiting text decomposition methods for nli-based factuality scoring of summaries. In *GEM*.

Tanya Goyal and Greg Durrett. 2020. Evaluating factuality in generation with dependency-level entailment. In *EMNLP*.

Som Gupta and Sanjai Kumar Gupta. 2019. Abstractive summarization: An overview of the state of the art. *Expert Systems with Applications*, 121:49–65.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *NeurIPS*.

Or Honovich, Roee Aharoni, Jonathan Herzig, Hagai Taitelbaum, Doron Kukliansy, Vered Cohen, Thomas Scialom, Idan Szpektor, Avinatan Hassidim, and Yossi Matias. 2022. TRUE: Re-evaluating factual consistency evaluation. In *NAACL*.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38.

Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. 2024. Mixtral of experts. *arXiv preprint arXiv:2401.04088*.

Wojciech Kryściński, Bryan McCann, Caiming Xiong, and Richard Socher. 2020. Evaluating the factual consistency of abstractive text summarization. In *EMNLP*.

Philippe Laban, Wojciech Kryściński, Divyansh Agarwal, Alexander Richard Fabbri, Caiming Xiong, Shafiq Joty, and Chien-Sheng Wu. 2023. SummEdits: Measuring llm ability at factual reasoning through the lens of summarization. In *EMNLP*.

Philippe Laban, Tobias Schnabel, Paul N Bennett, and Marti A Hearst. 2022. SummaC: Re-visiting nli-based models for inconsistency detection in summarization. *Transactions of the Association for Computational Linguistics*, 10:163–177.

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

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*.

Yen-Ting Lin and Yun-Nung Chen. 2023. Llm-eval: Unified multi-dimensional automatic evaluation for open-domain conversations with large language models. *arXiv preprint arXiv:2305.13711*.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-Eval: NLG evaluation using gpt-4 with better human alignment. *arXiv preprint arXiv:2303.16634*.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On faithfulness and factuality in abstractive summarization. In *ACL*.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. FActScore: Fine-grained atomic evaluation of factual precision in long form text generation. In *EMNLP*.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. *ArXiv*, abs/1808.08745.

Artidoro Pagnoni, Vidhisha Balachandran, and Yulia Tsvetkov. 2021. Understanding factuality in abstractive summarization with frank: A benchmark for factuality metrics. In *NAACL*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A method for automatic evaluation of machine translation. In *ACL*.

Keita Saito, Akifumi Wachi, Koki Wataoka, and Youhei Akimoto. 2023. Verbosity bias in preference labeling by large language models. *arXiv preprint arXiv:2310.10076*.

Thomas Scialom, Paul-Alexis Dray, Patrick Gallinari, Sylvain Lamprier, Benjamin Piwowarski, Jacopo Staiano, and Alex Wang. 2021. QuestEval: Summarization asks for fact-based evaluation. In *EMNLP*.

Chenhui Shen, Liying Cheng, Xuan-Phi Nguyen, Yang You, and Lidong Bing. 2023. Large language models are not yet human-level evaluators for abstractive summarization. In *EMNLP*.

Yuchen Shen and Xiaojun Wan. 2023. Opinsummeval: Revisiting automated evaluation for opinion summarization. *arXiv preprint arXiv:2310.18122*.

Xiaoming Shi, Jie Xu, Jinru Ding, Jiali Pang, Sichen Liu, Shuqing Luo, Xingwei Peng, Lu Lu, Haihong Yang, Mingtao Hu, et al. 2023. Llm-mini-cex: Automatic evaluation of large language model for diagnostic conversation. *arXiv preprint arXiv:2308.07635*.

Hwanjun Song, Igor Shalyminov, Hang Su, Singh Siffi, Kaisheng Yao, and Saab Mansour. 2023. Enhancing abstractiveness of summarization models through calibrated distillation. In *EMNLP*.

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

PA Utama, J Bambrick, NS Moosavi, and I Gurevych. 2022. Falsesum: generating document-level nli examples for recognizing factual inconsistency in summarization. In *NAACL*.

Alex Wang, Kyunghyun Cho, and Mike Lewis. 2020. Asking and answering questions to evaluate the factual consistency of summaries. In *ACL*.

Jiaan Wang, Yunlong Liang, Fandong Meng, Haoxiang Shi, Zhixu Li, Jinan Xu, Jianfeng Qu, and Jie Zhou. 2023. Is chatgpt a good nlg evaluator? a preliminary study. *arXiv preprint arXiv:2303.04048*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *NeurIPS*.

Zihan Yu, Liang He, Zhen Wu, Xinyu Dai, and Jiajun Chen. 2023. Towards better chain-of-thought prompting strategies: A survey. *arXiv preprint arXiv:2310.04959*.

Weizhe Yuan, Graham Neubig, and Pengfei Liu. 2021. BARTScore: Evaluating generated text as text generation. In *NeurIPS*.

Shengyu Zhang, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, Tianwei Zhang, Fei Wu, et al. 2023. Instruction tuning for large language models: A survey. *arXiv preprint arXiv:2308.10792*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. BERTScore: Evaluating text generation with bert. In *ICLR*.

Wei Zhao, Maxime Peyrard, Fei Liu, Yang Gao, Christian M Meyer, and Steffen Eger. 2019. MoverScore: Text generation evaluating with contextualized embeddings and earth mover distance. In *EMNLP*.

Ming Zhong, Yang Liu, Da Yin, Yuning Mao, Yizhu Jiao, Pengfei Liu, Chenguang Zhu, Heng Ji, and Jiawei Han. 2022. Towards a unified multi-dimensional evaluator for text generation. In *EMNLP*.

```
You will receive a transcript followed by a corresponding summary.
Your task is to assess the factuality of each summary sentence
across nine categories:
* no error: the statement aligns explicitly with the content of
the transcript and is factually consistent with it.
* out-of-context error: the statement contains information not
present in the transcript.
* entity error: the primary arguments (or their attributes) of
the predicate are wrong.
* predicate error: the predicate in the summary statement is
inconsistent with the transcript.
* circumstantial error: the additional information (like location
or time) specifying the circumstance around a predicate is wrong.
* grammatical error: the grammar of the sentence is so wrong that
it becomes meaningless.
* coreference error: a pronoun or reference with wrong or non-
existing antecedent.
* linking error: error in how multiple statements are linked
together in the discourse (for example temporal ordering or
causal link).
* other error: the statement contains any factuality error which
is not defined here.

Instruction:
First, compare each summary sentence with the transcript.
Second, provide a single sentence explaining which factuality
error the sentence has.
Third, answer the classified error category for each sentence in
the summary.

Provide your answer in JSON format. The answer should be a list
of dictionaries whose keys are "sentence", "reason", and
"category":
[{"sentence": "first sentence", "reason": "your reason",
"category": "no error"}, {"sentence": "second sentence", "reason":
"your reason", "category": "out-of-context error"}, {"sentence":
"third sentence", "reason": "your reason", "category": "entity
error"},]

Transcript:
{input text}

Summary with N sentences:
{summary sentence 1}
{summary sentence 2}
        ...
{summary sentence N}
```

Figure 3: **Prompt for fact checking:** the prompt is tailored for a categorization task, utilizing an instruction format with a structured reasoning step. For every summary sentence, the output is a dictionary that provides the category (one of the error types) along with a concise sentence of reasoning.

A Factuality Error Type

Following the work (Fu and Frank, 2023), we use seven categories to define factuality error types, namely "out of context", "predicate," "entity," "circumstance," "coreference," "discourse link," and "grammatical". Table 8 provides the detailed description and example of the error categories.

B Main Prompt

We employ LLMs as a tool to conduct fact checking and keyfact alignment tasks. Specifically, we design two prompts tailored for the two tasks, as shown in Figures 3-4.

C Keyfact Extraction

The list of key facts is crucial for evaluating completeness and conciseness. Ideally, they should be generated by humans, as the key facts in text summarization heavily depend on what information humans prioritize in various domains. For instance, in a medical scenario, keyfacts should encompass all medical symptoms and the doctor's recommended

```
You will receive a summary and a set of key facts for the same
transcript. Your task is to assess if each key fact is inferred
from the summary.

Instruction:
First, compare each key fact with the summary.
Second, check if the key fact is inferred from the summary and
then response "Yes" or "No" for each key fact. If "Yes", specify
the line number(s) of the summary sentence(s) relevant to each
key fact.

Provide your answer in JSON format. The answer should be a list
of dictionaries whose keys are "key fact", "response", and "line
number":
[{"key fact": "first key fact", "response": "Yes", "line number":
[1]}, {"key fact": "second key fact", "response": "No", "line
number": []}, {"key fact": "third key fact", "response": "Yes",
"line number": [1, 2, 3]}]

Summary:
[1] {summary sentence 1}
[2] {summary sentence 2}
          ...
[N] {summary sentence N}

M key facts:
{keyfact 1}
{keyfact 2}
    ...
{keyfact M}
```

Figure 4: **Prompt for keyfact alignment:** the prompt is tailored for keyfact matching, employing a simple instruction format. For every keyfact, the output is a dictionary that provides a binary labeling of "Yes" (align) or "No" (not align) and all aligned sentence IDs.

```
You will be provided with a summary. Your task is to decompose
the summary into a set of "key facts". A "key fact" is a single
fact written as briefly and clearly as possible, encompassing at
most 2-3 entities.

Here are nine examples of key facts to illustrate the desired
level of granularity:
* Kevin Carr set off on his journey from Haytor.
* Kevin Carr set off on his journey from Dartmoor.
* Kevin Carr set off on his journey in July 2013.
* Kevin Carr is less than 24 hours away from completing his trip.
* Kevin Carr ran around the world unsupported.
* Kevin Carr ran with his tent.
* Kevin Carr is set to break the previous record.
* Kevin Carr is set to break the record by 24 hours.
* The previous record was held by an Australian.

Instruction:
First, read the summary carefully.
Second, decompose the summary into (at most 16) key facts.

Provide your answer in JSON format. The answer should be a
dictionary with the key "key facts" containing the key facts as a
list:
{"key facts" ["first key fact", "second key facts", "third key
facts"]

Summary:
{summary}
```

Figure 5: **Prompt for keyfact extraction:** The prompt is tailored for extracting keyfacts, utilizing an instruction format with few-shot examples. Given a reference summary, the output is a dictionary that provides a list of keyfacts. We adhere to the REALSum's annotation guideline, which limits the number of key facts to 16.

treatment, while in a sales call, the customer's issue and action should be prioritized as the key facts.

We also automatically extract feasible keyfacts from the reference summary, similar to the approach taken by REALSumm in annotating key facts with human annotators. However, our process is fully automatic, utilizing LLMs instead of human's manual efforts. We designed the prompt tailored for extracting keyfacts in Figure 5. This prompt generates up to 16 key facts from the reference summaries by providing few-shot examples

	Category	Description	Example
OutE	Out of context Error	The statement contains information not present in the source article.	*China has already started clinical trials of the COVID-19 vaccine.*
EntE	Entity Error	The primary arguments (or their attributes) of the predicate are wrong.	*The COVID-19 vaccine was approved by the FDA in 2019.*
PredE	Predicate Error	The predicate in the summary statement is inconsistent with the source article.	*The Ebola vaccine was rejected by the FDA in 2019.*
CirE	Circumstance Error	The additional information (like location or time) specifying the circumstance around a predicate is wrong.	*The first vaccine for Ebola was approved by the FDA in 2014.*
GramE	Grammatical Error	The grammar of the sentence is so wrong that it becomes meaningless.	*The Ebola vaccine accepted have already started.*
LinkE	Discourse Link Error	Error in how multiple statements are linked together in the discourse (for example tempral ordering/causal link).	*To produce the vaccine, scientists have to show successful human trials, then sequence the DNA of the virus.*
CorefE	Coreference Error	A pronoun/reference with wrong or non-existing antecedent.	*The first vaccine for Ebola was approved in 2019. They say a vaccine for COVID-19 is unlikely to be ready this year.*

Table 8: **Typology of factual errors copied from (Pagnoni et al., 2021).** Original text for the examples: *The first vaccine for Ebola was approved by the FDA in 2019 in the US, five years after the initial outbreak in 2014. To produce the vaccine, scientists had to sequence the DNA of Ebola, then identify possible vaccines, and finally show successful clinical trials. Scientists say a vaccine for COVID-19 is unlikely to be ready this year, although clinical trials have already started.*

to ensure that the granularity of extracted key facts aligns with our requirements.

D Measurement

We utilize several measurement to compute the agreement with human judgements at the three different levels.

For sentence-level assessment, we utilize human binary annotations indicating the presence of factuality errors for each sentence, denoted as "0" for no error and "1" for error. Similarly, the LLM returns the binary decision of "0" (No) and "1" (Yes) by the fact checking prompt per sentence, as shown in Figure 3. Then, the *balanced accuracy (bACC)* is computed by:

$$\text{bACC} = (\text{sensitivity} + \text{specificity})/2, \quad (3)$$

where sensitivity refers to the true positive rate, which measures the proportion of correct predictions by LLMs out of all positive predictions. On the other hand, specificity is the true negative rate, measuring the proportion of correct predictions out of all negative predictions.

For the summary-level assessment, let $D = \{d_1, \ldots, d_k\}$ be the set of input documents and $S = \{s_1, \ldots, s_k\}$ be the set of summaries corresponding to the document set. Supposing that F_{gt} and F_{pred} are the functions that returns the percentage score of a specific evaluation dimensions based on human and predicted labels, respectively (F can be any scoring function in Eq. (1)-(2)). Then, the *summary-level* correlation is calculated as follows:

$$\text{Corr}\Big(\big[F_{gt}(d_1, s_1), \ldots, F_{gt}(d_k, s_k)\big], \big[F_{pred}(d_1, s_1), \ldots, F_{pred}(d_k, s_k)\big]\Big), \quad (4)$$

where Corr is one of the Pearson and Spearman correlation measure. The measurement reveals the agreement between the percentage scores generated by automated evaluators and human judgments.

For system-level assessment, we consolidate the percentage scores across all input documents , determining the average percentage score for each summarization model. Let $\mathbb{F}_m = \{F_m(d_1, s_1), \ldots, F_m(d_k, s_k)\}$ represents the percentage scores derived from the labels assigned by a summarization system m. Then, we make a list of the average percentage score for all summarization systems, $[\bar{\mathbb{F}}_{m_1}, \bar{\mathbb{F}}_{m_2}, \ldots]$ and compute their ranking by using the Rank function, returning the list $[\text{rank}_{m_1}, \text{rank}_{m_2}, \ldots]$, where rank_m is the ranking of the model m. Given a list of ground-truth rankings $[\text{rank}^*_{m_1}, rank^*_{m_2}, \ldots]$ using the human scores, we compute the *rank correlation* by:

$$\text{Spearman}\Big(\big[\text{rank}_{m_1}, \text{rank}_{m_2}, \ldots\big], \big[\text{rank}^*_{m_1}, rank^*_{m_2}, \ldots\big]\Big). \quad (5)$$

Model Summary	Human Keyfact List	Alignment by **Human**	Assignment by **FineSurE**	**G-Eval**
Zbigniew Huminski , 38 , has confessed to strangling [10] his nine - year - old victim . She was stripped naked and sexually assaulted and forced into Huminski's car [2] . The Little Girl's naked body was found [3] in the woods near Calais 90 minutes [4, 5] after she was taken . Chloe's mother , named only as Isabelle , heard her screams as she was being taken away from her school in Calais . DNA evidence corroborated by an autopsy [6] revealed strangulation and sexual violence [7] . He was on his way to Britain[9] from Calais when he snatched a schoolgirl .	[1] Chloe's was playing with friend.	X	X	N/A
	[2] Zbigniew Huminski forced Chloe into the car.	O	O	
	[3] Chloe's naked body was found.	O	O	
	[4] Chloe's body was found in nearby woods.	O	O	
	[5] Chloe's body was found an hour - and - A - half later.	O	O	
	[6] Prosecutors say there is evidence.	O	X	
	[7] The evidence was of ' strangulation and sexual violence.	O	O	
	[8] Zbigniew Huminski is a Polish immigrant.	X	X	
	[9] Zbigniew Huminski was heading to England.	O	O	
	[10] Zbigniew Huminski has admitted to killing.	O	O	
Summary-level Completeness Score		8/10 = 80%	7/10 = 70%	2/5

Table 9: **Qualitative analysis of completeness evaluation:** an example showing the limitations of the existing Likert-scale based evaluation using LLM (G-Eval). In "Model Summary" column, all the statements aligned with keyfacts are underlined with the keyfact number, e.g., [1]. The last row indicates the summary-level completeness scores from human, our fine-grained FineSurE, and likert-scale based G-Eval.

```
You will be given a news article. You will then be given one
summary written for this article.

Your task is to rate the summary on one metric.

Please make sure you read and understand these instructions
carefully. Please keep this document open while reviewing, and
refer to it as needed.


Evaluation Criteria:

Completeness (1-5) - the degree to which the summary includes all
key information present in the source document. A complete
summary accurately captures the main points, ideas, and relevant
details without omitting crucial elements.

Evaluation Steps:

1. Read the news article carefully and identify the main points,
key information, and relevant details.
2. Read the summary and compare it to the article. Check if the
summary captures all essential facts, main ideas, and pertinent
details presented in the original article.
3. Assign a score for completeness based on the Evaluation
Criteria.


Example:

Source Text:

{input text}

Summary:

{summary}


Evaluation Form (scores ONLY):

- Completeness:
```

Figure 6: G-Eval for completeness **without** keyfacts.

```
You will be given a news article. You will then be given one
summary written for this article.

Your task is to rate the summary on one metric.

Please make sure you read and understand these instructions
carefully. Please keep this document open while reviewing, and
refer to it as needed.


Evaluation Criteria:

Conciseness (1-5) - the extent to which the summary presents
information succinctly and without unnecessary elaboration. A
concise summary effectively conveys the essential content of the
source document using clear and concise language, avoiding
redundant or superfluous information.

Evaluation Steps:

1. Read the news article carefully and identify the main points,
key information, and relevant details.
2. Read the summary and compare it to the article. Check if the
summary effectively conveys the essential content of the document
in a concise manner, without unnecessary elaboration or
redundancy.
3. Assign a score for conciseness based on the Evaluation
Criteria.


Example:


Source Text:

{input text}

Summary:

{summary}


Evaluation Form (scores ONLY):

- Conciseness:
```

Figure 7: G-Eval for conciseness **without** keyfacts.

E Qualitative Analysis for Completeness

Table 9 shows an example showing the limitation of the existing Likert-scale based evaluation method, such as G-Eval. Although the model summary includes eight out of ten human keyfacts, G-Eval output a very low completeness score, i.e., two out of five. However, the proposed FineSurE exhibits the percentage score similar to that of human judgement. This is because our framework conducts a human-like keyfact alignment to determine the completeness score, which is more fine-grained than relying solely on Likert-scale judgments. Therefore, a fine-grained evaluation framework has great potential to enhance the quality of automated evaluation using LLMs.

F Fair Comparison with G-Eval

For a truly fair comparison, we recognize the necessity to tune G-Eval for our two dimensions of completeness and conciseness. Hence, we opted to adjust G-Eval's prompts to align with the evaluation dimensions of FineSurE. We employed two

Evaluation Dim.	(a) Completeness			(b) Conciseness		
Method	Summary-level Pearson (↑)	Summary-level Spearman (↑)	System-level Rank (↑)	Summary-level Pearson (↑)	Summary-level Spearman (↑)	System-level Rank (↑)
G-Eval (GPT4)	0.314	0.295	0.908	0.314	0.277	0.582
+ Adjusted Criteria	0.301	0.290	0.756	0.284	0.266	0.504
+ Using Keyfacts	0.546	0.527	0.934	0.453	0.434	0.795
FineSurE (GPT-4)	**0.688**	**0.677**	**0.949**	**0.505**	**0.451**	**0.880**

Table 10: Fair comparison with G-Eval tuned for completeness and conciseness.

variants of G-Eval tailored for the assessment of completeness and conciseness dimensions: one adjusted without the utilization of keyfacts, and another adjusted with the utilization of keyfacts. The two tuned prompts without using key facts are summarized in Figures 6 and 8, while those with using key facts are in Figures 7 and 9.

Table 10 compares FineSurE with two variants of G-Eval regarding completeness and conciseness. The findings illustrate that aligning evaluation criteria does not alter performance ("+Adjusted Criteria"), whereas the incorporation of human keyfacts notably enhances G-Eval ("+Using Keyfacts"). Nonetheless, FineSurE continues to outperform G-Eval, supporting that our fine-grained evaluation yields more precise assessments compared to the Likert-scale ratings used in G-Eval.

G Prompt Engineering

We test several prompt engineering techniques, including instruction format, solving categorization problem, reasoning, and evidence mapping. We summarize the agreement with human scores using different combination of prompting techniques with respect to faithfulness, completeness, and conciseness in Tables 11 - 12, where the figure number 10–15 corresponding to each prompt is enclosed in parentheses in each row.

Regarding the prompt engineering for faithfulness, our most effective prompt involves the incorporation of three techniques: instruction format,

```
You will be given a list of key facts. You will then be given one
summary written for an article.

Your task is to rate the summary on one metric.

Please make sure you read and understand these instructions
carefully. Please keep this document open while reviewing, and
refer to it as needed.


Evaluation Criteria:

Completeness (1-5) - the degree to which the summary includes all
key facts in the source document.

Evaluation Steps:

1. Read the list of key facts.
2. Read the summary and compare it to the key facts. Check if the
summary captures all the key facts.
3. Assign a score for completeness based on the Evaluation
Criteria.


Example:

Key Facts:

{keyfact 1}
{keyfact 2}
   ...
{keyfact M}

Summary:

{summary}


Evaluation Form (scores ONLY):

- Completeness:
```

Figure 8: G-Eval for completeness **with** keyfacts.

```
You will be given a list of key facts. You will then be given one
summary written for an article.

Your task is to rate the summary on one metric.

Please make sure you read and understand these instructions
carefully. Please keep this document open while reviewing, and
refer to it as needed.


Evaluation Criteria:

Conciseness (1-5) - the extent to which the summary presents the
key facts without unnecessary elaboration, avoiding redundant or
superfluous information.

Evaluation Steps:

1. Read the list of key facts.
2. Read the summary and compare it to the key facts. Check if the
summary effectively conveys the essential content of the key
facts in a concise manner, without unnecessary elaboration or
redundancy.
3. Assign a score for conciseness based on the Evaluation
Criteria.


Key Facts:

{keyfact 1}
{keyfact 2}
   ...
{keyfact M}

Summary:

{summary}


Evaluation Form (scores ONLY):

- Conciseness:
```

Figure 9: G-Eval for conciseness **with** keyfacts.

```
Is the summary sentence supported by the transcript? Response
with "Yes" or "No" for each sentence in the summary.

Provide your answer in JSON format. The answer should be a list
of dictionaries whose keys are "sentence" and "response":
[{"sentence": "first sentence", "response": "yes or no"},
{"sentence": "second sentence", "response": "yes or no"},
{"sentence": "third sentence", "response": "yes or no"},]

Transcript:
{input text}

Summary with N sentences:
{summary sentence 1}
{summary sentence 2}
          ...
{summary sentence N}
```

Figure 10: Basic prompt for faithfulness evaluation.

Technique	Sentence-lev. bAcc (True Pos., True Neg.)	Summary-lev. Pearson	Summary-lev. Spearman	System-lev. Rank
Basic Prompt (Figure 10)	92.9% (91.9%, 94.0%)	0.856	0.850	0.933
+ Inst. + Cat. (Figure 11)	91.3% (90.1%, 92.5%)	0.841	0.834	0.933
+ Inst. + Cat. + Rea. (Figure 3)	92.0% (91.9%, 92.1%)	0.844	0.841	0.933
+ Inst. + Cat. + Evi. (Figure 12)	91.7% (88.7%, 94.7%)	0.829	0.821	0.933
+ Inst. + Cat. + Rea. + Evi. (Figure 13)	91.5% (90.7%, 92.3%)	0.839	0.836	0.933

Table 11: Prompt engineering with Llama3-70B-Inst. for **faithfulness evaluation** using instruction (Inst), categorization (Cat), reasoning (Res), and evidence mapping (Evi) techniques.

Dimension	(a) Completeness			(b) Conciseness		
Technique	Summary-level Pearson	Summary-level Spearman	System-level Rank	Summary-level Pearson	Summary-level Spearman	System-level Rank
Basic Prompt (Figure 14)	0.658	0.648	0.891	0.415	0.402	0.875
+ Inst. (Figure 4)	0.775	0.747	0.881	0.445	0.444	0.786
+ Inst. + Rea (Figure 15)	0.664	0.650	0.926	0.424	0.400	0.785

Table 12: Prompt engineering with Llama3-70B-Inst. for **completeness and conciseness evaluation** using instruction (Inst) and reasoning (Res) techniques. The values in parenthesis represent p-values.

```
You will receive a transcript followed by a corresponding summary.
Your task is to assess the factuality of each summary sentence
across nine categories:
* no error: the statement aligns explicitly with the content of
the transcript and is factually consistent with it.
* out-of-context error: the statement contains information not
present in the transcript.
* entity error: the primary arguments (or their attributes) of
the predicate are wrong.
* predicate error: the predicate in the summary statement is
inconsistent with the transcript.
* circumstantial error: the additional information (like location
or time) specifying the circumstance around a predicate is wrong.
* grammatical error: the grammar of the sentence is so wrong that
it becomes meaningless.
* coreference error: a pronoun or reference with wrong or non-
existing antecedent.
* linking error: error in how multiple statements are linked
together in the discourse (for example temporal ordering or
causal link).
* other error: the statement contains any factuality error which
is not defined here.

Instruction:
First, compare each summary sentence with the transcript.
Second, classify each sentence into one of the nine categories
and then provide the classified category.

Provide your answer in JSON format. The answer should be a list
of dictionaries whose keys are "sentence" and "category":
[{"sentence": "first sentence", "category": "no error"},
{"sentence": "second sentence", "category": "out-of-context
error"}, {"sentence": "third sentence", "category": "entity
error"},]

Transcript:
{input text}

Summary with N sentences:
{summary sentence 1}
{summary sentence 2}
        ...
{summary sentence N}
```

Figure 11: Basic prompt with instruction format and categorization for faithfulness evaluation.

```
You will receive a transcript followed by a corresponding summary.
Your task is to assess the factuality of each summary sentence
across nine categories:
* no error: the statement aligns explicitly with the content of
the transcript and is factually consistent with it.
* out-of-context error: the statement contains information not
present in the transcript.
* entity error: the primary arguments (or their attributes) of
the predicate are wrong.
* predicate error: the predicate in the summary statement is
inconsistent with the transcript.
* circumstantial error: the additional information (like location
or time) specifying the circumstance around a predicate is wrong.
* grammatical error: the grammar of the sentence is so wrong that
it becomes meaningless.
* coreference error: a pronoun or reference with wrong or non-
existing antecedent.
* linking error: error in how multiple statements are linked
together in the discourse (for example temporal ordering or
causal link).
* other error: the statement contains any factuality error which
is not defined here.

Instruction:
First, compare each summary sentence with the transcript.
Second, find the exact quote which can confirm the factual
consistency of the sentence and insert them. If you cannot, write
"not mentioned".
Third, classify each sentence into one of the nine categories and
then provide the classified category.

Provide your answer in JSON format. The answer should be a list
of dictionaries whose keys are "sentence", "quote", and
"category":
[{"sentence": "first sentence", "quote": "identified quote",
"category": "no error"}, {"sentence": "second sentence", "quote":
"not mentioned", "category": "out-of-context error"}, {"sentence":
"third sentence", "quote": "identified quote", "category":
"entity error"},]

Transcript:
{input text}

Summary with N sentences:
{summary sentence 1}
{summary sentence 2}
        ...
{summary sentence N}
```

Figure 12: Basic prompt with instruction format, categorization, and evidence mapping for faithfulness evaluation.

solving categorization, and providing reasoning, as seen in in Figure 3. This prompt shows the highest overall agreement with human judgments among those that included fine-grained evaluation, such as error categorization.

On the other hand, for completeness and conciseness, the inclusion of the reasoning step results in a decrease in agreement during evaluation. Additionally, categorization is unnecessary for keyfact alignment, given the presence of only two classes–"no matched" and "matched"–for each keyfact. Consequently, the most effective prompt involves solely utilizing the instruction format in Figure 4.

```
You will receive a transcript followed by a corresponding summary.
Your task is to assess the factuality of each summary sentence
across nine categories:
* no error: the statement aligns explicitly with the content of
the transcript and is factually consistent with it.
* out-of-context error: the statement contains information not
present in the transcript.
* entity error: the primary arguments (or their attributes) of
the predicate are wrong.
* predicate error: the predicate in the summary statement is
inconsistent with the transcript.
* circumstantial error: the additional information (like location
or time) specifying the circumstance around a predicate is wrong.
* grammatical error: the grammar of the sentence is so wrong that
it becomes meaningless.
* coreference error: a pronoun or reference with wrong or non-
existing antecedent.
* linking error: error in how multiple statements are linked
together in the discourse (for example temporal ordering or
causal link).
* other error: the statement contains any factuality error which
is not defined here.

Instruction:
First, compare each summary sentence with the transcript.
Second, find the exact quote which can confirm the factual
consistency of the sentence, and insert them. If you cannot,
write "not mentioned".
Third, provide a single sentence explaining which factuality
error the sentence has.
Forth, answer the classified error category for each sentence in
the summary.

Provide your answer in JSON format. The answer should be a list
of dictionaries whose keys are "sentence", "quote", "reason", and
"category":
[{"sentence": "first sentence", "quote": "identified quote",
"reason": "your reason", "category": "no error"}, {"sentence":
"second sentence", "quote": "not mentioned", "rationale": "your
reason", "category": "out-of-context error"}, {"sentence": "third
sentence", "quote": "identified quote", "reason": "your reason",
"category": "entity error"},]

Transcript:
{input text}

Summary with N sentences:
{summary sentence 1}
{summary sentence 2}
        ...
{summary sentence N}
```

Figure 13: Basic prompt with instruction format, categorization, reasoning and evidence mapping for faithfulness evaluation.

```
You are given a summary and some semantic content units. For each
key fact, mark "Yes" if it can be inferred from the summary,
otherwise mark "No"
If "Yes", please indicate the line number(s) of the summary
sentence(s) relevant to the key fact.

Provide your answer in JSON format. The answer should be a list
of dictionaries whose keys are "key fact", "response", and "line
number":
[{"key fact": "first key fact", "response": "Yes", "line number":
[1]}, {"key fact": "second key fact", "response": "No", "line
number": []}, {"key fact": "third key fact", "response": "Yes",
"line number": [1, 2, 3]}
]

Summary:
[1] {summary sentence 1}
[2] {summary sentence 1}
          ...
[N] {summary sentence 1}

M key facts:
{keyfact 1}
{keyfact 2}
   ...
{keyfact M}
```

Figure 14: Basic prompt for completeness and conciseness evaluation.

H Summarization using LLMs

We consistently employ the prompt shown in Figure 16 across all LLMs, generating model summaries based on the given input text.

Furthermore, since the model summary is null when parsing fails, there is no hallucination, but also no summary sentences align with any key facts. Therefore, in cases where the LLMs produce incorrect JSON outputs that cannot be parsed, the scores for faithfulness, completeness, and conciseness are automatically set to 1.0, 0.0, and 0.0, respectively.

```
You will receive a summary and a set of key facts for the same
transcript. Your task is to assess if each key fact is inferred
from the summary.

Instruction:
First, compare each key fact with the summary.
Second, provide a single sentence explaining whether the key fact
is inferred from the summary.
Third, respond "Yes" (inferred) or "No" (not inferred) for each
key fact. If "Yes", specify the line number(s) of the summary
sentence(s) relevant to each key fact.

Provide your answer in JSON format. The answer should be a list
of dictionaries whose keys are "key fact", "reason", "response",
and "line number":
[{"key fact": "first key fact", "reason": "your reason",
"response": "Yes", "line number": [1]}, {"key fact": "second key
fact", "reason": "your reason", "response": "No", "line number":
[]}, {"key fact": "third key fact", "reason": "your reason",
"response": "Yes", "line number": [1,2,3]]}]

Summary:
[1] {summary sentence 1}
[2] {summary sentence 1}
          ...
[N] {summary sentence 1}

M key facts:
{keyfact 1}
{keyfact 2}
   ...
{keyfact M}
```

Figure 15: Basic prompt with instruction format and reasoning for completeness and conciseness evaluation.

```
Text: {input text}

Instruction: Summarize the Text.

Provide your answer in JSON format. The answer should be a
dictionary with the key "summary" containing a generated summary
as a string:
{"summary": "your summary"}

JSON Output:
```

Figure 16: Prompt to get the summary from LLMs.

I Extension of REALSumm

Although the REALSumm data contains human labels indicating which keyfacts are included in the model summary, there are no human labels indicating which summary sentences align with the keyfacts. The former is used to compute the ground-truth completeness score, while the latter is used to compute the conciseness score. Therefore, we conducted a human evaluation to verify which summary sentences align with the set of key facts. Specifically, three human annotators were asked to mark "yes" if at least one keyfact in the keyfact list could be inferred from each summary sentence, otherwise "no". This is quite simple task compared with faithfulness evaluation, since human ground-truth keyfacts are available in REALSumm data. The extended dataset is available with our FineSurE framework at https://github.com/DISL-Lab/FineSurE.