SIS-Fact: Towards *Systematic*, *Interpretable* and *Scalable* Factuality Evaluation for LLM

Anonymous ACL submission

Abstract

Despite Large Language Models' advances, document-grounded generation still suffers from factual errors. Current evaluations oversimplify error analysis by applying binary judgements, while costly human-annotated datasets contain under-representative error distributions. To address these challenges, we propose a novel framework named SIS-Fact (**S**ystematic, **I**nterpretable and **S**calable **Fact**uality Evaluation), which integrates systematic error typologies, synthetic data generation pipelines, and high-quality interpretable annotations for comprehensive factuality evaluation. Specifically, we first develop ten diverse methods to synthesize six error types in grounded generation, including both intrinsic and extrinsic errors. In this way, we develop SIS-Fact Dataset, a high-quality document-grounded factuality evaluation dataset characterized by challenging errors and interpretable error analysis. Based on SIS-Fact Dataset, we introduce SIS-Fact-Evaluator, an advanced factuality evaluation model capable of fine-grained analysis and correction. Our extensive experiments show that SIS-Fact-Evaluator achieves SOTA performance in SIS-Fact Datasetwhile maintaining strong generalization across existing multiple factuality benchmarks.

1 Introduction

Recent breakthroughs in Large Language Models (LLMs) have fundamentally transformed the paradigm of human-computer interaction (Achiam et al., 2023; Team, 2024; DeepSeek-AI et al., 2025). However, despite their impressive fluency and broad real-world utility, LLMs are prone to producing factual inaccuracies (also known as hallucinations) (Ji et al., 2023; Huang et al., 2025) in their outputs, posing significant risks and severely compromising their credibility. While factual inaccuracies may stem from models' limited knowledge of the external world, they frequently occur even in grounded generation tasks such as text summarization (Zhang et al., 2024a; Zhu et al., 2025) and question answering (Wang et al., 2024, 2025), where models fail to adhere to the content and facts provided in reference documents.

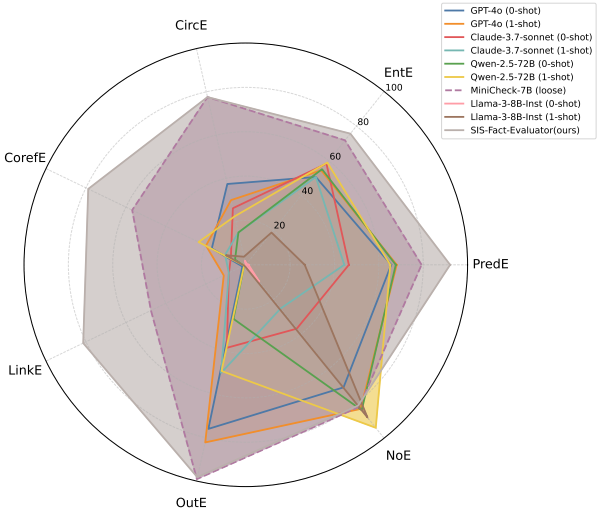

Figure 1: SIS-Fact-Evaluator outperforms frontier models across all error types in SIS-Fact Dataset. MiniCheck-7B (second-best) operates in binary setting, unable to categorize errors. Notably, while some competitors excel in No-Error precision, SIS-Fact-Evaluator maintains balanced accuracy for both factual and erroneous cases.

To address the aforementioned issue, several efforts (e.g., entailment-based (Kryściński et al., 2019; Goyal and Durrett, 2021a; Maynez et al., 2020), question-answering-based (Wang et al., 2020; Durmus et al., 2020; Fabbri et al., 2022), atomic-fact-based (Min et al., 2023) , and synthetic-data-based (Tang et al., 2024a) methods) have been developed to evaluate the factual accuracy of LLM outputs. However, they treat factual consistency as a binary property, that is, simply determining whether generated text is true or false, thereby overlooking the diversity and nuance of factual errors, as well as lacking in interpretability. Recently, sev-

eral studies (Pagnoni et al., 2021a; Cao and Wang, 2021; Zhong and Litman, 2025) have emphasized the importance of factual error typology and attempted to overcome the mentioned pitfalls by constructing datasets annotated with various types of errors for the training of factual evaluation models.

However, these datasets present several notable limitations: **(1) High annotation cost and low scalability:** Existing datasets rely on manual labeling, which is resource-intensive, challenging to scale, and subject to annotator disagreement, thereby limiting dataset size and quality. **(2) Lack of interpretability and fine-grained annotation:** They predominantly employ document- or sentence-level annotations without pinpointing error locations, explaining the underlying reasoning, or suggesting corrections. Such label-only annotations significantly restrict interpretability, hindering both quick identification of error sources and revision of inaccurate content. This limitation necessitates extensive manual analysis and reduces the system's applicability in critical scenarios such as academic citation verification or news fact-checking (DeVerna et al., 2024; Thorne and Vlachos, 2018). **(3) Oversimplification of errors:** Most existing datasets are derived from relatively simple corpora (e.g., short summaries from CNN/DailyMail and XSum (Narayan et al., 2018)) and are annotated based on generations from weaker baseline models like BERTSum (Liu and Lapata, 2019) and PEGASUS (Zhang et al., 2020). This results in distributions dominated by obvious, low-level errors that fail to capture the nuanced and subtle error types characteristic of state-of-the-art LLMs, creating a misalignment with contemporary machine-generated error distributions.

To bridge these gaps, we propose a novel framework called **SIS-Fact**, which integrates error typologies, synthetic data generation pipelines, and fine-grained annotations for comprehensive factuality evaluation. Specifically, inspired by error typology (Pagnoni et al., 2021a), we first develop diverse methodologies to synthesize six error types in grounded generation, including both intrinsic and extrinsic errors. In this way, we develop SIS-Fact Dataset, a novel dataset characterized by longer summaries and challenging document contexts. Enriched with reference analyses, error explanations, and correction guidelines, SIS-Fact Dataset enables deeper investigation of both error sources and solutions. Notably, human evaluations further validate the high quality and validity of our dataset.

Building upon SIS-Fact Dataset, we introduce SIS-Fact-Evaluator, an advanced factuality evaluation model capable of fine-grained analysis. The model performs comprehensive tasks including identifying relevant document references, detecting error types, locating specific errors, and providing actionable corrections. Our extensive experiments demonstrate that SIS-Fact-Evaluator achieves state-of-the-art performance in our dataset while maintaining strong generalization across diverse existing benchmarks, covering factuality evaluation benchmarks in summarization, QA and RAG scenarios. Additionally, the structured outputs of SIS-Fact-Evaluator offer fine-grained analysis and high interpretability.

To sum up, we highlight our contributions as follows:

- We propose a new framework SIS-Fact, which is cost-effective, automated pipeline for generating factual error datasets, scalable across various tasks and capable of producing arbitrarily large datasets. To our knowledge, our work is the first to focus on high-quality benchmark designed specifically for the challenges in systematically detecting factual inconsistencies without manual annotation.

- We develop a systematic, fine-grained factuality evaluation dataset that incorporates error categorization and fine-grained annotations, including explanations and correction instructions. Also, we design a model, namely SIS-Fact-Evaluator, which is capable of performing detailed and interpretable factuality evaluation tasks.

- We conduct extensive experiments on SIS-Fact Dataset and other existing benchmarks to show that our model has strong competitiveness and generalization ability compared with other state-of-the-art baselines. Meanwhile, ablation studies demonstrate efficacy and indispensability for core modules.

2 Related Work

Multiple studies have investigated whether large language models (LLMs) can generate factually accurate content. These works broadly categorize into three strands—evaluation, root cause analysis (Massarelli et al., 2020; Lu et al., 2022; Luo et al., 2023b; Liu et al., 2023a; Luo et al., 2023a), and mitigation approaches (Lee et al., 2022; Dai

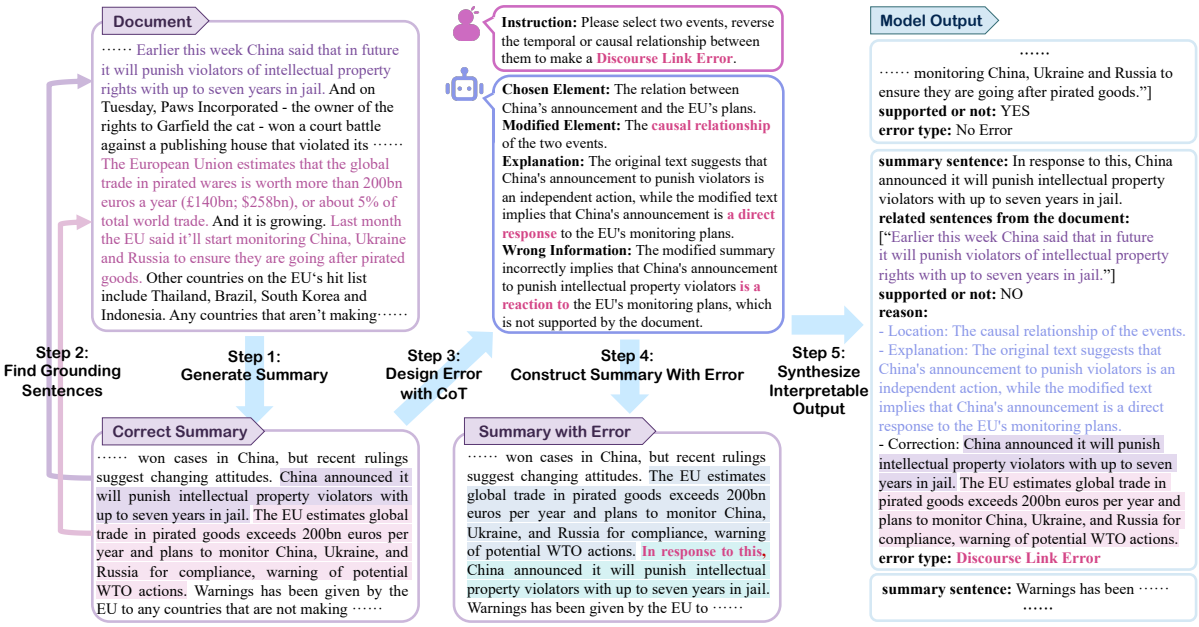

Figure 2: Overview of the SIS-Fact pipeline. Some of the text are simplified for better demonstration.

et al., 2022; Borgeaud et al., 2022; Moiseev et al., 2022; Asai et al., 2023; Du et al., 2024)—while our research focuses on the evaluation dimension.

Kryściński et al. (2019) first argued that factuality assessment in abstractive summarization should transcend overlap-based metrics like ROUGE, introducing entailment models to verify whether generated claims are supported by the source context—a direction also explored by Maynez et al. (2020) and Goyal and Durrett (2021a). Early alternatives employed question-answering models to check context-summary consistency (Wang et al., 2020; Durmus et al., 2020; Fabbri et al., 2022), and Zha et al. (2023) and Ribeiro et al. (2022) later improved performance via model ensembling and semantic-graph representations, respectively.

Building on these foundations, researchers have precisely annotated factual errors in machine-generated summaries to assemble datasets for quantitative factuality assessment (Fabbri et al., 2021; Cao and Wang, 2021; Pagnoni et al., 2021b; Zhang et al., 2024b; Tang et al., 2024a; Zhong and Litman, 2025). Despite their utility, these datasets exhibit significant limitations, as detailed in Section 1.

3 Methodology

In this section, we elaborate on our proposed framework, SIS-Fact, in five steps: *Generate Summary*, *Find Grounding Sentences*, *Design Error with CoT*, *Construct Summary With Error* and *Synthesize Interpretable Output*. Figure 2 illustrates the pipeline of SIS-Fact. Specifically, we first generate a grounded summary dataset from a set of documents, with each summary sentence labeled with grounding sentences from the document. Then, we construct erroneous summaries based on various error designs, covering all error categories. With the Chain-of-Thought output from the error construction process, we synthesize gold output, consisting of sentence-by-sentence analysis, grounding sentence extraction, and interpretable reasoning. Notably, the entire process imposes no constraints on the size or type of the original document dataset. It can be scaled to arbitrary large sizes by adjusting the error construction position and generating new summaries. Examples of the prompts for all stages in the pipeline can be found at Appendix D.

3.1 Grounded Summary Dataset

Our pipeline starts with an arbitrary document dataset, which can include pre-existing summaries or simply the documents themselves. We base our factuality evaluation system on summarization tasks because, by nature, all claims in a summary are expected to be grounded in the source document. This inherent characteristic makes summarization a suitable foundation. Section 5.2 demonstrates that our pipeline and model possess strong generalizability to the evaluation of other document-grounded generation tasks.

For clarity and to better illustrate the full process, we demonstrate using a document-only dataset. To

be specific, for each document in the dataset, we first prompt LLMs to generate summaries for the document. Subsequently, we apply an extract-and-rewrite process to ensure the grounding-based factuality of each summary sentence: 1) LLMs are prompted to locate the grounding sentences from the source document for each sentence in these summaries. 2) A voting mechanism involving three LLMs is applied to determine whether a summary sentence is sufficiently supported by its grounding sentences. If a sentence lacks adequate support, it undergoes a rewriting process, and the voting process repeats until complete support from the reference is achieved. Also, the aforementioned process serves as the preparation for the synthetic model output. Ultimately, we achieve a set of summaries, each with sentence-level grounding evidences. Importantly, due to the flexibility of the base dataset, our pipeline can be applied to any document dataset, supporting the scalability of SIS-Fact.

3.2 Systematic Error Data Construction with Category Distinction

Previous studies have extensively explored the typology of factual errors (Pagnoni et al., 2021a; Goyal and Durrett, 2021b). In our error construction process, we follow the fine-grained error typology proposed by Pagnoni et al. (2021a) for text summarization, which also generalizes to other grounded generation tasks. We exclude the category of Grammatical Error as it is not a factual error and has been broadly addressed by modern LLMs. Table 1 summarizes the error categories and provides examples of error construction. Thereafter, we detail the construction process for each error category.

Semantic Frame Errors Semantic frame errors involve incorrect elements in semantic frames, such as predicate, entity, or circumstance. We adapt swapping methods from previous works (Kryściński et al., 2019; Cao and Wang, 2021), prompting LLMs to select an element (e.g., an entity) to swap with a congeneric element from the summary or source document. Additionally, we introduce two novel strategies to address common yet subtle errors in summarization tasks, which are hard to detect and cannot be constructed by simple swapping:

- *Modifying Predictions*: This strategy addresses errors arising from the confusion between speculative language (e.g., predictions, hypotheses) and factual statements. While such errors may preserve the original predicate, they alter the statement's semantic certainty. We guide LLMs to detect these subtle shifts by analyzing modal verbs (e.g., those indicating attitude, speculation, permission, or obligation) and predictive phrases (e.g., "predict" and "suppose").
- *Compressing Words*: This strategy targets errors where specific terms or parallel entities are oversimplified, altering their original meaning. For instance, simplifying "net revenue attributable to parent company" to "net revenue" distorts semantics. Such error are particularly difficult to detect due to their lexical overlap with the source text. To ensure high-quality error construction, we implement a two-stage verification by prompting models to filter out examples and omission of whole event-triplet. Of note, we find that neither entailment-based nor atomic-fact-extraction models can reliably identify these errors, which proves the high-degree challenge of our dataset.

Discourse Errors Discourse errors refer to errors beyond a single semantic frame, sometimes involving inconsistencies across multiple sentences. To address the complex nature of discourse errors, we systematically investigate three representative error-generation strategies that manipulate discourse-level semantic relations to introduce plausibly inconsistent narratives. Specifically,

- *Swapping Pronouns*: We follow Kryściński et al. (2019) to generate a co-reference error by extracting pronouns and swapping it with another pronoun in the same group. While Kryściński et al. (2019) focuses solely on gender-specific pronouns, our strategy extends to all pronoun types. Meanwhile, to enhance diversity, we additionally implement a two-step transformation: first converting entities to pronouns, then replacing these with alternative pronouns. This dual process intentionally reduces referential specificity, thereby introducing controlled ambiguity.
- *Merging Sentences*: A common co-reference error arises when two events involving distinct yet akin subjects are erroneously conflated into a single narrative within the summary. To

Categorization	Method
Predicate Error (PredE)	Swapping Relation, Modifying Predictions
Entity Error (EntE)	Swapping Entities, Compressing Words
Circumstance Error (CircE)	Swapping Circumstances
Co-Reference Error (CorefE)	Swapping Pronouns, Merging Sentences
Discourse Link Error (LinkE)	Reverse Logical Relationship
Extrinsic Error (OutE)	Introducing Extrinsic Information

Table 1: The error constructing system of SIS-Fact, full categorization and examples are shown in Table 6

simulate this, we select two sentences with similar but different subjects, retain only one subject, and merge the sentences into one.

- *Reverse Logical Relationship*: Discourse link error originates from inaccuracies in the discourse relationship between different statements. Such error is particularly challenging to identify, sometimes eluding even human annotators' consensus (Pagnoni et al., 2021a). Our strategy involves instructing LLMs to first recognize two events in the document that have a temporal or causal relationship. Then, we invert this relationship, and finally rewrite the corresponding summary sentences to reflect this altered relationship.

Extrinsic Errors Extrinsic error, also termed out-of-article error, indicates information not derived from the reference document. Such error is inherently elusive due to their subtlety and the contextual ambiguity they entail. Even human annotators may struggle to distinguish between intrinsic and extrinsic errors (Tang et al., 2023). Given the difficulty of assessing whether all supporting information for a fact has been removed, we opt to prompt LLMs to insert extrinsic information not mentioned in the document into a specific sentence, rather than deleting sentences from the document.

3.3 Interpretable Factuality Data Construction

Our error design employs a Chain-of-Thought process. When applying each error construction strategy, the model must first select the target sentence(s) and specific element to modify. Next, it applies the modification and explains how this change alters the sentence meaning. Finally, it reports the modified element, the revised sentence, and analyzes the incorrect information introduced. This Chain-of-Thought approach not only improves error construction quality by prompting step-by-step generation but also supplies fine-grained data for interpretable analysis of synthetic model outputs.

Specifically, the output takes a structured, sentence-by-sentence way. For each sentence, it includes five components: First, the model repeats the **summary sentence**. Second, it extracts **relevant sentence(s) from the document**, utilizing the grounding sentence(s) from Section 3.1. Third, the model determines **whether the summary sentence is supported**. If not, it provides a **reason**. Here, we rephrase the Chain-of-Thought output to prompt reasoning: identifying the error's exact location, explaining it, and correcting the sentence. Finally, the model gives the sentence's **error label decision** as "No Error" or a specific error type from Section 3.2 (e.g., "Co-reference Error"). We provide specific examples in Appendix B.

Our design brings two benefits: 1) **Interpretability through traceable reasoning.** The interpretability of SIS-Fact-Evaluator stems from its transparent reasoning process, which mitigates the black-box nature of the model, making its decision-making traceable. Additionally, by supplying grounding sentences and possible corrections, we save time otherwise spent browsing the entire document. Also, comparing correct and incorrect summary sentences makes errors more obvious. 2) **Higher quality from test-time scaling.** The customized output structure strengthens control. The model focuses on specific grounding contexts and the reasoning process guiding the final judgment. Thus, it can detect difficult errors needing more logical reasoning.

4 The SIS-Fact Dataset

Based upon the SIS-Fact pipeline, we construct SIS-Fact Dataset, which not only capable for training factuality evaluators, but also a high-quality, challenging benchmark for assessing the error detecting, error categorizing, and explanation

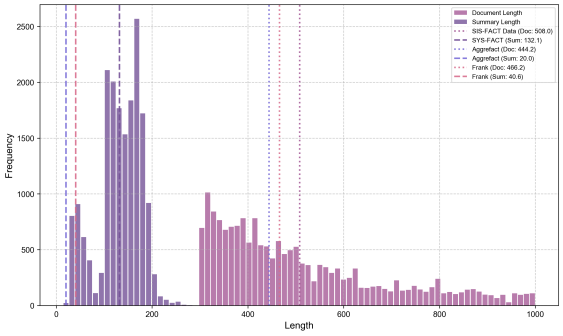

Figure 3: Document and summary length (words) distribution of SIS-Fact Datasetwith average length (words) comparison

generating abilities for factuality evaluators.

4.1 Dataset Construction

Existing summary datasets have length limitations and some auto-extracted summaries lack complete factuality. Therefore, we build our dataset starting with the document itself. Concretely, we randomly select diverse-length news and encyclopedic documents from the BBC News Summary (Gupta et al., 2022) and the DetNet Wikipedia (Xu and Lapata, 2019) Datasets. To ensure robust, unbiased grounded-summary generation, we use multiple LLMs at each step. Details on data selection and construction are in Appendix A. For each document-summary pair, we leverage the designed error-construction strategies to generate one sample per each error, with the original summary as "No Error" sample, obtaining 10 samples in total. To prevent training data leakage, we split dataset based on document-(original)summary pairs. This ensures the model doesn't encounter trained summaries during testing, avoiding potential memorization-based unfairness.

4.2 Analysis for SIS-Fact Dataset

SIS-Fact Dataset comprises $18,093$ samples, with the train/validation/test set having $15660/1192/1241$ samples. Meanwhile, we report the distribution of the document and summary length for our dataset is shown in Figure 3. One can observe that compared with datasets with similar length, SIS-Fact Dataset consists of longer summaries. This lower compression of the document ensures more detail in the summary, making the errors more subtle. Importantly, we conduct Human Evaluation on a random sample of 100 instances from SIS-Fact Dataset. Human experts (all of whom are Ph.D. or Master students) label the quality of summaries, grounding sentences and the validity of error constructions. Table 8 shows a 95% agreement on the error constructions, and 78% overall full correctness (Please refer to Table 8 in Appendix E). Most mistakes occur in writing summaries and extracting grounding sentences, confirming the vulnerabilities of LLMs in grounded fact generation and evaluation.

5 Experiment

5.1 Experimental Setup

Baselines. In our experiments, we choose the Llama-3-8B model (AI@Meta, 2024) from Meta as the base model for SIS-Fact-Evaluator, and trained on the training set of SIS-Fact Dataset. Further training details can be found in Appendix C

To gauge the effectiveness of SIS-Fact-Evaluator, we compare several baselines, which includes **open-source models**—Llama-3-8B-Instruct (AI@Meta, 2024) (denoted as Llama-3-8B-Inst), and Qwen2.5-72B-Instruct (denoted as Qwen2.5-72B) (Team, 2024)—as well as **closed-source models**, namely GPT-4o (Jaech et al., 2024), Claude-3.7-Sonnet-20250219 (denoted as Claude-3.7) (Anthropic, 2025). Notably, we conduct both zero-shot and one-shot testing on these models. We intentionally limit the number of demonstration examples to avoid performance degradation caused by excessive prompt length (see Appendix D for detailed prompts). Furthermore, we adopt **MiniCheck-7B** (Tang et al., 2024a)—the largest model in state-of-the-art binary evaluator—as our baseline for specialized factuality assessment. Given MiniCheck-7B's binary (0/1) classification output, we evaluate its performance through a lenient scoring scheme where all error categories are uniformly mapped to the 0 label.

Other Benchmarks. To evaluate SIS-Fact-Evaluator's out-of-distribution generalization, we conduct experiments across existing multiple factuality evaluation benchmarks, including *Summarization Tasks*: **FRANK** (A human-annotated benchmark aligned with our error taxonomy) (Pagnoni et al., 2021a), **CNN** (news article with a different data source) from AggreFact (Tang et al., 2023), and **MediaSum** (Dialogue-based data with different data types) from TofuEval (Tang et al., 2024b); *Document-Grounded Tasks*: **ClaimVerify** (Liu et al., 2023b) for search engine responses, **ExpertQA** (Malaviya et al., 2024) for expert-curated

Model	PredE	EntE	CircE	CorefE	LinkE	OutE	NoE	Avg(Type)	Avg(Item)
GPT-4o	65.56	50.64	37.38	17.01	1.69	75.89	70.68	45.55	47.78
GPT-4o※	68.05	54.49	29.91	19.92	11.02	82.14	83.08	49.80	52.78
Claude-3.7	46.47	58.33	26.17	7.47	8.47	38.39	36.84	31.73	32.23
Claude-3.7※	44.40	50.64	14.95	10.37	8.47	50.00	25.19	29.15	29.01
Qwen2.5-72B	67.63	55.13	14.95	5.39	0.85	25.00	83.83	36.11	42.71
Qwen2.5-72B※	65.15	58.97	22.43	23.65	0.85	49.11	**93.98**	44.88	51.25
MiniCheck-7B	(79.25)	(71.79)	(77.57)	(56.85)	(47.46)	(99.11)	(81.95)	(73.43)	(73.17)
Llama-3-8B-Inst	1.24	0.64	1.87	0.00	0.00	0.00	9.77	1.93	2.58
Llama-3-8B-Inst※	26.56	18.59	3.74	9.96	0.85	0.89	87.97	21.22	28.77
SIS-Fact-Evaluator	**92.12**	**75.64**	**77.57**	**78.84**	**81.36**	**98.21**	81.20	**83.56**	**83.40**

Table 2: Main results (%) on SIS-Fact Dataset. We display the precision of each model on the categorized setting. Avg(Type) took average by type, and Avg(Item) took average by each data item. Best performances are marked bold. ※ means the model is tested in one-shot settings, otherwise in zero-shot settings. Results of MiniCheck-7B is enclosed in parentheses for it's been tested in a looser setting (0/1 label rather than output the error type).

QA, **RAGTruth** (Niu et al., 2024) for retrieval-augmented generation scenarios. Note that we test in categorized settings for FRANK, and binary setting for other benchmarks as they do not have error categories. We will release our code and dataset upon paper acceptance.

5.2 Main Results

We report the results of different models on SIS-Fact Dataset in Table 2 and other benchmarks in Table 3, as well as demonstrate the following claims for SIS-Fact-Evaluator and SIS-Fact Dataset:

SIS-Fact-Evaluator consistently outperforms all baselines on SIS-Fact Dataset, especially on complex data. As shown in Tabl 2, SIS-Fact-Evaluator achieves remarkable improvements of 33.76% (error-type-averaged) and 30.62% (item-averaged) in terms of precision over the second-best performer, one-shot GPT-4o. Meanwhile, the performance advantage is most pronounced for discourse-level errors (*CorefE* and *LinkE*), where SIS-Fact-Evaluator outperforms baselines by up to 8×. This substantial gap highlights two key strengths: 1) Cross-frame analysis capability: Effective handling of errors spanning multiple semantic frames; 2) Structured reasoning: Our categorized error construction and traceable reasoning process specifically address LLMs' limitations in complex factual analysis.

SIS-Fact-Evaluator generalizes well to other datasets and tasks. From Table 3, one can observe that although only trained on document-summary pairs, SIS-Fact-Evaluator not only achieves good performance on summary-based factuality evaluation, but also generalizes well to other tasks, which justifies the universality of our summarization-based generation pipeline.

SIS-Fact brings performance gain for base models. By comparing our model trained on Llama-3-8B (base) with the results obtained from prompting Llama-3-8B-Instruct using detailed output prompts and in-context learning example (Llama-3-8B-Inst※), we find that the improvement in our model's performance is not solely attributable to instruction-guided chain-of-thought reasoning. Instead, it primarily stems from the model's ability to learn the underlying nature of the errors from our carefully constructed dataset, thereby enabling more accurate reasoning in identifying factuality errors.

The SIS-Fact Dataset is a more challenging benchmark. Comparing the results on SIS-Fact Dataset and other datasets, it is evident that our dataset is more challenging than existing benchmarks. On other datasets, frontier models like GPT-4o can achieve scores ranging from 60% to 85% under the zero-shot setting (see Table 3). However, on our dataset, the highest score is only about 50% even in in-context settings (see Table 2), with scores for the most difficult error types dropping below 20%. This confirms that our dataset contains a higher proportion of difficult instances than those constructed using a "model-first-then-annotate" paradigm, establishing it as a high-quality benchmark for factual consistency evaluation.

5.3 Further Analysis on Fine-grained Test Setting

To assess error localization capability, we conducted fine-grained experiments analyzing model

Model	Summary-Related Tasks			Generalization Task		
	FRANK(*sys*)	CNN	MediaSum	Claim Verify	ExpertQA	RAGTruth
GPT-4o	71.20	68.10	71.40	69.00	59.60	84.30
Claude-3.7	51.00	48.21	63.91	45.61	22.49	65.83
Qwen2.5-72B	75.43	63.60	71.90	70.00	60.10	81.90
MiniCheck-7B	(75.37)	64.70	**76.30**	75.40	59.40	84.10
Llama-3-8B-Inst	70.41	79.70	71.80	70.80	52.90	76.83
SIS-Fact-Evaluator	**81.11**	**85.30**	73.10	**77.70**	**68.90**	**85.80**

Table 3: Main results (%) on other datasets. Best performances are marked bold. Results of MiniCheck-7B on FRANK is enclosed in parentheses for it's been tested in a looser setting (0/1 label rather than output the error type).

Model	Avg(Type)	Avg(Item)	PAR(Item)
GPT-4o※	45.38	48.35	**91.61**
claude-3.7※	23.99	24.42	84.18
Qwen2.5-72B※	37.14	43.35	84.59
MiniCheck-7B	(55.46)	(56.89)	(77.75)
Llama-3-8B-Inst※	15.16	20.79	72.26
SIS-Fact-Evaluator	75.51	76.15	**91.31**

Table 4: Fine-grained results (%) and the Precision Alignment Ratio (PAR) on SIS-Fact Dataset. ※ means the model is tested in one-shot settings. Results of MiniCheck-7B is enclosed in parentheses for it's been tested in a looser setting (0/1 label rather than output the error type).

Index	**R**eason	**G**round	**S**ent	Avg(Type)	Avg(Item)
RGS	✓	✓	✓	83.56	83.4
GS	-	✓	✓	72.6	73.57
RS	✓	-	✓	63.08	68.65
RG	✓	✓	-	55.26	59.23
R	✓	-	-	28.3	29.81
G	-	✓	-	49.05	48.83
S	-	-	✓	16.52	20.47
raw	-	-	-	19.37	26.75

Table 5: Result (%) for ablation study. "Reason" means outputting synthesized error explanation an correction, "Ground" means outputting grounding document sentences, "Sent" means outputting sentence-by-sentence.

outputs at the sentence level. This evaluation requires correct error type classification, precise identification of error locations and no false positives in error-free sentences. Moreover, we introduce Precision Alignment Ratio (PAR)–the ratio of fine-grained precision to category-only precision. While standard evaluation only verifies error types, fine-grained testing demands accurate localization. Higher PAR values indicate genuine error understanding rather than random guessing. As Table 4 shows, SIS-Fact-Evaluator and GPT-4o achieve the highest PAR scores, with SIS-Fact-Evaluator demonstrating 30% superior fine-grained precision. This confirms SIS-Fact-Evaluator's performance stems from authentic comprehension rather than coincidental accuracy.

5.4 Ablation Study

Our dataset and model outputs consist of three key components: detailed error reasoning, grounding sentence extraction, and a sentence-by-sentence summary analysis. Table 5 presents an ablation study on these components. The results stress each component's importance in our model design. Our full model (SIS-Fact-Evaluator-RGS), which integrates error reasoning (R), grounding sentence extraction (G), and sentence-by-sentence analysis (S), achieves the best performance. Meanwhile, removing any single component results in a notable performance drop, indicating each plays a significant role.

6 Conclusion

In this paper, we propose a new framework termed SIS-Fact, which integrates error typologies, synthetic data generation pipelines, and fine-grained annotations for comprehensive factuality evaluation. To be specific, we first develop diverse methods to synthesize six error types in grounded generation, including both intrinsic and extrinsic errors. In this way, we construct SIS-Fact Dataset, a novel dataset characterized by longer summaries and challenging document contexts. Building upon SIS-Fact Dataset, we develop SIS-Fact-Evaluator, an advanced factuality evaluation model capable of fine-grained analysis. Also, we conduct extensive experiments to verify the superiority of SIS-Fact Dataset and SIS-Fact-Evaluator.

Limitations

Our pipeline's effectiveness is constrained by inherent limitations in LLM capabilities. While we employ sentence-level verification, the models still generate document-unsupported summaries. Additionally, they struggle to differentiate between factual incompleteness and legitimate information simplification, particularly affecting error construction quality for more complex cases.

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*.

AI@Meta. 2024. Llama 3 model card.

Anthropic. 2025. Claude 3.7 sonnet system card.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2023. Self-rag: Learning to retrieve, generate, and critique through self-reflection. In *The Twelfth International Conference on Learning Representations*.

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George Bm Van Den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, Diego De Las Casas, Aurelia Guy, Jacob Menick, Roman Ring, Tom Hennigan, Saffron Huang, Loren Maggiore, Chris Jones, Albin Cassirer, Andy Brock, Michela Paganini, Geoffrey Irving, Oriol Vinyals, Simon Osindero, Karen Simonyan, Jack Rae, Erich Elsen, and Laurent Sifre. 2022. Improving language models by retrieving from trillions of tokens. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 2206–2240. PMLR.

Shuyang Cao and Lu Wang. 2021. CLIFF: Contrastive learning for improving faithfulness and factuality in abstractive summarization. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6633–6649, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. 2022. Knowledge neurons in pretrained transformers. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8493–8502, Dublin, Ireland. Association for Computational Linguistics.

DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Haowei Zhang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Li, Hui Qu, J. L. Cai, Jian Liang, Jianzhong Guo, Jiaqi Ni, Jiashi Li, Jiawei Wang, Jin Chen, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, Junxiao Song, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Lei Xu, Leyi Xia, Liang Zhao, Litong Wang, Liyue Zhang, Meng Li, Miaojun Wang, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingming Li, Ning Tian, Panpan Huang, Peiyi Wang, Peng Zhang, Qiancheng Wang, Qihao Zhu, Qinyu Chen, Qiushi Du, R. J. Chen, R. L. Jin, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, Runxin Xu, Ruoyu Zhang, Ruyi Chen, S. S. Li, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shaoqing Wu, Shengfeng Ye, Shengfeng Ye, Shirong Ma, Shiyu Wang, Shuang Zhou, Shuiping Yu, Shunfeng Zhou, Shuting Pan, T. Wang, Tao Yun, Tian Pei, Tianyu Sun, W. L. Xiao, Wangding Zeng, Wanjia Zhao, Wei An, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, X. Q. Li, Xiangyue Jin, Xianzu Wang, Xiao Bi, Xiaodong Liu, Xiaohan Wang, Xiaojin Shen, Xiaokang Chen, Xiaokang Zhang, Xiaosha Chen, Xiaotao Nie, Xiaowen Sun, Xiaoxiang Wang, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xingkai Yu, Xinnan Song, Xinxia Shan, Xinyi Zhou, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, Y. K. Li, Y. Q. Wang, Y. X. Wei, Y. X. Zhu, Yang Zhang, Yanhong Xu, Yanhong Xu, Yanping Huang, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Li, Yaohui Wang, Yi Yu, Yi Zheng, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Ying Tang, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yu Wu, Yuan Ou, Yuchen Zhu, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yukun Zha, Yunfan Xiong, Yunxian Ma, Yuting Yan, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Z. F. Wu, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhen Huang, Zhen Zhang, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhibin Gou, Zhicheng Ma, Zhigang Yan, Zhihong Shao, Zhipeng Xu, Zhiyu Wu, Zhongyu Zhang, Zhuoshu Li, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Ziyi Gao, and Zizheng Pan. 2025. Deepseek-v3 technical report. *Preprint*, arXiv:2412.19437.

Matthew R. DeVerna, Harry Yaojun Yan, Kai-Cheng Yang, and Filippo Menczer. 2024. Fact-checking information from large language models can decrease headline discernment. *Proceedings of the National Academy of Sciences*, 121(50):e2322823121.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. 2024. Improving factuality and reasoning in language models through multiagent debate. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org.

Esin Durmus, He He, and Mona Diab. 2020. FEQA: A

question answering evaluation framework for faithfulness assessment in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5055–5070, Online. Association for Computational Linguistics.

Alexander Fabbri, Chien-Sheng Wu, Wenhao Liu, and Caiming Xiong. 2022. QAFactEval: Improved QA-based factual consistency evaluation for summarization. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2587–2601, Seattle, United States. Association for Computational Linguistics.

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.

Tanya Goyal and Greg Durrett. 2021a. Annotating and modeling fine-grained factuality in summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1449–1462, Online. Association for Computational Linguistics.

Tanya Goyal and Greg Durrett. 2021b. Annotating and modeling fine-grained factuality in summarization. *Preprint*, arXiv:2104.04302.

Anushka Gupta, Diksha Chugh, Anjum, and Rahul Katarya. 2022. Automated news summarization using transformers. In *Sustainable advanced computing: select proceedings of ICSAC 2021*, pages 249–259. Springer.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, et al. 2025. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*, 43(2):1–55.

Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. 2024. Openai o1 system card. *arXiv preprint arXiv:2412.16720*.

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.

Wojciech Kryściński, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Evaluating the factual consistency of abstractive text summarization. *Preprint*, arXiv:1910.12840.

Nayeon Lee, Wei Ping, Peng Xu, Mostofa Patwary, Pascale Fung, Mohammad Shoeybi, and Bryan Catanzaro. 2022. Factuality enhanced language models for open-ended text generation. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA. Curran Associates Inc.

Alisa Liu, Zhaofeng Wu, Julian Michael, Alane Suhr, Peter West, Alexander Koller, Swabha Swayamdipta, Noah Smith, and Yejin Choi. 2023a. We're afraid language models aren't modeling ambiguity. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 790–807, Singapore. Association for Computational Linguistics.

Nelson Liu, Tianyi Zhang, and Percy Liang. 2023b. Evaluating verifiability in generative search engines. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 7001–7025, Singapore. Association for Computational Linguistics.

Yang Liu and Mirella Lapata. 2019. Text summarization with pretrained encoders. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3730–3740, Hong Kong, China. Association for Computational Linguistics.

Pan Lu, Swaroop Mishra, Tony Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. 2022. Learn to explain: multimodal reasoning via thought chains for science question answering. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA. Curran Associates Inc.

Hongyin Luo, Tianhua Zhang, Yung-Sung Chuang, Yuan Gong, Yoon Kim, Xixin Wu, Helen Meng, and James Glass. 2023a. Search augmented instruction learning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 3717–3729, Singapore. Association for Computational Linguistics.

Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. 2023b. An empirical study of catastrophic forgetting in large language models during continual fine-tuning. *arXiv preprint arXiv:2308.08747*.

Chaitanya Malaviya, Subin Lee, Sihao Chen, Elizabeth Sieber, Mark Yatskar, and Dan Roth. 2024. ExpertQA: Expert-curated questions and attributed answers. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 3025–3045, Mexico City, Mexico. Association for Computational Linguistics.

Luca Massarelli, Fabio Petroni, Aleksandra Piktus, Myle Ott, Tim Rocktäschel, Vassilis Plachouras, Fabrizio Silvestri, and Sebastian Riedel. 2020. How decoding strategies affect the verifiability of generated text. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 223–235, Online. Association for Computational Linguistics.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On faithfulness and factuality in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1906–1919, Online. Association for Computational Linguistics.

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 *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12076–12100, Singapore. Association for Computational Linguistics.

Fedor Moiseev, Zhe Dong, Enrique Alfonseca, and Martin Jaggi. 2022. SKILL: Structured knowledge infusion for large language models. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1581–1588, Seattle, United States. Association for Computational Linguistics.

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. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807, Brussels, Belgium. Association for Computational Linguistics.

Cheng Niu, Yuanhao Wu, Juno Zhu, Siliang Xu, KaShun Shum, Randy Zhong, Juntong Song, and Tong Zhang. 2024. RAGTruth: A hallucination corpus for developing trustworthy retrieval-augmented language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10862–10878, Bangkok, Thailand. Association for Computational Linguistics.

Artidoro Pagnoni, Vidhisha Balachandran, and Yulia Tsvetkov. 2021a. Understanding factuality in abstractive summarization with frank: A benchmark for factuality metrics. *Preprint*, arXiv:2104.13346.

Artidoro Pagnoni, Vidhisha Balachandran, and Yulia Tsvetkov. 2021b. Understanding factuality in abstractive summarization with FRANK: A benchmark for factuality metrics. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4812–4829, Online. Association for Computational Linguistics.

Leonardo F. R. Ribeiro, Mengwen Liu, Iryna Gurevych, Markus Dreyer, and Mohit Bansal. 2022. FactGraph: Evaluating factuality in summarization with semantic graph representations. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3238–3253, Seattle, United States. Association for Computational Linguistics.

Liyan Tang, Tanya Goyal, Alex Fabbri, Philippe Laban, Jiacheng Xu, Semih Yavuz, Wojciech Kryscinski, Justin Rousseau, and Greg Durrett. 2023. marization: Errors, summarizers, datasets, error detectors. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11626–11644, Toronto, Canada. Association for Computational Linguistics.

Liyan Tang, Philippe Laban, and Greg Durrett. 2024a. Minicheck: Efficient fact-checking of llms on grounding documents. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 8818–8847.

Liyan Tang, Igor Shalyminov, Amy Wong, Jon Burnsky, Jake Vincent, Yu'an Yang, Siffi Singh, Song Feng, Hwanjun Song, Hang Su, Lijia Sun, Yi Zhang, Saab Mansour, and Kathleen McKeown. 2024b. TofuEval: Evaluating hallucinations of LLMs on topic-focused dialogue summarization. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4455–4480, Mexico City, Mexico. Association for Computational Linguistics.

Qwen Team. 2024. Qwen2.5: A party of foundation models.

James Thorne and Andreas Vlachos. 2018. Automated fact checking: Task formulations, methods and future directions. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3346–3359, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Alex Wang, Kyunghyun Cho, and Mike Lewis. 2020. Asking and answering questions to evaluate the factual consistency of summaries. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5008–5020, Online. Association for Computational Linguistics.

Minzheng Wang, Longze Chen, Fu Cheng, Shengyi Liao, Xinghua Zhang, Bingli Wu, Haiyang Yu, Nan Xu, Lei Zhang, Run Luo, Yunshui Li, Min Yang, Fei Huang, and Yongbin Li. 2024. Leave no document behind: Benchmarking long-context LLMs with extended multi-doc QA. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 5627–5646, Miami, Florida, USA. Association for Computational Linguistics.

Yujing Wang, Hainan Zhang, Liang Pang, Binghui Guo, Hongwei Zheng, and Zhiming Zheng. 2025.

Maferw: Query rewriting with multi-aspect feedbacks for retrieval-augmented large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 25434–25442.

Yumo Xu and Mirella Lapata. 2019. Weakly supervised domain detection. *Transactions of the Association for Computational Linguistics*, 7:581–596.

Yuheng Zha, Yichi Yang, Ruichen Li, and Zhiting Hu. 2023. AlignScore: Evaluating factual consistency with a unified alignment function. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11328–11348, Toronto, Canada. Association for Computational Linguistics.

Haopeng Zhang, Philip S. Yu, and Jiawei Zhang. 2024a. A systematic survey of text summarization: From statistical methods to large language models. *Preprint*, arXiv:2406.11289.

Huajian Zhang, Yumo Xu, and Laura Perez-Beltrachini. 2024b. Fine-grained natural language inference based faithfulness evaluation for diverse summarisation tasks. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1701–1722, St. Julian's, Malta. Association for Computational Linguistics.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. 2020. Pegasus: pre-training with extracted gap-sentences for abstractive summarization. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org.

Yang Zhong and Diane Litman. 2025. Discourse-driven evaluation: Unveiling factual inconsistency in long document summarization. *arXiv preprint arXiv:2502.06185*.

Mengna Zhu, Kaisheng Zeng, Mao Wang, Kaiming Xiao, Lei Hou, Hongbin Huang, and Juanzi Li. 2025. Eventsum: A large-scale event-centric summarization dataset for chinese multi-news documents. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 26138–26147.

A Construction Details for SIS-Fact Dataset

A.1 Detailed Categorization and Error construction Examples

Full catagorization and examples for each error construction methods is shown in Table 6.

A.2 Document Dataset Selection

Due to the limitations of existing summary datasets in terms of length (news datasets being relatively short, and academic paper dataset being excessively long), combined with challenges that some automatically extracted summaries lack complete support from the document, we opted to construct our dataset beginning with the document itself, which also streamlines the extraction of reference information and proves the broad usability of our method.

Initially, we randomly selected documents from the BBC News Summary Dataset (Gupta et al., 2022) and the DetNet Wikipedia Dataset (Xu and Lapata, 2019). These datasets offer a diverse range of documents covering news and encyclopedic content, classified by domain, with a varied length distribution. We ensured diversity by choosing documents from different domains and lengths.

The BBC News Summary Dataset (Gupta et al., 2022) consists of extractive summaries, so we did not use the original summaries. This data set categorizes news articles into five distinct categories: business, entertainment, politics, sports, and technology. The DetNet Wikipedia Dataset (Xu and Lapata, 2019) is designed for domain detection, with Wikipedia data labeled for seven domains: "Business and Commerce" (BUS), "Government and Politics" (GOV), "Physical and Mental Health" (HEA), "Law and Order" (LAW), "Lifestyle" (LIF), "Military" (MIL), and "General Purpose" (GEN). For the BBC News Summary Dataset, we selected 150 documents from each category. For the DetNet Wikipedia Dataset, we extracted 100 documents from each domain. The document length in both datasets highly varies. To ensure the models have a certain degree of robustness, but also efficient while training, we filtered documents to have lengths within the range of 300 to 1000 words.

A.3 Summary and Reference Generation

In order to ensure robustness and unbiased results, multiple models were used at each step of the generation of the Grounded Summary Dataset instead of relying on a single model. We prompted the language models (LLMs) to control the summary length within the range of $[100, \min(doc\_len/3 + 10, 200)]$ words. To address potential biases where a single model might favor its own generated text, and to avoid issues where training exclusively with one model's outputs might cause out-of-domain problems for texts generated by other models, we utilized two different sets of LLMs at each step of our pipeline. In addition, the models used for generation and evaluation were different. Usage of LLMs are outlined in Table 7.

Categorization			Method	Example
Intrinsic Errors	Semantic Frame Errors	Predicate Error (PredE)	Swapping Relation Masking	Its impact on theaters is now emerging. Its absence from theaters is now receding.
			Modifying Predictions	Despite it all, 2023 should still reach the $9 billion in domestic gross hoped for this year. Despite it all, 2023 reached the $9 billion in domestic gross hoped for this year.
		Entity Error (EntE)	Swapping Entities	Her last stage role was in My Fair Lady. Her last stage role was in Bless This House.
			Compressing Words	In France and Italy, he wrote his last work. In France, he wrote his last work.
		Circumstance Error (CircE)	Swapping Circumstances	The shooting left 10 students and 2 teachers dead. The shooting left 2 students and 10 teachers dead.
	Discourse Errors	Co-reference Error (CorefE)	Swapping Pronouns	Gonzales was also indicted. She was also indicted.
			Merging Sentences	The charges were first reported by the San Antonio Express-News. District Attorney Christina Mitchell did not return requests for comment. The charges were first reported by the San Antonio Express-News, who did not return requests for comment.
		Discourse Link Error (LinkE)	Reverse Logical Relationship	The six-month Hollywood labor disruption has finally ended. Immediately following the settlement, Disney announced delays in its upcoming release schedule. After the announcement of the delays in Disney's upcoming release schedule, the six-month Hollywood labor disruption has finally ended.
Extrinsic Errors (OutE)			Introducing Extrinsic Information	Robert escaped to Visegrád disguised as a civilian, aided by Nicholas, son of Radoslav, who defended him against five attackers. Robert escaped to Visegrád disguised as a civilian, aided by Nicholas, son of Radoslav, a renowned swordsman known for his exceptional skill in battle, who defended him against five attackers.

Table 6: Categories of the error data. The blue part is the selected text for modification, and the red part is the modified text.

Task	LLM Set 1	LLM Set 2
Summarization	GPT-4o	DeepSeek-R1
Reference extraction	Claude-3.7-Sonnet	GPT-4o
Support determination	(Claude-3.7-Sonnet, Qwen-2.5, Gemini-1.5)	(GPT-4o, Qwen-2.5, Gemini-2.0)
Rewriting	Claude-3.7-Sonnet	GPT-4o
Error data construction	GPT-4o	Claude-3.7-Sonnet

Table 7: Usage of LLMs in dataset construction

B Data Structure Example

See Figure 4

C Training Details of SIS-Fact-Evaluator

We fine-tune Llama-3-8B-Base for 3 epochs on the train set of SIS-Fact Dataset, using a batch size of 32 and the Adam optimizer. The learning rate follows a cosine-decay from $1e-5$ to $1e-6$, and we set the warm-up fraction to 0.1.

D Prompts

Figure 5, Figure 6, Figure 7 and Figure 8 are the prompts in the SIS-Fact pipeline. Figure 9 are the prompt used for the LLM baselines in our experiment.

E Human Evaluation

Label	Proportion(%)
No Problem	78
Flaws in Summary	11
Incomplete Grounding	6
False Negative Error	3
Wrong Error Type	1
Wrong Error Reasoning	1

Table 8: Human evaluation results on a sample of 100 instances from our dataset.

```
[
    {
        "summary sentence": "The column in Piazza Santa Maria Maggiore in Rome, crowned with a statue of the Virgin in 1614, set a precedent for many European columns.",
        "related sentence(s) from the document": [
            "The column in Piazza Santa Maria Maggiore in Rome was one of the first.",
            "Within decades it served as a model for many columns in Italy and other European countries."
        ],
        "supported or not": "YES",
        "reason": "The summary sentence faithfully reflects the related sentences.",
        "error type": "No Error"
        },
        {
        "summary sentence": "The first Marian column north of the Alps was Munich's Mariensäule in 1714, inspiring similar structures in Prague and Vienna.",
        "related sentence(s) from the document": [
            "The first column of this type north of the Alps was the Mariensäule built in Munich in 1638 to celebrate the sparing of the city from both the invading Swedish army and the plague.",
            "It inspired for example Marian columns in Prague and Vienna, but many others also followed very quickly."
        ],
        "supported or not": "NO",
        "reason": "This sentence is not supported by the related sentence(s).\n- Location: '1714'.\n- Explanation: The year of the construction of Munich's Mariensäule was changed from 1638 to 1714, falsely altering the historical timeline.\n- Correction: The first Marian column north of the Alps was Munich's Mariensäule in 1638, inspiring similar structures in Prague and Vienna.",
        "error type": "Circumstance Error"
    },
    {
        "summary sentence": "The Prague column, built post-Thirty Years' War, was destroyed in 1918 due to its association with Habsburg rule.",
        "related sentence(s) from the document": [
            "The Prague column was built in Old Town Square (Staroměstské náměstí) shortly after the Thirty Years' War in thanksgiving to the Virgin Mary Immaculate for helping in the fight with the Swedes.",
            "Unfortunately, many Czechs later connected its placement and erection with the hegemony of the Habsburgs in their country, and after declaring the independence of Czechoslovakia in 1918 a crowd of people pulled this old monument down and destroyed it in an excess of revolutionary fervor."
        ],
        "supported or not": "YES",
        "reason": "All content in the summary sentence is accurately derived from the related sentences.",
        "error type": "No Error"
    }
]
```


Figure 4: An example of the data in SIS-Fact Dataset, the data is truncated due to space limitations.

I'll provide you with a document. Your task is to write a short summary for this document according to the following requirements:
1. The length of the summary should be within <WORD_CONSTRAIN> words.
2. Every sentence in the summary should be directly supported by the content of the document.
3. For each event, make sure every important entity such as person, location and time is kept in the summary, especially entities that occurs in parallel.
4. When doing simplification, make sure each complex event or idea remains true to the original meaning. Avoid over-simplification that leads to in-consistency with the origin document.

Document:<Document>

Directly output the summary without any extra words.

Figure 5: Prompt for writing summaries.

Here is a document with a corresponding summary. Your task is to analyze the summary sentence by sentence. For each sentence in the summary, provide the exact sentences from the document that supports the summary sentence. If the summary comes from multiple sentences, report all sentences.
You should ensure that sentences are considered as individual units based on punctuation. Specifically:
- Treat each sentence as ending at a period (.), question mark (?), or exclamation mark (!), even if multiple sentences are enclosed within quotation marks.
- Do not truncate any sentence. If a portion of a sentence is extracted (e.g., ending at a comma, semicolon, or any punctuation other than ., ?, !), you must include the rest of the sentence so that the entire sentence is fully reproduced.

You should only respond in format as described below. Do not return anything else. START YOUR RESPONSE WITH '['
Return the result as a Python list of dictionaries, where each dictionary has the following keys:
"summary sentence": The sentence from the summary.
"sentences from the document": A Python list of the exact supporting sentences from the document. Ensure that the sentences are in the same order as they appear in the original document. Each sentence should be reported fully, without any omission.

Figure 6: Prompt for locating the grounding sentences.

Here are some pieces from the SOURCE_DESCRIPTION, and a summary sentence of it. Your task is to:
1. Compare the summary with the document and determine whether if the summary is fully supported by the content in the document. Specifically, verify if all key points made in the summary are traceable back to the sentence in the document. State whether the summary is fully supported with 'YES' or 'NO'.
2. If the answer is 'NO', revise the summary to align it fully with the document. Make sure the fluency and grammar correctness of the revised sentence, and ensure it accurately reflects the information.

You should only respond in format as described below. Do not return anything else. START YOUR RESPONSE WITH '['
Return the result as a Python list of dictionaries, where each dictionary has the following keys:
"summary sentence": The sentence from the summary.
"sentences from the document": A Python list of the exact supporting sentences from the document. Ensure that the sentences are in the same order as they appear in the original document. Each sentence should be reported fully, without any omission.

Figure 7: Prompt for determining whether a summary sentence is sufficiently supported by its grounding sentences. The purple part is only used in the LLM for re-writing.

Here is a document with a summary. Please create a fake summary based on the origin summary by the following steps:
Instruction:
1. Analyze the document and identify sentences that contain future predictions (e.g., those using modal verbs like 'will,' 'might,' or phrases like 'predict,' 'suppose').
2. Select a sentence where the prediction is the main clause, not just a subordinate clause, and modifying the prediction into a factual statement will result in the most significant change in meaning. You can modify this sentence so that the prediction is completely transformed into a factual statement about an event that has already occurred. Ensure the modified sentence is grammatically correct and fully removes any speculative language.
3. Based on the changed sentence, modify some part of the summary to include the fake information in the changed sentence, so the summary cannot be fully supported by the origin document.
Make sure the new summary should not be fully supported by the document, and not change any other part in the summary besides those associated with the modification.

You should only respond in format as described below. Do not return anything else. START YOUR RESPONSE WITH ‘{{‘.
Return the result as a Python dictionary with the following keys:
Format:
- "original text in summary": The original sentence containing the prediction from the document.
- "chosen element": The chosen prediction or future-oriented statement in the original text.
- "modification explanation": Description of the modification.
- "modified element": The new factual statement replacing the prediction.
- "modified text": The sentence after the modification, now a factual statement.
- "explanation": A clear explanation of how the meaning of the original text has been altered.
- "full text of modified summary": The full text of the modified summary.
- "wrong information": Point out the specific wrong information introduced in the summary after the modification.
Replace any line breaks in the values with '\n' so that the dictionary can be parsed using eval().

Figure 8: Prompt for generating SIS-Fact Dataset. The colored part is construction-method specific.

Task:
Evaluate the given summary by comparing it with the original document and identify any errors. These errors may include incorrect information, over-simplifications, misrepresentations, or other discrepancies. The possible types of errors to consider are as follows: Predicate Error, Entity Error, Circumstance Error, Co-reference Error, Discourse Link Error, Extrinsic Error.

Instructions:
You are provided with the full text of the original document and a summary that may contain errors. You should analyze the summary sentence by sentence and returning the results in the following Python list format:
Each item in the list should correspond to a summary sentence and be represented as a Python dictionary with the following keys:
- "summary sentence": The summary sentence.
- "related sentence(s) from the document": A list of sentences from the original document that support the summary sentence, if the summary sentence is fully supported. If the summary sentence contains an error, list the sentences needed to point out the error.
- "supported or not": "YES" if the summary sentence is fully supported by the related sentences, otherwise "NO".
- "reason": A brief analysis explaining whether the summary sentence is supported or not. If the summary sentence is not supported, specify where the error occurs, explain the incorrect information conveyed in the summary, and provide a corrected version of the sentence.
- "error type": The type of error found (choose from the types listed above), or "No Error" if the sentence is fully supported.

Document: <Document>
Summary: <Summary>

Figure 9: Prompt for evaluating LLM baseline.