Improving Faithfulness of Large Language Models in Summarization via Sliding Generation and Self-Consistency

Taiji Li, Zhi Li, Yin Zhang
College of Computer Science and Technology, Zhejiang University, China
{litaiji, zhili, zhangyin98}@zju.edu.cn

Abstract

Despite large language models (LLMs) have demonstrated impressive performance in various tasks, they are still suffering from the factual inconsistency problem called hallucinations. For instance, LLMs occasionally generate content that diverges from source article, and prefer to extract information that appears at the beginning and end of the context, especially in long document summarization. Inspired by these findings, we propose to improve the faithfulness of LLMs in summarization by impelling them to process the entire article more fairly and faithfully. We present a novel summary generation strategy, namely **SliSum**, which exploits the ideas of sliding windows and self-consistency. Specifically, SliSum divides the source article into overlapping windows, and utilizes LLM to generate local summaries for the content in the windows. Finally, SliSum aggregates all local summaries using clustering and majority voting algorithm to produce more faithful summary of entire article. Extensive experiments demonstrate that SliSum significantly improves the faithfulness of diverse LLMs including LLaMA-2, Claude-2 and GPT-3.5 in both short and long text summarization, while maintaining their fluency and informativeness and without additional fine-tuning and resources. We further conduct qualitative and quantitative studies to investigate why SliSum works and impacts of hyperparameters in SliSum on performance.

Keywords: Large Language Model, Summarization, Faithfulness

1. Introduction

Abstractive summarization aims to generate summaries that are fluent, informative, and faithful to the source articles. Benefiting from the popularity and development of large language models (LLMs), abstractive summarization has achieved remarkable progress in fluency and coherence (Zhang et al., 2023b; Goyal et al., 2023; Zhang et al., 2023a). However, LLMs have propensity to generate content that contradicts or is not present in the source article (Tam et al., 2023; Maynez et al., 2020; Lin et al., 2022; Li et al., 2023a), which is commonly referred to as hallucination. Alleviating the hallucination of LLMs is a critical challenge for their reliability in real-world applications.

Many works (Zhang et al., 2023c; Zheng et al., 2023) explore the mechanisms why LLMs exhibit hallucinations. Liu et al. (2023a) observe that the performance of LLMs significantly decreases as the length of input contexts increases, resulting in the hallucination phenomenon of LLMs being particularly serious in long document summarization. Furthermore, LLMs are sensitive to the order of context and are more likely to select information presented first or last, even for short contexts (Xie et al., 2024; Wang et al., 2023a). That is, the summaries generated by LLMs contain more content that occurs at the beginning and end of the source article, which has a detrimental effect on the summary quality of the entire article.

Article: Albert Einstein was a German-born theoretical physicist, widely held to be one of the greatest and most influential scientists of all time. Best known for developing the theory of relativity, he also made important contributions to quantum mechanics. ... He received the 1921 Nobel Prize in Physics "for his services to theoretical physics, and especially for his discovery of the law of the photoelectric effect", a pivotal step in the development of quantum theory. His work is also known for its influence on the philosophy of science. ... His mass–energy equivalence formula $E = mc^2$, which arises from relativity theory, has been called "the world's most famous equation". His intellectual achievements and originality have made the word Einstein broadly synonymous with genius. ... For much of the last phase of his academic life, Einstein fought a long rearguard action against quantum theory's introduction of fundamental randomness into science's picture of the world, objecting that "God does not play dice". ...
Local Summaries
Paragraph 1 and 2: Einstein is one of the greatest and most influential theoretical physicist and won the 1921 Nobel Prize in Physics for philosophy of science.
Paragraph 2 and 3: Einstein received the 1921 Nobel Prize in Physics for his discovery of the law of the photoelectric effect. Einstein is synonymous with genius.
Paragraph 3 and 4: The mass-energy equivalence formula is the most famous equation. Einstein objected to the randomness of quantum theory in his academic life.

Table 1: The self-contradiction problem between local summaries. The red sentences are contradictory statements about the reason why Einstein won the Nobel Prize in Physics. The purple sentences are different summaries of the third paragraph.

Recent works attempt to leverage post-processing models and the CoT (Chain of Thought) technique to improve factual consistency of summarization. Some works (Liu et al., 2023b; Xiao et al., 2024) employ LLMs as critic and editor models, the critic model generates editing instructions for the initial summary, and the editor model corrects factual errors by following the instructions. Wang et al.

[*] Corresponding Author: Yin Zhang.

(2023c) propose a CoT-based method called Sum-CoT to elicit LLMs to generate summaries step by step. However, they do not mitigate position bias and performance degradation in long context scenarios.

In this paper, we propose **SliSum**, a novel summary generation strategy that improves the faithfulness of LLMs in both short and long text summarization by sliding generation and self-consistency. As shown in Figure 1, our approach consists of the following three steps: (1) **Sliding Generation**: SliSum uses LLM to generate local summaries for overlapping windows, and local summaries contain multiple statements about the same event that may have contradictions (see Table 1). (2) **Filtration**: According to the principle of self-consistency, the statements generated more times by LLMs are more faithful and important to the source article (Wang et al., 2023b; Manakul et al., 2023). Therefore, we obtain statements about the same event by lexical clustering over sentences of all local summaries and then filter out small sentence clusters and outliers so that the global summary contains only relatively more important information of the source article. (3) **Aggregation**: SliSum employs LLM to detect whether there are contradictions in statements about the same event and classify them into different categories based on their semantics. Finally, SliSum uses a majority voting algorithm to select the statements with the most proportion, and concatenates them to form a more faithful summary of the entire article. SliSum brings three major benefits: (1) Sliding window provides LLMs with more diverse and adequate information by splitting articles in an overlapping manner; (2) The filtration and aggregation based on self-consistency ingeniously mitigate the self-contradiction problem and further leverage the potential of LLMs to improve the faithfulness of them; (3) The combination of sliding windows and self-consistency impels LLMs to process the entire article more fairly and faithfully. Therefore, SliSum is capable of improving the faithfulness of LLMs in summarization without external resources and additional fine-tuning.

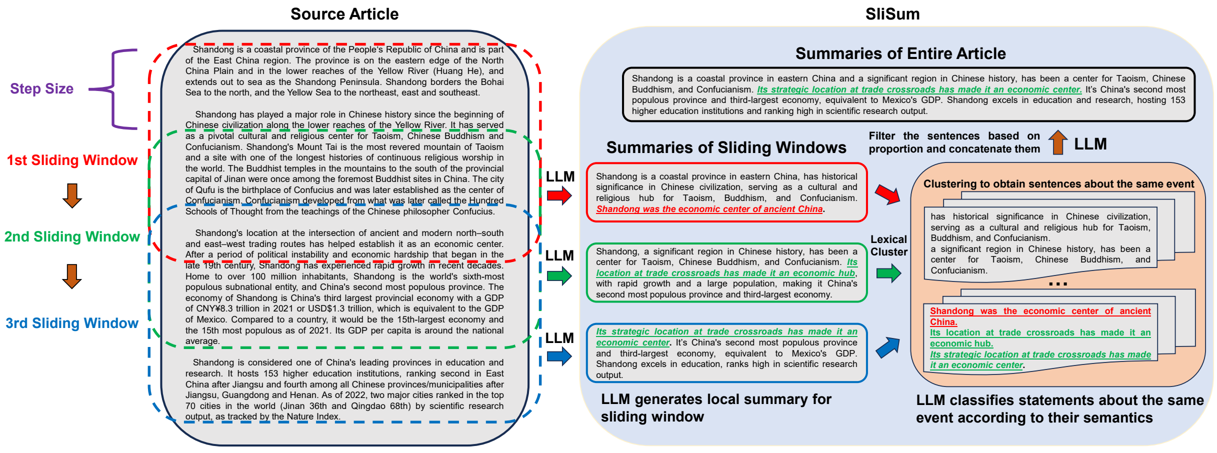
We evaluate the effectiveness of SliSum applied to three advanced LLMs, LLaMA-2-13B (Touvron et al., 2023), Claude-2 (Anthropic, 2023) and GPT-3.5 (OpenAI, 2023), on four popular summarization datasets. Extensive experiments have shown that SliSum significantly improves the faithfulness of three LLMs on the short news datasets CNN/DM and XSum, and the long scientific papers datasets arXiv and PubMed, respectively, while without sacrificing their fluency and informativeness. Besides, we conduct ablation studies to further investigate why SliSum works. We also perform quantitative investigations on hyperparameters in SliSum. Our contributions are summarized as follows.

- We propose a novel summary generation architecture, **SliSum**, that improves faithfulness of LLMs by sliding windows and self-consistency without additional resources and fine-tuning. To the best of our knowledge, we are the first to apply overlapping context windows into LLMs for abstractive summarization.
- We demonstrate that SliSum uniformly improves factual consistency of summaries generated by diverse LLMs while maintaining their fluency and informativeness, more importantly, SliSum is applicable to text of various lengths and styles.
- We conduct extensive qualitative and quantitative experiments to validate the effectiveness of sliding generation and aggregation based on self-consistency and impacts of hyperparameters in SliSum on performance.

2. Related Works

Factual Consistency of Summarization The factual consistency in abstractive summarization has received increasing attention recently. Existing work has proposed various methods to improve the factual consistency, such as contrastive learning (Wan and Bansal, 2022; Xie et al., 2023), adversarial learning (Wang et al., 2022; Wu et al., 2022), textual entailment (Zhang et al., 2022b; Roit et al., 2023) and post-editing (Fabbri et al., 2022; Balachandran et al., 2022). However, these methods can not be directly applied in long document summarization due to the difficulty of modeling long texts accurately. Although recent studies have addressed this problem by leveraging Graph Neural Networks (Zhang et al., 2022a; Doan et al., 2022; Phan et al., 2022; Xie et al., 2022), reinforcement learning (Gu et al., 2022) and structure information (Cao and Wang, 2022; Cho et al., 2022; Pu et al., 2023), they do not improve the faithfulness of long document summarization systems. There are currently few works that focus on improving factual consistency of long document summarization. In contrast, we propose a unified architecture that consistently improves factual consistency of both short and long text summarization.

Mitigation of LLM Hallucination Recent studies have made several attempts to mitigate the hallucination of LLMs, including retrieval-augmented generation (Peng et al., 2023; Ram et al., 2023; Kang et al., 2023; Xu et al., 2024), post-processing models (Chen et al., 2023; Gou et al., 2024; Huang et al., 2023), prompt engineering (Xue et al., 2023; Shi et al., 2023; Luo et al., 2023; Dhuliawala et al., 2023) and self-supervised learning (Gekhman et al., 2023; Manakul et al., 2023; Du et al., 2023). However, most of these works require training additional


Source Article
SliSum
Step Size
1st Sliding Window
2nd Sliding Window
3rd Sliding Window
LLM
LLM
LLM
Summaries of Entire Article
Summaries of Sliding Windows
Filter the sentences based on proportion and concatenate them
LLM
Lexical Cluster
Clustering to obtain sentences about the same event
...
LLM generates local summary for sliding window
LLM classifies statements about the same event according to their semantics

Shandong is a coastal province of the People's Republic of China and is part of the East China region. The province is on the eastern edge of the North China Plain and in the lower reaches of the Yellow River (Huang He), and extends out to sea as the Shandong Peninsula. Shandong borders the Bohai Sea to the north, and the Yellow Sea to the northeast, east and southeast.

Shandong has played a major role in Chinese history since the beginning of Chinese civilization along the lower reaches of the Yellow River. It has served as a pivotal cultural and religious center for Taoism, Chinese Buddhism and Confucianism. Shandong's Mount Tai is the most revered mountain of Taoism and a site with one of the longest histories of continuous religious worship in the world. The Buddhist temples in the mountains to the south of the provincial capital of Jinan were once among the foremost Buddhist sites in China. The city of Qufu is the birthplace of Confucius and was later established as the center of Confucianism. Confucianism developed from what was later called the Hundred Schools of Thought from the teachings of the Chinese philosopher Confucius.

Shandong's location at the intersection of ancient and modern north–south and east–west trading routes has helped establish it as an economic center. After a period of political instability and economic hardship that began in the late 19th century, Shandong has experienced rapid growth in recent decades. Home to over 100 million inhabitants, Shandong is the world's sixth-most populous subnational entity, and China's second most populous province. The economy of Shandong is China's third largest provincial economy with a GDP of CNY¥8.3 trillion in 2021 or USD$1.3 trillion, which is equivalent to the GDP of Mexico. Compared to a country, it would be the 15th-largest economy and the 15th most populous as of 2021. Its GDP per capita is around the national average.

Shandong is considered one of China's leading provinces in education and research. It hosts 153 higher education institutions, ranking second in East China after Jiangsu and fourth among all Chinese provinces/municipalities after Jiangsu, Guangdong and Henan. As of 2022, two major cities ranked in the top 70 cities in the world (Jinan 36th and Qingdao 68th) by scientific research output, as tracked by the Nature Index.

Shandong is a coastal province in eastern China and a significant region in Chinese history, has been a center for Taoism, Chinese Buddhism, and Confucianism. ***Its strategic location at trade crossroads has made it an economic center.*** It's China's second most populous province and third-largest economy, equivalent to Mexico's GDP. Shandong excels in education and research, hosting 153 higher education institutions and ranking high in scientific research output.

Shandong is a coastal province in eastern China, has historical significance in Chinese civilization, serving as a cultural and religious hub for Taoism, Buddhism, and Confucianism. ***Shandong was the economic center of ancient China***.

Shandong, a significant region in Chinese history, has been a center for Taoism, Chinese Buddhism, and Confucianism. ***Its location at trade crossroads has made it an economic hub***, with rapid growth and a large population, making it China's second most populous province and third-largest economy.

Its strategic location at trade crossroads has made it an economic center. It's China's second most populous province and third-largest economy, equivalent to Mexico's GDP. Shandong excels in education, ranks high in scientific research output.

has historical significance in Chinese civilization, serving as a cultural and religious hub for Taoism, Buddhism, and Confucianism.
a significant region in Chinese history, has been a center for Taoism, Chinese Buddhism, and Confucianism.

Shandong was the economic center of ancient China.
Its location at trade crossroads has made it an economic hub.
Its strategic location at trade crossroads has made it an economic center.

Figure 1: The pipeline and example of our proposed **SliSum** approach. In order to solve self-contradiction problem, SliSum take majority vote over sentences of each cluster base on their semantics and select the category with the most votes. For instance, the **green sentences** have the similar semantics and appear twice, while the **red sentence** with different semantics appear only once. Hence, the **second green sentence** is selected to be output to the final summary. In the implementation, SliSum processes the source article at the sentence level. For the simplicity of the illustration, the windows in the figure are represented by text lines.

models or external knowledge, or lack sufficient evaluation on long contexts tasks. For example, LLM-Augmenter (Peng et al., 2023) acquires evidence by retrieving external knowledge for the LLM to generate candidate responses grounded in evidence. Du et al. (2023) propose a post-processing method that improves factuality of language models using multiagent debate. SELF-FAMILIARITY (Luo et al., 2023) has revealed excellent performance in short context tasks, but its advancement has not been demonstrated on long-context datasets. Different from previous works, SliSum improves faithfulness of LLMs in long context scenarios without additional fine-tuning and external resources.

Long Context for LLMs With the application fields of LLMs continue to expand, boosting the performance of LLMs in long context scenarios has aroused a surge of interest. Han et al. (2023) propose a decoding method, LM-Infinite, to maintain fluency and generation quality of LLMs on long sequences, which can only be applied to LLMs using relative positional encodings, but our approach is applicable to diverse LLMs, even black-box models, because it does not modify the original structure and implementation of the models. Jiang et al. (2023) and Li et al. (2023b) compress long context by removing redundancy to enhance LLMs' perception of the key information, but there is no guarantee that compressed contexts contain faithful information to original contexts. Ratner et al. (2023) propose a parallel context windows method that LLMs process contexts located in these windows separately to mitigate the negative impact of long contexts on performance. However, parallel windows separate the semantic relationships within the article, which inevitably affects LLMs' understanding and extraction of global information. We are the first to introduce overlapping sliding windows into LLMs for abstractive summarization tasks. Unlike traditional parallel windows, sliding windows overlap each other so that each segment of the source article can be located at the beginning of a certain window, allowing them to overcome their preference for position and process the entire context more fairly and faithfully. In previous works, the sliding window method is only applied to extractive (Cui and Hu, 2021) and query-focused (Vig et al., 2022) summarization, because they do not solve the self-contradiction problem (Mündler et al., 2024; Liu et al., 2022) between local summaries. SliSum addresses this problem by combining sliding windows and self-consistency and achieves better performance than existing methods.

3. Approach

3.1. Sliding Generation

Sliding Window SliSum first divide the source article A into a list of sentences $[C_1, C_2, \cdots, C_n]$. The sliding window with predefined ***window size*** initially consists of several consecutive sentences $[C_1, \cdots, C_t]$ at the beginning of the article. The number of words in the window only needs to be approximately equal to the window size. Figure 1 shows that LLM generates a local summary for the content in the sliding window. SliSum restricts the attention of LLM within a window, alleviating the distraction issue caused by long contexts (Han et al., 2023; Tworkowski et al., 2023). Then the sliding window moves downward by a distance of ***step size*** and LLM generates a local summary for current window again. Similarly, the distance moved does not need to be strictly equal to step size. That is, the starting index of the window advances a few sentences. Importantly, SliSum processes the source article at the sentence level. Finally, LLM iteratively generates summaries for the content in the current window until the sliding window traverses the entire article. Given an article $A = [C_1, C_2, \cdots, C_n]$, the process of sliding generation can be described as follows:

$$\begin{aligned}
\mathrm{S}_1 &= SUM(\mathrm{W}_1) = SUM([C_{p_1}, \cdots, C_{q_1}]) \\
\mathrm{S}_2 &= SUM(\mathrm{W}_2) = SUM([C_{p_2}, \cdots, C_{q_2}]) \\
&\vdots \\
\mathrm{S_m} &= SUM(\mathrm{W_m}) = SUM([C_{p_m}, \cdots, C_{q_m}])
\end{aligned}$$

where $SUM(W_i)$ represents LLM summarizes the content in the window W_i, W_i consists of from p_i-th to q_i-th sentences. $p_1 = 1$, $q_m = n$, and

$$p_i \leq p_{i+1} \leq q_i, \quad i = 1, \cdots, m-1$$

Each text segment except the beginning and end can locate in different positions in the context window. SliSum balances the position distribution in the context window of text segments, eliminating the position preference of LLMs for source article.

Process Repeatedly Eventually SliSum will obtain a concatenation of summaries $[S_1, S_2, \cdots, S_m]$ by sliding generation. In the sliding window method, step size must be less than or equal to window size, otherwise there will be gaps between windows. Figure 1 shows that LLM processes the same part of the source article multiple times due to overlaps between windows. Generally, given window size L_w, step size L_s ($L_s \leq L_w$) and an article whose length L_a is far larger than the window size, i.e, $L_a \gg L_w$, The minimum number of times the middle part of the article is summarized by LLM:

$$K = \left\lfloor \frac{L_w}{L_s} \right\rfloor \tag{1}$$

It can be easily seen that if L_w is an integer multiple of L_s in sliding window method, LLM will read the middle part of the source article the same number of times, so we set $L_w = KL_s$ in SliSum. In particular, for several windows at the beginning and end, we repeatedly generate local summaries for them so that all parts of the source article are summarized K times.

3.2. Events Filtering

Lexical Clustering The event is usually composed of entity, behavior, reason, result and other elements, so statements about different events have their own characteristic words that can be used to distinguish them. We observe that the statements generated by LLMs for the same event are usually lexically similar. Consequently, we can obtain a set of sentences about the same event by lexical similarity clustering. As shown in Figure 1, SliSum divides all summaries into sentences, then uses DBSCAN (Ester et al., 1996) algorithm to cluster them based on lexical similarity. Specifically, we use ROUGE-1 (Lin, 2004) F1 score to define the distance between two sentences C_1 and C_2:

$$dist(C_1, C_2) = 1 - \mathcal{R}(C_1, C_2) \tag{2}$$

where $\mathcal{R}(\cdot)$ is ROUGE-1 F1 score, and is computed as follow:

$$\mathcal{R}(C_1, C_2) = \frac{2 * \text{number of overlap words}}{\text{length of } C_1 + \text{length of } C_2}$$

Obviously, $\mathcal{R}(C_1, C_2) \in [0, 1]$. There are two important parameters in the DBSCAN algorithm: distance threshold ε and minimum number of points (**MinPts**) in a cluster. Formally, let $\hat{C}$ represents a sentence, if exists a set of sentences (including $\hat{C}$ itself) $\mathbb{C} = \{C_1, C_2, \cdots, C_N\}$ and the number of sentences $|\mathbb{C}| \geq \text{MinPts}$, every sentence $C \in \mathbb{C}$ satisfies

$$dist(\hat{C}, C) \leq \varepsilon$$

$\mathbb{C}$ will be considered a cluster. A lower MinPts induces the algorithm to construct more clusters, while a higher MinPts will ensure more robust and consistent clusters.

Filtering Noise Intuitively, if an event appears in the local summary of a window, it means that the event is important to the content in the window. We notice that some sentences are only important in minority windows and should not appear in the overall summary. Furthermore, LLMs occasionally generate few hallucination statements that deviate from the source article. These statements tamper with information in the source article or add information that cannot be inferred from the source article, thus they generally have low lexical similarity to factually consistent statements. These undesirable statements are treated as outliers and noise in DBSCAN algorithm. Due to MinPts determines the minimum number of points in a cluster, so we can filter out unimportant and hallucination statements by using an appropriate MinPts.

As mentioned in Section 3.1, SliSum generate K summaries for the content of the source article by sliding generation, that is, sentences about the same event occur at most K times. Under normal circumstances, a cluster contains at most K sentences. In order to generate more concise and relative summaries, we select sentences that exist in as least MinPts local summaries, i.e, we only retains clusters whose size exceeds MinPts. Therefore, we can adjust MinPts in the range of $[1, K]$ to filter noise and hallucination as well as improve self-consistency.

3.3. Contradictions Detection and Sentences Aggregation

Sentences Selection As shown in Table 1 and Figure 1, there may be contradictory statements in sentences about the same event. Due to LLMs have ability to detect self-contradictory statements without relying on additional knowledge (Mündler et al., 2024), SliSum first manipulates the LLM used in the sliding generation to divide sentences into different categories according to their semantics. Generally, the more times a certain response is output by LLMs, the higher the probability that the response is desirable (Wang et al., 2023b; Manakul et al., 2023), since self-consistency reflects factual consistency. Consequently, SliSum can utilize majority voting algorithm to select sentences that are faithful to the source article.

After distinguishing sentences that are high lexically similar but semantically different, sentences of the same category state the same fact. SliSum selects the sentence with the largest proportion to output to the final summary. When there are clusters of sentences with the same proportion or when it is necessary to choose sentence from a certain cluster, the relatively last generated sentence is selected by SliSum. Because the last generated sentence is located at the beginning of the corresponding window, and LLMs have the highest factual consistency for the information occurs at the beginning of the input context (Ravaut et al., 2024; Chhabra et al., 2024). For example, an event has five related statements $[D_1, \cdots, D_5]$ generated sequentially, and LLM divides them into $[D_2]$, $[D_1, D_4]$, $[D_3, D_5]$, SliSum selects D_5 as the most faithful statement to form the summary of the entire article.

Sentences Integration SliSum finally integrates all selected sentences to output the final summary of the entire article. SliSum uses LLM to generate connectives and concatenate these sentences in order, so that the order of the final summary is consistent with the source article. We determine the order of the sentences according to the order of the events they describe. In the integration stage, SliSum do not need to change any information of the selected sentences.

Summary Generation: Generate local summary for content in the window.
Instruction: Summarize the above article.
Sentence Classification: Divide the statements about the same event.
Instruction: Classify the above statements into different categories. Statements of the same category describe the same facts, and statements of different categories have different semantics.
Sentences Integration: concatenate selected sentences in order.
Instruction: Generate connectives to concatenate sentences to form a fluent text. DO NOT change the original semantics.

Table 2: Prompts used in SliSum.

4. Experimental Setup

4.1. Datasets

We evaluate the performance of SliSum for short and long text summarization on four popular benchmark datasets: **CNN/DM** (Hermann et al., 2015) and **XSum** (Narayan et al., 2018) are two widely-used short news summarization datasets. **PubMed** and **arXiv** (Cohan et al., 2018) are two scientific paper datasets for long document summarization, which are much longer than the common news articles. PubMed contains academic papers from the biotechnology domain, while arXiv contains papers from different scientific domains. Limited by the maximum number of requests and time cost, we randomly select 100 articles from test set of each dataset to construct our test set, respectively.

4.2. Baselines

To validate the effectiveness of our proposed approach, we apply SliSum to three state-of-the-art instruction-tuned LLMs, including **LLaMA-2 13B Chat**[1], **Claude-2**[2] and **GPT-3.5-Turbo-0613**[3]. We also compare SliSum with recent faithfulness enhancement summarization models on CNN/DM and XSum, such as **CLIFF** (Cao and Wang, 2021) and **FES** (Chen et al., 2022). We select GPT-3.5 with **SumCoT** (Wang et al., 2023c) as LLM news summarization baseline. For arXiv and PubMed, we compare SliSum with two long document summarization baselines: **FactorSum** (Fonseca et al., 2022), a factorized energy-based abstractive model that improves the performance and applicability by separate budget decisions from selecting important content in the document, and **Lodoss** (Cho et al., 2022), an extractive architecture that learns robust

[1]https://ai.meta.com/resources/models-and-libraries/llama/
[2]https://www.anthropic.com/index/claude-2
[3]https://openai.com/blog/chatgpt

	R-1	R-2	R-L	BS	FC	SC	R-1	R-2	R-L	BS	FC	SC
Model	**CNN/DM**						**XSum**					
CLIFF	43.87	20.63	40.75	88.67	51.68	47.19	44.83	21.52	36.59	91.54	22.68	23.71
FES	46.49	22.43	43.19	89.21	54.91	52.08	47.46	24.61	39.24	**91.78**	24.29	24.54
LLaMA-2	43.79	20.46	40.69	86.65	49.53	46.89	44.58	21.39	36.47	91.28	22.61	25.34
LLaMA-2 + SliSum	45.37	22.54	42.19	88.51	54.81	51.74	47.94	24.93	39.26	91.45	27.83	25.76
Claude-2	44.83	21.62	41.54	88.18	52.97	50.26	45.85	22.68	37.24	91.17	24.35	23.56
Claude-2 + SliSum	**47.75**	**23.16**	**44.26**	**90.17**	**60.34**	58.19	48.31	**25.48**	40.07	91.46	27.19	26.84
GPT-3.5	44.35	21.02	41.28	88.36	51.62	49.25	47.69	24.82	39.42	91.29	26.74	23.79
GPT-3.5 + SumCoT	45.25	21.72	41.68	88.61	52.54	50.87	47.83	25.17	39.52	91.02	27.91	25.48
GPT-3.5 + SliSum	46.48	22.82	43.26	89.77	58.31	**58.26**	**48.74**	25.46	**40.18**	91.27	**28.58**	**27.42**
Model	**arXiv**						**PubMed**					
FactorSum	48.29	20.35	42.38	88.27	56.38	53.29	47.41	20.46	42.73	82.35	64.21	58.36
Lodoss	48.35	20.75	42.56	88.61	68.44	65.31	49.34	23.52	44.86	88.74	77.18	75.23
LLaMA-2	37.57	12.82	33.71	74.46	47.31	42.67	39.28	16.67	36.47	76.39	58.39	52.73
LLaMA-2 + SliSum	47.23	19.94	41.29	85.61	66.39	64.12	47.84	20.96	42.85	84.69	72.54	69.58
Claude-2	44.95	17.62	39.54	80.83	64.26	61.57	45.27	19.39	41.54	81.52	71.24	67.42
Claude-2 + SliSum	**50.94**	**21.86**	**45.91**	88.46	**77.43**	**75.25**	**51.49**	**24.58**	46.37	**89.62**	81.34	**79.82**
GPT-3.5	42.69	17.28	38.68	82.47	64.59	62.33	43.98	18.56	39.36	79.45	67.25	65.49
GPT-3.5 + SumCoT	45.28	18.39	40.25	83.26	65.17	62.56	45.71	19.62	41.33	81.27	67.82	62.94
GPT-3.5 + Refine	47.53	20.26	41.37	86.74	65.91	64.31	48.69	22.75	43.16	86.27	72.39	69.84
GPT-3.5 + SliSum	48.57	20.82	42.62	**88.75**	70.81	68.49	51.08	24.57	**46.76**	89.38	**81.74**	79.63

Table 3: Comprehensive evaluation results of SliSum on four datasets for factual consistency, relevance and fluency. **R-1/2/L** stands for ROUGE-1/2/L, **BS** is BERTScore, **FC** is FactCC, **SC** is SummaC. The best result per metric for each dataset is **bolded**.

sentence representations by performing summarization and segmentation simultaneously. We compare SliSum with **Refine** (LangChain, 2023b), a LLM-based long document summarization method that joins the summary of the previous text segment with the next segment, and then iteratively generate summaries for them.

4.3. Metrics

We evaluate the factual consistency, fluency and informativeness of summaries using four different metrics: (1) **FactCC** (Kryscinski et al., 2020), a weakly-supervised, model-based approach for verifying factual consistency and identifying conflicts between source documents and generated summaries, (2) **SummaC** (Laban et al., 2022) that enables natural language inference models to detect inconsistency, (3) **ROUGE**(Lin, 2004), an automatic evaluation metric for the informativeness and fluency of a summary based on lexical overlap, and (4) **BERTScore** (Zhang et al., 2020), that computes a similarity score between candidate and reference summaries using contextual embeddings.

4.4. Implementation

We run LLaMA-2-13B with Text Generation Inference[4] on 8 $\times$ 24GB NVIDIA GeForce RTX 3090 GPUs. For CNN/DM and XSum, we cluster and filter sentences using the DBSCAN algorithm with ε = 0.25 and MinPts = 2. The window size L_w is 150 words, and the step size L_s is 50 words. For arXiv and PubMed, we set distance threshold ε to 0.25, and MinPts to 3. The window size L_w is 750 words, and the step size L_s is 150 words. For SumCoT, we use the same prompts in paper. We set chunk size to 750 words in Refine. The prompts used for summary generation, sentences classification and sentences integration are listed in Table 2. Regarding other baselines used in the experiments, we use standard checkpoints provided by the authors and adopt the same configuration as in the corresponding papers, respectively.

[4]https://github.com/huggingface/text-generation-inference

5. Results

5.1. Main Results

Overall Performance. The experimental results of SliSum and baselines on four summarization datasets are reported in Table 3. In terms of short text summarization, Claude-2 with SliSum achieves a relative gain of 13.9% on FactCC and 15.8% on SummaC for CNN/DM respectively. Notably, Claude-2 with SliSum also performs the best on both ROUGE and BERTScore compared with other baselines, indicating that SliSum simultaneously improves informativeness and fluency of summaries. For long document summarization, Claude-2 with SliSum achieves a notable relative gain of 20.5% on FactCC and 22.2% on SummaC for arXiv respectively. Although FactorSum and Lodoss achieve close performance to SliSum on reference-based similarity metrics, they are significantly lower than SliSum on FactCC and SummaC. Regarding

PubMed dataset, SliSum enables Claude-2 and GPT-3.5 to achieve almost the same excellent performance, and outperform other baselines on all metrics.

Overall, LLM combined with SliSum uniformly outperforms base model itself with a wide margin on all metrics, which demonstrates the generality and effectiveness of SliSum. Specially, among three LLMs, SliSum reveals the most outstanding performance on Claude-2 and the greatest gain on LLaMA-2. Besides, SliSum applied to GPT-3.5 outperforms SumCoT and Refine on all four datasets. This shows that the superiority of our approach to other LLM-based baselines. We observe that SliSum achieves higher improvement on arXiv and PubMed than CNN/DM and XSum, and leads to greater performance gain on faithfulness metrics than on informativeness metrics. Therefore, our approach can substantially improves faithfulness of various LLMs in short and long text summarization, while maintaining their fluency and informativeness.

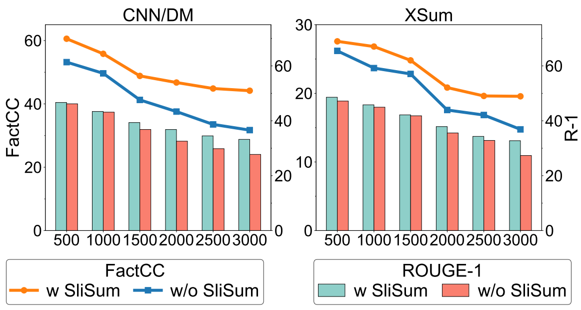


Figure 2: The performance of GPT-3.5 evaluated on samples of different length.

SliSum enhances the ability to process long contexts. We observe that SliSum has different performance gains on different datasets, which may be due to the style or length of the text. To eliminate the distractor of the text style and further demonstrate the effectiveness of our approach on news texts. We evaluate the factual consistency of GPT-3.5 on test samples of length ranges from 500 to 3000. We select 25 samples from CNN/DM and XSum with a length that is less than 5% different from the specified length to construct the test sets, respectively.

As shown in Figure 2, compared to the original model, SliSum significantly improves the faithfulness of GPT-3.5 in news texts of various length, and the performance gain is positively correlated with the length. This is consistent with the results in Table 3 and indicates that SliSum is capable of leading to larger benefits from increasingly long context. Besides, SliSum prevents the FactCC and R-1 score of the summaries from continually decreasing as source articles grow longer. For example, when the length of articles increases from 1000 to 3000, the FactCC score of summaries generated for CNN/DM by SliSum remains stable, while the scores of the baseline still decreases. We demonstrate that our approach enhances the ability to handle long texts in a variety of styles.

5.2. Ablation Study

SliSum contains two essential modifications: sliding generation, filteration and aggregation based on self-consistency. In order to investigate the effect of modification designed in the SliSum, we conduct the following ablation studies by comparing SliSum with corresponding variation using GPT-3.5.

Dataset	Method	R-1	R-L	FC	SC
PubMed	Single	44.76	40.29	70.35	67.41
	Parallel	47.38	42.72	74.36	71.84
	Sliding	**51.08**	**46.76**	**81.74**	**79.63**

	Position	1-1000	1001-2000	2001-3000	3001-
arXiv	Single	39.26%	24.63%	13.58%	22.53%
	Sliding	29.61%	26.48%	18.83%	25.08%
PubMed	Single	41.83%	14.74%	11.29%	32.11%
	Sliding	32.96%	21.57%	18.32%	27.15%

Table 4: Evaluation results of different context processing method, and the position distribution of the generated sentences in the source article. For SW method, we use the same parameters with Section 4.4. For PW method, we only modify L_s equal to L_w.

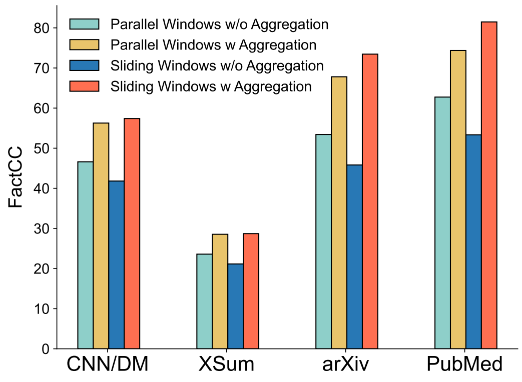


Figure 3: The factual consistency of sliding generation with aggregation and without aggregation.

Sliding Windows vs. Parallel Windows. Parallel window(**PW**), a special case of sliding window(**SW**), partitions contexts in a non-overlapping manner. To investigate the contribution of sliding generation to SliSum, we compare comprehensive performance of SW and other baselines (also generate the summary K times to perform aggregation). Table 4 shows that single window method is worse than others on PubMed, due to the distraction issue caused by long texts. In contrast, SW outperforms PW method on all three metrics, which demonstrates that sliding generation brings performance improvement to LLMs. This is because SW can take advantage of longer context to support generating summaries, whereas PW lack direct interaction between windows, resulting in the inability to adequately extract global important information. Specifically, benefiting from overlapping windows, SW can utilize information of $L_w + L_s(\lceil L_w/L_s \rceil - 1)$ words during generating summary for each text segment, while PW can only utilize L_w words. Furthermore, Table 4 shows that sliding windows facilitates LLMs to more fairly and faithfully process contexts of various length without position bias. The summaries generated by SliSum contain the content of each position range of the source article relatively evenly.

Filteration and Aggregation improve faithfulness. In order to understand the importance of filteration and aggregation based on self-consistency, we conduct ablation study by comparing SliSum to variation w/o aggregation. We remove two steps by directly generating global summaries for all local summaries (**Map-Reduce** (LangChain, 2023a)). As shown in Figure 3, the FactCC scores of both SW and PW w/o aggregation are lower than their counterparts. Notably, sliding windows w/o aggregation substantially diminish the factual consistency of summaries, even worse than parallel windows w/o aggregation. This indicates that aggregation is essential for SliSum, because sliding generation while providing more diverse information also leads to self-contradiction problem, which asks us to address this problem by filteration and aggregation based on self-consistency.

5.3. Impacts of Hyperparameters

We conduct quantitative experiments to investigate impacts of hyperparameters in SliSum on performance. Limited by computational resources and budget, we randomly select 25 articles from corresponding dataset to construct reduced test sets.

Ratio of L_w to L_s. Figure 4 (left) shows that the FactCC score when varying in ratio $K = \lfloor L_w/L_s \rfloor$ from 1 to 10. As shown in Equation 1, K represents the minimum number of process content of the article. A large K can more adequately exploit the self-consistency of LLMs to improve performance. When the ratio grows from 1 to 5, the FactCC scores increased by 9.5% and 11.5% on arXiv and PubMed respectively, yet gradually converges when the ratio grows from 5 to 10, which indicates that processing articles too many times is extremely expensive but unnecessary. To save computational resources, we set the L_w/L_s to 5 in our experiments.

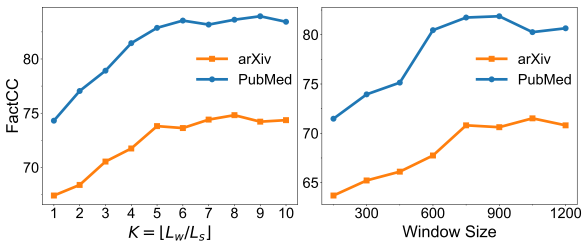


Figure 4: Impact of ratio L_w/L_s (left) and window size (right) on faithfulness of GPT-3.5. To analyze ratio, We fix $L_w = 900$ and gradually decrease L_s to adjust the ratio. To analyze window size, We fix $L_w/L_s = 5$ and increase L_w 150 words each time from 150 to 1200 words.

Window Size. Intuitively, shorter window means that LLMs can more accurately exploit information in context. But this has harmful effect on global understanding of entire article, resulting in the summary only contains locally important content. Nevertheless, a longer window will cause distraction issue and more uncertainty. As shown in Figure 4 (right), the FactCC score of summaries initially raises as the window size grows. However, when window size increases from 750 to 1200, the performance remains stable or slightly drops. The quantitative experiments results indicate that LLMs require sufficient information to generate high-quality summaries from context, but a larger window will not further boost performance.

Dataset	**arXiv**		**PubMed**	
MinPts	**R-1**	**FC**	**R-1**	**FC**
1	35.47	59.35	37.59	63.79
2	41.52	65.86	43.16	71.46
3	**48.63**	71.44	**51.38**	81.57
4	43.18	75.12	45.62	83.65
5	29.41	**76.31**	32.74	**83.92**

Table 5: The performance of SliSum with varying MinPts.

Minimum Number of Clusters (MinPts). MinPts controls clustering and filtering of sentences about the same event. A higher MinPts forces SliSum to select sentences that are important in more windows. Table 5 shows the FactCC and R-1 scores significantly increase with a bigger MinPts, but it will also lower R-1 score when MinPts greater than 3, as a result of many relatively important sentences are filtered out. Therefor, we set MinPts to 3 for arXiv and PubMed. More generally, considering the balance between faithfulness and informativeness of summaries, the MinPts is set to $\frac{1}{2}K$ for different parameter configurations.

Dataset	Avg	Max	Hausdorff
CNN/DM	0.257	0.426	0.758
XSum	0.248	0.395	0.721
arXiv	0.253	0.418	0.764
PubMed	0.246	0.409	0.743

Table 6: The distribution of all statements in local summaries.

Distance Threshold ε The choice of distance threshold ε depends on the data distribution. We count the average and the maximum distance between statements about the same event on the four datasets. Besides, we also calculate the average Hausdorff distance between sets of statements about different events. Given the set of statements X and Y for events A and B, the Hausdorff distance between X and Y is

$$d_H(X,Y) = \max\{\sup_{x\in X}\inf_{y\in Y} d(x,y), \sup_{y\in Y}\inf_{x\in X} d(x,y)\}$$

Table 6 shows that the average distance between statements about the same event is around 0.25, which is much smaller than the distance between statements about different events. Hence, we set ε to 0.25 for all datasets. The results in table 6 are consistent with our observation that statements generated by LLMs for the same event are lexically similar.

5.4. Complexity Analysis

Theoretical Analysis SliSum reduces the computational complexity in the process of summary generation compared with the base model. For the input of length L, the computational complexity of most LLMs (such as LLaMA) is $O(L^2)$. However, after applying SliSum to base models, the computational complexity of summary generation decreases to $O(L)$.

Generally, given window size L_w, step size L_s ($L_s \le L_w$) and an article whose length is L is far larger than the window size, $K = L_w/L_s$ (see Equation 1). Our approach needs to process a total of $(L/L_s + (K-1)^2)$ text segments of length L_w, hence, the complexity of SliSum is

$$O(K \times L_w \times L + (K-1)^2 \times L_w^2) = O(L) \quad (3)$$

By solving the equation, we can see that when the input length is greater than $1.36 \times K \times L_w$, the computational cost of summary generation of SliSum is less than base model.

Of course, SliSum also includes filteration and aggregation, which brings additional computational costs. However, the DBSCAN algorithm runs very fast. As for the final aggregation process, LLM usually only needs to process very few sentences. Importantly, SliSum splits the original task into smaller subtasks, so parallelization can be used to speed up inference. Overall, SliSum slightly increases the computational cost compared to the base model.

Dataset	CNN/DM	XSum	arXiv	PubMed
LLaMA-2	1.78	1.06	11.83	7.19
LLaMA-2 + SliSum	3.41	2.27	17.68	11.75

Table 7: The time cost (min) of LLaMA-2-13B with and without SliSum.

Quantitative Experiments In order to verify the above theoretical studies, we recorded the inference time of the LLaMA-2-13B on four datasets (including 100 articles) to evaluate the computational cost. Table 7 indicates the LLaMA-2-13B with SliSum only took twice as long as the LLaMA-2-13B without SliSum. By contrast, as shown in Table 3, SliSum substantially improves faithfulness of various LLMs. Therefore, SliSum slightly increases the computational cost, but brings relatively high performance gains.

6. Conclusion

In this paper, we propose **SliSum**, a novel LLM summary generation architecture, which improves faithfulness of LLMs in both short and long text summarization without additional resources and fine-tuning. SliSum leverages sliding generation as instruments for exploiting the self-consistency of LLMs, enables LLMs to more fairly and faithfully process contexts of various length, and solve contradictions between local summaries by clustering and filtering. Extensive experiments demonstrate the effectiveness of SliSum in diverse LLMs. SliSum empowers LLMs to generate faithful summaries while maintaining their fluency and informativeness. Furthermore, we conduct ablation studies to justify the modification made in the SliSum and investigate the impact of hyperparameters in SliSum. We also demonstrate through theoretical analysis and quantitative experiments that SliSum only slightly increases the computational cost.

Acknowledgements

This work was supported by the Zhejiang Provincial Natural Science Foundation of China under Grant No. LZ23F020009, the NSFC project (No. 62072399), MoE Engineering Research Center of Digital Library, China Research Centre on Data and Knowledge for Engineering Sciences and Technology, and the Fundamental Research Funds for the Central Universities. We also express our sincere gratitude to anonymous reviewers for their invaluable feedback and constructive comments.

7. Bibliographical References

Anthropic. 2023. Model card and evaluations for claude models. Technical report, Anthropic.

Vidhisha Balachandran, Hannaneh Hajishirzi, William Cohen, and Yulia Tsvetkov. 2022. Correcting diverse factual errors in abstractive summarization via post-editing and language model infilling. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9818–9830, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

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.

Shuyang Cao and Lu Wang. 2022. HIBRIDS: Attention with hierarchical biases for structure-aware long document summarization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 786–807, Dublin, Ireland. Association for Computational Linguistics.

Anthony Chen, Panupong Pasupat, Sameer Singh, Hongrae Lee, and Kelvin Guu. 2023. Purr: Efficiently editing language model hallucinations by denoising language model corruptions.

Xiuying Chen, Mingzhe Li, Xin Gao, and Xiangliang Zhang. 2022. Towards improving faithfulness in abstractive summarization. In *Advances in Neural Information Processing Systems*, volume 35, pages 24516–24528. Curran Associates, Inc.

Anshuman Chhabra, Hadi Askari, and Prasant Mohapatra. 2024. Revisiting zero-shot abstractive summarization in the era of large language models from the perspective of position bias.

Sangwoo Cho, Kaiqiang Song, Xiaoyang Wang, Fei Liu, and Dong Yu. 2022. Toward unifying text segmentation and long document summarization. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 106–118, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Arman Cohan, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Seokhwan Kim, Walter Chang, and Nazli Goharian. 2018. A discourse-aware attention model for abstractive summarization of long documents. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 615–621, New Orleans, Louisiana. Association for Computational Linguistics.

Peng Cui and Le Hu. 2021. Sliding selector network with dynamic memory for extractive summarization of long documents. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5881–5891, Online. Association for Computational Linguistics.

Shehzaad Dhuliawala, Mojtaba Komeili, Jing Xu, Roberta Raileanu, Xian Li, Asli Celikyilmaz, and Jason Weston. 2023. Chain-of-verification reduces hallucination in large language models.

Xuan-Dung Doan, Le-Minh Nguyen, and Khac-Hoai Nam Bui. 2022. Multi graph neural network for extractive long document summarization. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 5870–5875, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. 2023. Improving factuality and reasoning in language models through multiagent debate.

Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, pages 226–231, Portland, USA. AAAI Press.

Alex Fabbri, Prafulla Kumar Choubey, Jesse Vig, Chien-Sheng Wu, and Caiming Xiong. 2022. Improving factual consistency in summarization with compression-based post-editing. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9149–9156, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Marcio Fonseca, Yftah Ziser, and Shay B. Cohen. 2022. Factorizing content and budget decisions in abstractive summarization of long documents. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6341–6364, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Zorik Gekhman, Jonathan Herzig, Roee Aharoni, Chen Elkind, and Idan Szpektor. 2023.

TrueTeacher: Learning factual consistency evaluation with large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2053–2070, Singapore. Association for Computational Linguistics.

Zhibin Gou, Zhihong Shao, Yeyun Gong, yelong shen, Yujiu Yang, Nan Duan, and Weizhu Chen. 2024. CRITIC: Large language models can self-correct with tool-interactive critiquing. In *The Twelfth International Conference on Learning Representations*.

Tanya Goyal, Junyi Jessy Li, and Greg Durrett. 2023. News summarization and evaluation in the era of gpt-3.

Nianlong Gu, Elliott Ash, and Richard Hahnloser. 2022. MemSum: Extractive summarization of long documents using multi-step episodic Markov decision processes. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6507–6522, Dublin, Ireland. Association for Computational Linguistics.

Chi Han, Qifan Wang, Wenhan Xiong, Yu Chen, Heng Ji, and Sinong Wang. 2023. Lm-infinite: Simple on-the-fly length generalization for large language models.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

Kung-Hsiang Huang, Hou Pong Chan, and Heng Ji. 2023. Zero-shot faithful factual error correction. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5660–5676, Toronto, Canada. Association for Computational Linguistics.

Huiqiang Jiang, Qianhui Wu, Xufang Luo, Dongsheng Li, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. 2023. Longllmlingua: Accelerating and enhancing llms in long context scenarios via prompt compression.

Minki Kang, Jin Myung Kwak, Jinheon Baek, and Sung Ju Hwang. 2023. Knowledge graph-augmented language models for knowledge-grounded dialogue generation.

Wojciech Kryscinski, Bryan McCann, Caiming Xiong, and Richard Socher. 2020. Evaluating the factual consistency of abstractive text summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9332–9346, Online. Association for Computational Linguistics.

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.

LangChain. 2023a. Summarization option 2. map-reduce.

LangChain. 2023b. Summarization option 3. refine.

Junyi Li, Xiaoxue Cheng, Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. 2023a. HaluEval: A large-scale hallucination evaluation benchmark for large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6449–6464, Singapore. Association for Computational Linguistics.

Yucheng Li, Bo Dong, Frank Guerin, and Chenghua Lin. 2023b. Compressing context to enhance inference efficiency of large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6342–6353, Singapore. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. TruthfulQA: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3214–3252, Dublin, Ireland. Association for Computational Linguistics.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2023a. Lost in the middle: How language models use long contexts.

Tianyu Liu, Yizhe Zhang, Chris Brockett, Yi Mao, Zhifang Sui, Weizhu Chen, and Bill Dolan. 2022. A token-level reference-free hallucination detection benchmark for free-form text generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6723–6737, Dublin, Ireland. Association for Computational Linguistics.

Yixin Liu, Budhaditya Deb, Milagro Teruel, Aaron Halfaker, Dragomir Radev, and Ahmed Hassan Awadallah. 2023b. On improving summarization factual consistency from natural language feedback. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15144–15161, Toronto, Canada. Association for Computational Linguistics.

Junyu Luo, Cao Xiao, and Fenglong Ma. 2023. Zero-resource hallucination prevention for large language models.

Potsawee Manakul, Adian Liusie, and Mark Gales. 2023. SelfCheckGPT: Zero-resource black-box hallucination detection for generative large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9004–9017, Singapore. 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.

Niels Mündler, Jingxuan He, Slobodan Jenko, and Martin Vechev. 2024. Self-contradictory hallucinations of large language models: Evaluation, detection and mitigation. In *The Twelfth International Conference on Learning Representations*.

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.

OpenAI. 2023. Introducing chatgpt.

Baolin Peng, Michel Galley, Pengcheng He, Hao Cheng, Yujia Xie, Yu Hu, Qiuyuan Huang, Lars Liden, Zhou Yu, Weizhu Chen, and Jianfeng Gao. 2023. Check your facts and try again: Improving large language models with external knowledge and automated feedback.

Tuan-Anh Phan, Ngoc-Dung Ngoc Nguyen, and Khac-Hoai Nam Bui. 2022. HeterGraphLongSum: Heterogeneous graph neural network with passage aggregation for extractive long document summarization. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 6248–6258, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Dongqi Pu, Yifan Wang, and Vera Demberg. 2023. Incorporating distributions of discourse structure for long document abstractive summarization. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5574–5590, Toronto, Canada. Association for Computational Linguistics.

Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2023. In-context retrieval-augmented language models.

Nir Ratner, Yoav Levine, Yonatan Belinkov, Ori Ram, Inbal Magar, Omri Abend, Ehud Karpas, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2023. Parallel context windows for large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6383–6402, Toronto, Canada. Association for Computational Linguistics.

Mathieu Ravaut, Aixin Sun, Nancy F. Chen, and Shafiq Joty. 2024. On context utilization in summarization with large language models.

Paul Roit, Johan Ferret, Lior Shani, Roee Aharoni, Geoffrey Cideron, Robert Dadashi, Matthieu Geist, Sertan Girgin, Leonard Hussenot, Orgad Keller, Nikola Momchev, Sabela Ramos Garea, Piotr Stanczyk, Nino Vieillard, Olivier Bachem, Gal Elidan, Avinatan Hassidim, Olivier Pietquin, and Idan Szpektor. 2023. Factually consistent summarization via reinforcement learning with textual entailment feedback. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6252–6272, Toronto, Canada. Association for Computational Linguistics.

Yiwen Shi, Ping Ren, Jing Wang, Biao Han, Taha ValizadehAslani, Felix Agbavor, Yi Zhang, Meng Hu, Liang Zhao, and Hualou Liang. 2023. Leveraging gpt-4 for food effect summarization to enhance product-specific guidance development via iterative prompting.

Derek Tam, Anisha Mascarenhas, Shiyue Zhang, Sarah Kwan, Mohit Bansal, and Colin Raffel. 2023. Evaluating the factual consistency of large language models through news summarization. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 5220–5255, Toronto, Canada. Association for Computational Linguistics.

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

Szymon Tworkowski, Konrad Staniszewski, Mikoł aj Pacek, Yuhuai Wu, Henryk Michalewski, and Piotr Mił oś. 2023. Focused transformer: Contrastive training for context scaling. In *Advances in Neural Information Processing Systems*, volume 36, pages 42661–42688. Curran Associates, Inc.

Jesse Vig, Alexander Fabbri, Wojciech Kryscinski, Chien-Sheng Wu, and Wenhao Liu. 2022. Exploring neural models for query-focused summarization. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1455–1468, Seattle, United States. Association for Computational Linguistics.

David Wan and Mohit Bansal. 2022. FactPEGASUS: Factuality-aware pre-training and fine-tuning for abstractive summarization. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1010–1028, Seattle, United States. Association for Computational Linguistics.

Peiyi Wang, Lei Li, Liang Chen, Zefan Cai, Dawei Zhu, Binghuai Lin, Yunbo Cao, Qi Liu, Tianyu Liu, and Zhifang Sui. 2023a. Large language models are not fair evaluators.

Tianshu Wang, Faisal Ladhak, Esin Durmus, and He He. 2022. Improving faithfulness by augmenting negative summaries from fake documents. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11913–11921, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023b. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*.

Yiming Wang, Zhuosheng Zhang, and Rui Wang. 2023c. Element-aware summarization with large language models: Expert-aligned evaluation and chain-of-thought method. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8640–8665, Toronto, Canada. Association for Computational Linguistics.

Wenhao Wu, Wei Li, Jiachen Liu, Xinyan Xiao, Ziqiang Cao, Sujian Li, and Hua Wu. 2022. FRSUM: Towards faithful abstractive summarization via enhancing factual robustness. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 3640–3654, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Wen Xiao, Yujia Xie, Giuseppe Carenini, and Pengcheng He. 2024. Personalized abstractive summarization by tri-agent generation pipeline. In *Findings of the Association for Computational Linguistics: EACL 2024*, pages 570–581, St. Julian's, Malta. Association for Computational Linguistics.

Jian Xie, Kai Zhang, Jiangjie Chen, Renze Lou, and Yu Su. 2024. Adaptive chameleon or stubborn sloth: Revealing the behavior of large language models in knowledge conflicts. In *The Twelfth International Conference on Learning Representations*.

Jiawen Xie, Qi Su, Shaoting Zhang, and Xiaofan Zhang. 2023. Alleviating exposure bias via multi-level contrastive learning and deviation simulation in abstractive summarization. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 9732–9747, Toronto, Canada. Association for Computational Linguistics.

Qianqian Xie, Jimin Huang, Tulika Saha, and Sophia Ananiadou. 2022. GRETEL: Graph contrastive topic enhanced language model for long document extractive summarization. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 6259–6269, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Peng Xu, Wei Ping, Xianchao Wu, Lawrence McAfee, Chen Zhu, Zihan Liu, Sandeep Subramanian, Evelina Bakhturina, Mohammad Shoeybi, and Bryan Catanzaro. 2024. Retrieval meets long context large language models. In *The Twelfth International Conference on Learning Representations*.

Tianci Xue, Ziqi Wang, Zhenhailong Wang, Chi Han, Pengfei Yu, and Heng Ji. 2023. Rcot: Detecting and rectifying factual inconsistency in reasoning by reversing chain-of-thought.

Haopeng Zhang, Xiao Liu, and Jiawei Zhang. 2022a. HEGEL: Hypergraph transformer for long document summarization. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 10167–10176, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Haopeng Zhang, Xiao Liu, and Jiawei Zhang. 2023a. Extractive summarization via ChatGPT for faithful summary generation. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 3270–3278, Singapore. Association for Computational Linguistics.

Haopeng Zhang, Semih Yavuz, Wojciech Kryscinski, Kazuma Hashimoto, and Yingbo Zhou. 2022b. Improving the faithfulness of abstractive summarization via entity coverage control. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 528–535, Seattle, United States. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Tianyi Zhang, Faisal Ladhak, Esin Durmus, Percy Liang, Kathleen McKeown, and Tatsunori B. Hashimoto. 2023b. Benchmarking large language models for news summarization.

Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, Longyue Wang, Anh Tuan Luu, Wei Bi, Freda Shi, and Shuming Shi. 2023c. Siren's song in the ai ocean: A survey on hallucination in large language models.

Shen Zheng, Jie Huang, and Kevin Chen-Chuan Chang. 2023. Why does chatgpt fall short in providing truthful answers?