On Context Utilization in Summarization with Large Language Models

Mathieu Ravaut[1,2], **Aixin Sun**[1], **Nancy F. Chen**[1,2,4,5], **Shafiq Joty**[1,3]
[1] Nanyang Technological University, Singapore
[2] Institute of Infocomm Research (I[2]R), A*STAR, Singapore
[3] Salesforce Research
[4] CNRS@CREATE, Singapore
[5] Centre for Frontier AI Research (CFAR), A*STAR, Singapore

Abstract

Large language models (LLMs) excel in abstractive summarization tasks, delivering fluent and pertinent summaries. Recent advancements have extended their capabilities to handle long-input contexts, exceeding 100k tokens. However, in question answering, language models exhibit uneven utilization of their input context. They tend to favor the initial and final segments, resulting in a U-shaped performance pattern concerning where the answer is located within the input. This bias raises concerns, particularly in summarization where crucial content may be dispersed throughout the source document(s). Besides, in summarization, mapping facts from the source to the summary is not trivial as salient content is usually re-phrased. In this paper, we conduct the first comprehensive study on context utilization and position bias in summarization. Our analysis encompasses 6 LLMs, 10 datasets, and 5 evaluation metrics. We introduce a new evaluation benchmark called MiddleSum on the which we benchmark two alternative inference methods to alleviate position bias: hierarchical summarization and incremental summarization[1].

1 Introduction

Large language models (LLMs) have drastically transformed the landscape of NLP recently (Brown et al., 2020). With instruction tuning (Ouyang et al., 2022; Chung et al., 2022), LLMs made a major leap forward in conditional (prompted) content generation, and can generate satisfying outputs without the need to fine-tune on a specific task. In abstractive summarization specifically, this approach has arguably opened a new paradigm: summaries generated by LLMs are highly fluent, grammatical and relevant (Goyal et al., 2022). Despite noticeably lower scores on automatic metrics such as ROUGE (Lin, 2004) or BERTScore (Zhang et al., 2019), summaries generated by LLMs are largely preferred by humans over summaries from state-of-the-art fine-tuned models like BRIO (Liu et al., 2022b, 2023c). In fact, on XSum, GPT-3.5 summaries are even on par with re-annotated human-written summaries, and much better than the dataset's original ground-truth, according to human evaluators (Zhang et al., 2023b). LLMs also show promising capability in evaluating summaries generated by other systems, including LLMs (Fu et al., 2023; Luo et al., 2023; Shen et al., 2023a).

Despite this success, a few major technological bottlenecks remain with LLMs, including the maximum length of their context window. The standard context window length for open-source LLMs is 2k tokens (Brown et al., 2020; Scao et al., 2022; Penedo et al., 2023; Touvron et al., 2023a), which drastically limits their usefulness for long-input summarization (Shaham et al., 2022). Several techniques were proposed to extend the context window, including ALiBi (Press et al., 2021), LeX (Sun et al., 2022), position interpolation (Chen et al., 2023) and YaRN (Peng et al., 2023). While some of them claim up to 100k+ tokens processing capacity (Peng et al., 2023), it remains unclear how much such methods help on long-context summarization.

Scaling up context length would only succeed if a key question gets addressed first: ***do LLMs make proper use of their entire context?*** Recent work (Liu et al., 2023a) suggested that, surprisingly, such a simple assumption may not hold: through experiments on multi-document question answering and key-value retrieval, the authors find that LLMs mostly focus on the *beginning* and *end* of the (long) context window. Plotting performance with regards to the position of the important information exhibits a *U-shape*, with performance high at first (beginning of the source), then dropping, and rising again at the end. Worryingly, in the middle of the context window, LLMs' performance can drop to even *below random chance*, calling for greater

[1]Our code and data can be found here: https://github.com/ntunlp/MiddleSum.

Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 2764–2781
August 11-16, 2024

examination of LLMs' behaviors with regard to the position of information within the source.

In this work, we investigate in depth how LLMs use their context window in abstractive summarization. Unlike in question-answering, mapping facts in the output to a specific snippet in the source is not straightforward in abstractive summarization, due to the high-level of re-phrasing and compression. We conduct a large-scale study with 6 LLMs, 10 datasets covering many aspects of summarization, and 5 highly diverse automatic metrics. Our contributions are threefold:

- We conduct the first large-scale analysis on context utilization in abstractive summarization, and the impact of the position of salient information on performance. We show that the U-shape or *middle-curse* exhibited by (Liu et al., 2023a) also holds in abstractive summarization.
- We craft an evaluation dataset (**MiddleSum**) where important information is concentrated in the middle of the context, enabling us to automatically quantify how much LLMs are affected by the *middle-curse*.
- We benchmark two alternative methods for inference on MiddleSum: *hierarchical summarization* and *incremental summarization*, showing their promise at alleviating the *middle curse* (especially in the scientific paper domain).

2 Experimental Setup

Datasets We cover a broad set of diverse abstractive summarization tasks, varying length and domain. We include 5 datasets of standard length (source is below 2k tokens, which always fits in the context window): (i) **CNN/DailyMail** (Hermann et al., 2015), (ii) **XSum** (Narayan et al., 2018), (iii) **Reddit-TIFU** (Kim et al., 2019), (iv) **SAMSum** (Gliwa et al., 2019), and (v) **Multi-XScience** (Lu et al., 2020). We also include another 5 long-input summarization datasets: (i) **Arxiv** and (ii) **PubMed** (Cohan et al., 2018), (iii) **GovReport** (Huang et al., 2021), (iv) **SummScreenFD** (Chen et al., 2022), and (v) **Multi-News** (Fabbri et al., 2019). A high-level view of each dataset is shown in Table 1, and detailed statistics are presented in Appendix A. For all datasets, we run experiments on the test set, subsampling 1,000 data points if its size is greater than 1,000, or using the entire test set otherwise.

Models We experiment with 6 popular and high-performing LLMs:

Dataset	Input length		# Documents		In Flan?	
	Standard	Long	Single	Multi	Yes	No
CNN/DM (Hermann et al., 2015)	✓		✓		✓	
XSum (Narayan et al., 2018)	✓		✓		✓	
Reddit-TIFU (Kim et al., 2019)	✓		✓			✓
SAMSum (Gliwa et al., 2019)	✓		✓		✓	
Multi-XScience (Lu et al., 2020)	✓			✓		✓
Arxiv (Cohan et al., 2018)		✓	✓			✓
PubMed (Cohan et al., 2018)		✓	✓			✓
GovReport (Huang et al., 2021)		✓	✓			✓
SummScreenFD (Chen et al., 2022)		✓	✓			✓
Multi-News (Fabbri et al., 2019)		✓		✓	✓	

Table 1: Summarization datasets under study. In standard length datasets, the context and summary fit within a 2k tokens LLM context window.

- **Flan-UL2** is a 20B parameters encoder-decoder model pre-trained on 1T tokens. It is based on the UL2 20B model (Tay et al., 2022), with the addition of Flan-T5 (Chung et al., 2022) instruction fine-tuning. The context window is 2k tokens.
- **Llama-2** (Touvron et al., 2023b) is a recently introduced powerful decoder-only model pre-trained on 2T tokens, ranging from 7B to 70B parameters, and with a 4k tokens context window. We use the **7B** and **13B** models.
- **Xgen-7B** (Nijkamp et al., 2023) is a 7B decoder-only model pre-trained on up to 1.5T tokens. It supports an 8k tokens context window.
- **Mistral-7B** (Jiang et al., 2023a) is also an 8k-context 7B decoder-only model, with performance slightly better than Llama-2-13B.
- **GPT-3.5** (`gpt-3.5-turbo-0125`), which is known to be good at generating summaries and has a 16k tokens context window.[2]

We analyze the open-source models through HuggingFace *transformers* library (Wolf et al., 2020), and use the OpenAI API for GPT-3.5. We use the instruction-tuned (or *chat*) checkpoints for Llama-2, Xgen-7B and Mistral-7B. Note that popular instruction-tuning datasets such as Flan (Wei et al., 2021) include some of the datasets we study: CNN/DM, XSum, SAMSum and Multi-News. To run inference, we use the following prompt: `Read the following text and summarize it: [text]. Summarize the above text in [n] sentences. Summary:` where `n` is set to an average number of target sentences per dataset (see Appendix A). We infer all models in *bfloat16* and sample summaries with top-k sampling (Fan et al., 2018) using $k = 50$ and temperature $T = 0.3$.

[2]To reduce cost, we subsample 300 data points when using GPT-3.5. Our total API cost is inferior to 300 USD.

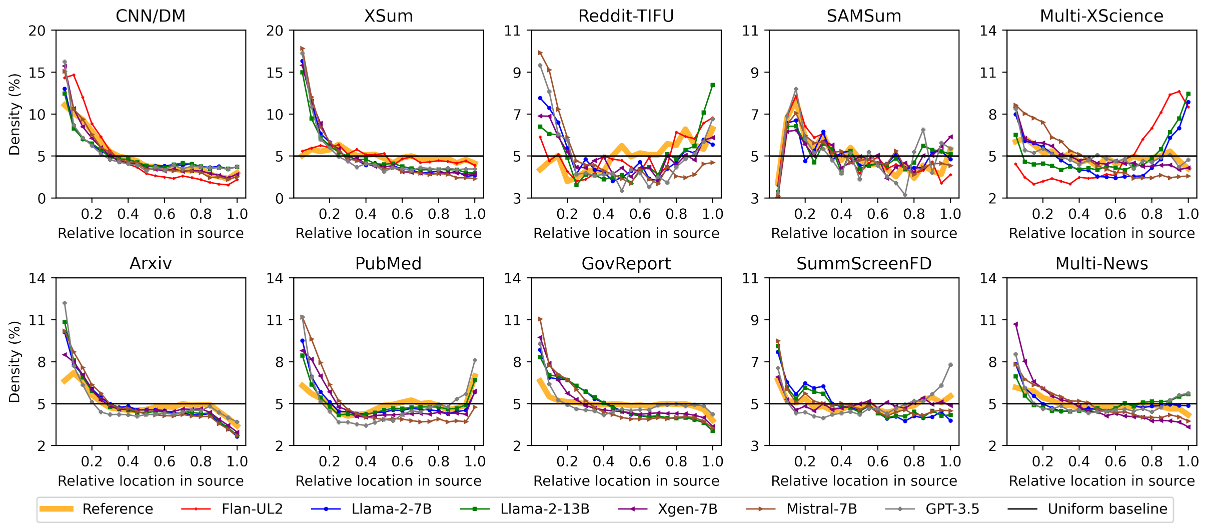

Figure 1: Distribution of the relative location of summary bigrams within the source. We split each source document into 20 bins of the same number of words, and plot the distribution of summary bigrams over source bins.

Evaluation Measures Summarization evaluation is especially challenging in the LLM era, as most automatic metrics poorly correlate with human preferences (Goyal et al., 2022; Liu et al., 2023b). To get a broad picture of performance, we evaluate with metrics as diverse as possible. First, we consider **reference-based** metrics: **ROUGE-2** (Lin, 2004), which measures bigram overlap, **BERTScore** (Zhang et al., 2019), which measures semantic similarity with BERT (Devlin et al., 2019) embeddings, and **A3CU** (Liu et al., 2023c), which extracts facts in the form of Atomic Content Units (ACUs) (Liu et al., 2023b), and checks the presence of ACUs between prediction and reference. As **reference-free** metrics, we include **SummaC** (Laban et al., 2022), a leading factual consistency evaluation metric relying on entailment scores between pairs of source and summary sentences. We also leverage **GPT-3.5** again (still `gpt-3.5-turbo-0125`), this time as a summarization *evaluator*, which is proven to be a strong natural language generation evaluator (Wang et al., 2023a; Shen et al., 2023a; Jain et al., 2023). We prompt the model with the source and generated summary (which fits in GPT-3.5's 16k context window) and ask to output a score on a likert scale from 1 to 5. We refer to Appendix B for the full prompt template.[3]

We report the performance of LLMs on all 10 datasets, alongside a comparison to SOTA, in Appendix C. FLan-UL2 dominates on standard-length datasets, but GPT-3.5 has the upper hand on the long-input ones. Performance itself is not our focus in this paper, but rather which position-related factors influence it. We discard Flan-UL2 on long-input datasets due to very poor performance.

3 Experiments

In this section, we describe a series of experiments aimed at understanding how LLMs treat information in their input depending on the position.

3.1 RQ1: Where in the *source* do LLMs take their information from?

We first investigate summaries generated by LLMs, and map them to specific parts of the input. Unlike in question-answering or *extractive* summarization, mapping salient information from a summary to the source is not trivial in *abstractive* summarization.

We follow the approach used in (Kim et al., 2019; Zhao et al., 2022a) and compute the relative position of bigams from generated summaries within the source documents, as a proxy for the position of salient information. We only use unique bigrams from summaries, and for each bigram, find all its occurrences within the source, if there are any. We then split the source into 20 bins of the same number of words, and compute the fraction of matched bigrams found in each bin. On top of the LLMs described above, we include the position of bigrams from reference summaries, and a uniform baseline.

As seen in Fig. 1, all summarization datasets except XSum, and Reddit-TIFU show some *lead bias*: salient bigrams from the reference (orange

[3]We also subsample 300 data points when using GPT-3.5 as an evaluator, to reduce API cost.

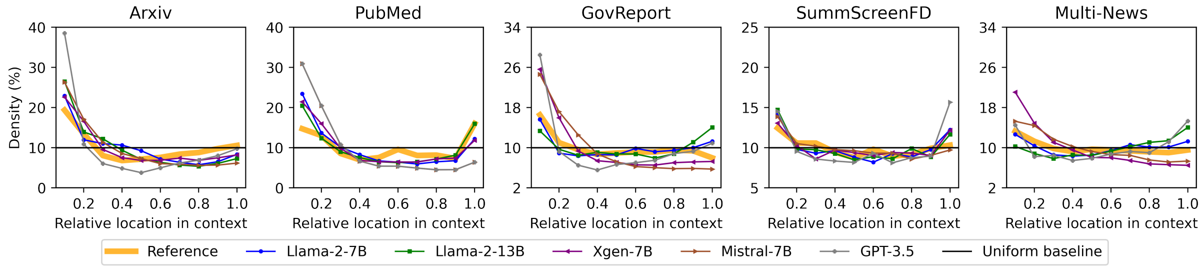

Figure 2: Distribution of relative location of input context sentences aligned with sentences from summaries. X-axis corresponds to the source sentence bin, y-axis to the fraction of aligned sentences in each bin.

curves) are more likely to be found at the beginning of the source. However, LLMs show a significantly stronger lead bias on all datasets: bigrams from LLMs summaries are much more likely to be found in the first 20% words of the source. It is especially striking on XSum (except for Flan-UL2), Reddit-TIFU, Arxiv, PubMed and GovReport. On XSum, Flan-UL2 closely matches the reference distribution, which we attribute to its better instruction tuning. Results in Appendix D confirm that bigram distribution for LLMs and references are statistically different (p-value of Kolmogorov-Smirnov test (Massey Jr, 1951) inferior to 0.001) in all but 4 out of the 55 (dataset, LLM) setups: Flan-UL2 on XSum and SAMSum, Llama-2-7B on SAMSum and GPT-3.5 on SAMSum. We conclude that ***LLMs focus on contents at the beginning of the source document(s).***

3.2 RQ2: Where do LLMs look at within their *context window*?

In the previous experiment's design, LLMs may not see the entire source in long-input summarization datasets, due to their limited context window, which is shorter than the source on the long-input datasets. We now focus on input information accessible to LLMs, and only consider salient information if it falls within the context window. Besides, since the same bigram may occur multiple times throughout the source, we adjust the methodology for saliency estimation. We align sentences in generated summaries to sentences in the context, following the procedure described in (Zhou et al., 2018) and also used in (Adams et al., 2023). Specifically, we greedily select source sentences (among the ones fitting in context window) until the ROUGE-1 F1 score between the set of selected source sentences and the summary stops increasing. The resulting set of source sentences forms a proxy of the visible salient input information being rephrased by the model when summarizing. We split each truncated source document into 10 bins of the same number of sentences, and map each aligned source sentence to its bin. Note that bins are not directly comparable across models, as context length varies across models.

As we can see in Fig. 2, sentences from the first 10% or last 10% of the input context are much more represented than others. A clear U-shape emerges on PubMed and SummScreenFD for all LLMs. This is intriguing knowing that the LLMs have different context window lengths, and the last 10% of each context window may contain content of varying saliency. In other words, ***LLMs seem to be mostly re-phrasing information from the beginning or the end of their context window***.

Results in Appendix E confirm an even stronger position bias with *base* models.

3.3 RQ3: Does LLMs performance depend on the position of salient information?

Results from the last experiment raise the question of whether LLMs' summarization performance changes depending on *where salient information is located within the input*. As an approximation for salient information, we consider the alignment between summary sentences and source sentences like in Fig. 2, but this time using the *reference* summaries. Each reference summary is mapped to the source sentences it maximizes ROUGE-1 F1 against, which may be scattered across the whole source. We convert each source sentence to its cumulative word count from the beginning of the source, and take the average as an approximation of the mean position of salient information within the source. We keep data points with mean salient position fitting within the LLM context window.

We examine performance changes with regard to this salient position. To do so, we compute the Spearman correlation coefficient between salient

Metric	Model	CNN/DM	XSum	Reddit	SAMSum	Multi-X	*AVG*	Arxiv	PubMed	GovReport	SummScreenFD	Multi-N	*AVG*
ROUGE-2	Flan-UL2	-0.296	-0.124	0.048	-0.069	-0.201	*-0.128*	–	–	–	–	–	–
	Llama-2-7B	-0.160	-0.023	0.063	-0.059	-0.100	*-0.056*	0.022	-0.113	-0.109	-0.079	-0.210	*-0.098*
	Llama-2-13B	-0.166	-0.086	0.031	-0.078	-0.039	*-0.068*	-0.017	-0.081	-0.166	-0.139	-0.213	*-0.123*
	Xgen-7B	-0.228	-0.042	0.066	-0.039	-0.041	*-0.056*	0.028	-0.091	-0.405	0.063	-0.283	*-0.138*
	Mistral-7B	-0.289	-0.031	0.006	-0.024	-0.052	*-0.078*	-0.270	-0.279	-0.585	-0.132	-0.324	*-0.318*
	GPT-3.5	-0.323	-0.027	-0.031	-0.097	0.088	*-0.078*	0.026	-0.093	-0.123	-0.061	-0.233	*-0.097*
BERTScore	Flan-UL2	-0.331	-0.185	0.062	-0.144	-0.399	*-0.187*	–	–	–	–	–	–
	Llama-2-7B	-0.173	-0.012	0.062	-0.130	-0.385	*-0.128*	-0.031	-0.203	-0.104	-0.067	-0.256	*-0.132*
	Llama-2-13B	-0.193	-0.102	0.038	-0.089	-0.352	*-0.140*	-0.082	-0.209	-0.063	-0.152	-0.279	*-0.157*
	Xgen-7B	-0.252	-0.106	0.046	-0.075	-0.343	*-0.146*	-0.017	-0.125	-0.353	-0.093	-0.345	*-0.187*
	Mistral-7B	-0.278	-0.052	0.014	-0.108	-0.416	*-0.168*	-0.348	-0.367	-0.567	-0.356	-0.403	*-0.408*
	GPT-3.5	-0.280	-0.021	-0.025	-0.174	-0.247	*-0.149*	-0.087	-0.210	-0.175	-0.142	-0.262	*-0.175*
A3CU	Flan-UL2	-0.258	-0.090	0.050	-0.123	-0.069	*-0.098*	–	–	–	–	–	–
	Llama-2-7B	-0.182	-0.076	0.028	-0.121	-0.090	*-0.088*	-0.038	-0.209	-0.154	-0.129	-0.217	*-0.149*
	Llama-2-13B	-0.190	-0.098	0.009	-0.166	0.016	*-0.086*	-0.104	-0.232	-0.111	-0.198	-0.228	*-0.175*
	Xgen-7B	-0.212	-0.130	0.023	-0.126	-0.025	*-0.094*	-0.036	-0.255	-0.211	-0.076	-0.287	*-0.173*
	Mistral-7B	-0.291	-0.110	0.004	-0.105	-0.119	*-0.126*	-0.160	-0.283	-0.305	-0.010	-0.283	*-0.208*
	GPT-3.5	-0.256	-0.022	-0.039	-0.216	0.141	*-0.078*	-0.064	-0.293	-0.201	-0.059	-0.264	*-0.176*
SummaC	Flan-UL2	-0.012	0.548	0.270	0.186	-0.035	*0.191*	–	–	–	–	–	–
	Llama-2-7B	0.088	0.552	0.375	0.227	0.224	*0.293*	0.090	0.108	0.126	-0.020	0.205	*0.102*
	Llama-2-13B	0.162	0.556	0.394	0.173	0.096	*0.276*	0.090	0.265	0.192	-0.144	0.232	*0.127*
	Xgen-7B	0.001	0.161	0.220	0.117	0.004	*0.101*	-0.208	-0.087	-0.313	-0.141	0.046	*-0.141*
	Mistral-7B	-0.045	0.515	0.149	0.069	0.154	*0.128*	-0.250	-0.103	-0.387	0.124	-0.010	*-0.125*
	GPT-3.5	0.156	0.590	0.444	0.180	0.008	*0.276*	0.058	0.237	0.061	-0.055	0.089	*0.078*
GPT-3.5	Flan-UL2	-0.020	0.196	0.027	-0.009	-0.193	*0.000*	–	–	–	–	–	–
	Llama-2-7B	0.036	0.036	-0.152	0.077	-0.153	*-0.031*	0.008	-0.116	-0.013	-0.068	-0.120	*-0.062*
	Llama-2-13B	0.039	0.072	-0.038	0.066	-0.052	*0.017*	-0.010	-0.084	-0.084	-0.060	-0.051	*-0.058*
	Xgen-7B	0.056	-0.007	-0.101	0.006	-0.174	*-0.044*	-0.058	-0.096	-0.317	-0.063	-0.055	*-0.118*
	Mistral-7B	-0.115	0.124	-0.133	0.036	-0.204	*-0.058*	-0.446	-0.322	-0.580	-0.188	-0.163	*-0.342*
	GPT-3.5	0.024	0.014	-0.108	0.131	-0.054	*0.001*	0.156	-0.008	-0.172	0.068	0.024	*0.014*

Table 2: Spearman correlation coefficient between each LLM's metric, and the mean position of salient information within the context window. Flan-UL2 is not applied to long-context summarization datasets due to its too short context window. **Multi-X** is short for Multi-XScience, **Multi-N** is Multi-News dataset, **AVG** columns represent the average over standard-length and long-input datasets, respectively. Numbers in gray correspond to non-significant Spearman scores (p-value greater than 0.05).

position and each evaluation metric in Table 2. A high absolute Spearman value means that summary quality (as measured by this metric) can change (and deteriorate) with the position of important information within the context.

There are several takeaway findings from this Table. First, we notice that on standard-length datasets, reference-based evaluation metrics are negatively correlated to position of salient information. The correlation is only moderate, yet remarkably consistent across datasets (except Reddit-TIFU) and models. This is surprising, since such datasets fit entirely in context and are not affected by truncation. In contrast, reference-free metrics show either no significant or *positive* correlation to information position. For long-input datasets, the negative trend for reference-based metrics is confirmed. On these lengthy datasets, SummaC and GPT-3.5 tend to switch from positive to negative correlation, especially for Xgen-7B and Mistral-7B. We highlight that since GPT-3.5 itself is affected by the *middle-curse* from Liu et al. (2023a), it may not accurately evaluate summarization when salient content lays in the middle of the context. In light of these results, we conservatively conclude that ***LLMs' summarization performance is sensitive to the position of salient information in the context window***.

4 Analysis

4.1 How is information in the *middle* treated?

Previous experiments show that LLMs place more emphasis on the beginning and the end of their context. We now narrow down on how LLMs treat the *middle*. To remove the effect of spread of salient information, we perform two controlled experiments in multi-document summarization. This setup enables us to shuffle the order of the input, which is not realistic for the single-document setup as it would break coherence. We only consider data points with the same number k of documents: $k = 7$ documents for Multi-XScience ($n = 329$), and $k = 5$ documents for Multi-News ($n = 219$)[4].

In the first experiment, we vary the position of salient information throughout the input. We keep a single document (the abstract of the query paper on Multi-XScience, and the document with the highest BERTScore with the reference on Multi-News), and place it at position j for $j \in \{1, \ldots, k\}$, using $k - 1$ documents from a random data point for the other slots. The single relevant document is accompanied by a [RELEVANT] header, while the other documents have an [IRRELEVANT] header, and we prompt the LLM to only summarize the relevant document. For reference-free evaluation

[4]We don't subsample from these subsets for GPT-3.5.

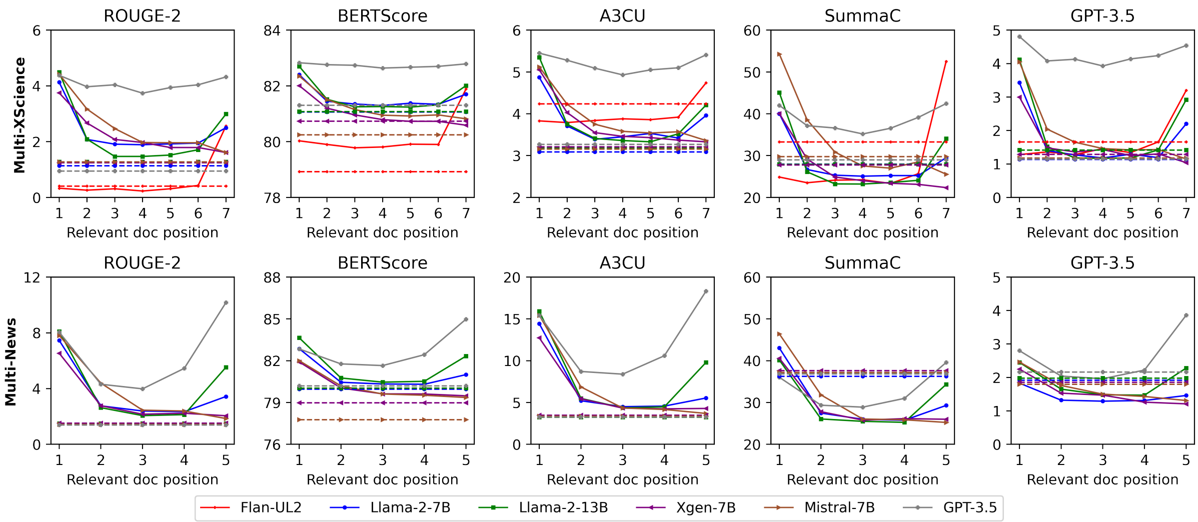

Figure 3: Multi-document summarization performance on Multi-XScience (top row) and Multi-News (bottom row) when a unique relevant document is used, and its position is varied (x-axis). Dashed horizontal lines correspond to the random baseline.

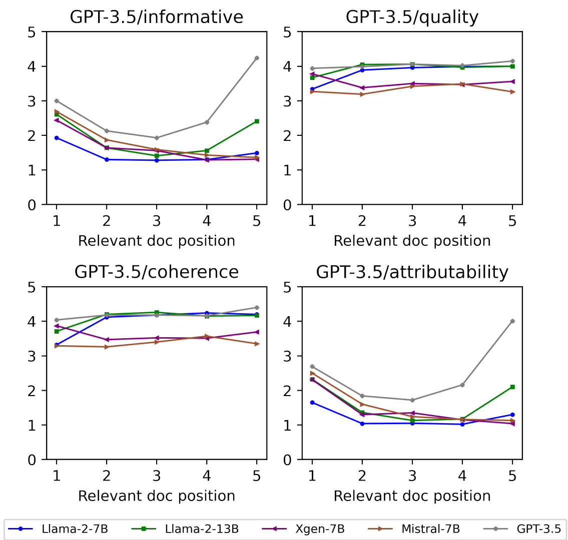

Figure 4: Fine-grained evaluation of multi-document summarization on Multi-News with GPT-3.5 when varying the position of a unique relevant input document.

Dataset	Model	Input documents	R-2	BS	A3CU	SummaC	GPT-3.5	%
Multi-X	Llama-2-7B	All 7	4.64	82.83	5.88	54.56	4.17	100.00
		First + last	4.62	82.82	5.78	47.00	4.50	98.36
		First + 5 random + last	4.43	82.64	5.37	43.25	4.01	92.40
	Llama-2-13B	All 7	4.78	83.00	6.64	42.62	4.35	100.00
		First + last	4.73	82.86	5.76	43.74	4.53	98.46
		First + 5 random + last	4.61	82.80	5.72	46.42	4.31	98.07
	Xgen-7B	All 7	5.37	82.68	6.59	44.34	4.19	100.00
		First + last	5.01	82.73	5.86	49.08	4.45	99.83
		First + 5 random + last	3.89	82.16	5.03	55.29	3.01	88.93
	Mistral-7B	All 7	5.40	82.60	6.35	63.78	4.26	100.00
		First + last	5.12	82.67	6.15	60.91	4.76	99.80
		First + 5 random + last	4.45	82.40	5.35	58.19	4.06	90.59
	GPT-3.5	All 7	5.26	83.45	7.71	35.66	4.59	100.00
		First + last	4.75	83.04	6.05	39.81	4.69	96.42
		First + 5 random + last	4.37	82.90	5.67	43.26	4.63	95.63
Multi-N	Llama-2-7B	All 5	10.76	85.04	19.06	60.09	4.00	100.00
		First + last	9.50	84.43	15.88	54.52	3.80	91.32
		First + 3 random + last	7.57	83.36	12.39	50.35	2.94	78.13
	Llama-2-13B	All 5	10.42	84.60	18.15	57.26	3.83	100.00
		First + last	9.55	84.58	16.99	49.84	3.73	93.93
		First + 3 random + last	8.27	83.79	14.84	50.09	3.18	86.14
	Xgen-7B	All 5	9.04	83.18	17.05	60.55	3.32	100.00
		First + last	7.82	83.27	14.18	51.59	3.60	92.68
		First + 3 random + last	6.30	81.85	11.66	49.02	2.66	79.51
	Mistral-7B	All 5	9.52	83.55	17.03	63.02	3.15	100.00
		First + last	9.11	83.69	14.99	67.14	3.51	100.37
		First + 3 random + last	6.59	81.66	12.50	52.14	2.45	80.17
	GPT-3.5	All 5	10.26	85.06	18.45	49.21	4.09	100.00
		First + last	8.94	84.52	15.85	45.94	4.01	92.76
		First + 3 random + last	8.53	84.31	15.33	44.67	3.80	89.81

Table 3: Performance in multi-document summarization on Multi-XScience (7 documents) and Multi-News (5 documents) when infilling the middle of the context window with random documents. **R-2** is ROUGE-2, **BS** refers to BERTScore. **%** is the mean relative performance across all metrics compared to the baseline with all documents.

metrics, we use the single relevant document as source. We also include a *random* baseline of shuffled inputs and model predictions. In Fig. 3, we see a noticeable drop in performance for all metrics when the salient document is not in the first or final position. Flan-UL2 seems to focus on the ***end*** of the context, Xgen-7B and Mistral-7B on the ***beginning***, and Llama-2 models and GPT-3.5 on ***both***. Performance can fall quite below random range, especially for reference-free metrics, confirming the worrying trend from Liu et al. (2023a).

A more fine-grained analysis with GPT-3.5 in Fig. 4 evaluating specific attributes (following the method in Adams et al. (2023), see Appendix B) reveals more details. *Coherence* and *quality* remain high and stable. In other words, the text outputs from the LLMs are always of good quality. But for *informativeness* and *attributability*, the U-shape appears again: it shows that the LLMs (even the powerful GPT-3.5) are struggling to generate content specifically sticking to the document inserted in the middle.

In the second experiment, we take the opposite approach, and put salient information at the beginning and the end, while the middle of the prompt

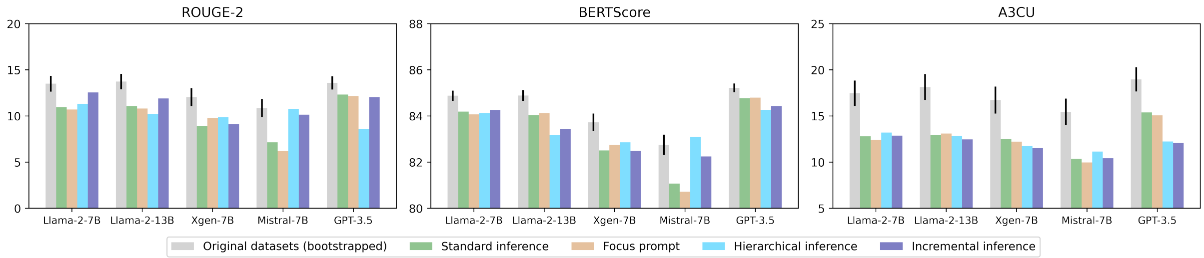

Figure 5: Reference-based evaluation on the MiddleSum dataset. We also report (gray bars) performance achieved by uniformly sampling subsets of the same size as MiddleSum from the original datasets, alongside bootstrapping variance (black lines).

is filled with noise. We keep the first and last documents, and fill the $k-2$ middle ones with random documents. We also run a baseline just using the first and last documents as input, expected to be close to the result with random documents in between. As displayed in Table 3, filling with random noise between the first and last document (which amounts to a prompt mostly irrelevant to the reference) leads to a moderate drop in performance. For instance, on Multi-XScience, with 5 random documents between the first and last, Llama-2-13B maintains 98% of its performance, and reaches a GPT-3.5 score of 4.31 as compared to 4.35 when using all 7 documents.

We conclude from these two experiments that ***LLMs can focus on the beginning and/or the end of their input, but largely ignore the middle. The U-shape or* middle curse *from Liu et al. (2023a) also applies to abstractive summarization.***

4.2 Can we alleviate the *middle curse*?

To evaluate the loss of performance due to the *middle curse* in a natural setup, we subsample data points from each of the 5 long-input summarization datasets. We obtain sentences from the (untruncated) source aligned with the reference summary, following the procedure from §3.2. Only data points where the start index of the earliest aligned source sentence is at least 1,200 words, are kept, ensuring no salient information at the start. We randomly sample 50 data points from each of Arxiv, PubMed, GovReport and Multi-News, and 25 from SummScreenFD, forming an evaluation dataset of 225 samples which we name **MiddleSum**.

We evaluate LLMs on MiddleSum, keeping only reference-based evaluation as the dataset is built using saliency with regard to the reference. As expected, in Fig. 5 we see that LLMs perform noticeably worse on MiddleSum (green bars) as compared to the full set (gray bars), confirming that MiddleSum is a more challenging task.

We benchmark alternative inference methods on MiddleSum: *hierarchical* summarization and *incremental* summarization, both of which are explored in the concurrent work of Chang et al. (2023). Namely, let us divide an input $\boldsymbol{x}$ of length n into k consecutive blocks of size at most m (yielding $k = \lceil \frac{n}{m} \rceil$): $\boldsymbol{x} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k)$.

Hierarchical summarization consists in summarizing each block and then summarizing the concatenation of summaries:

$$\boldsymbol{y}_i = \text{LLM}(\boldsymbol{x}_i) \quad \forall i \in \{1, \ldots, k\} \tag{1}$$
$$\boldsymbol{y} = \text{LLM}(\boldsymbol{y}_1, \ldots, \boldsymbol{y}_k) \tag{2}$$

Incremental summarization consists in updating a summary of the text so far with content from the current text block (we have $\boldsymbol{y}_0 = \emptyset$):

$$\boldsymbol{y}_i = \text{LLM}(\boldsymbol{y}_{i-1}, \boldsymbol{x}_i) \forall i \in \{1, \ldots, k\} \tag{3}$$
$$\boldsymbol{y} = \boldsymbol{y}_k \tag{4}$$

In both methods, the final output is $\boldsymbol{y}$. Noting l the output length, standard inference has complexity in $\mathcal{O}(l.n^2)$, while both alternative methods have complexity in $\mathcal{O}(k.l.m^2) = \mathcal{O}(l.n.m)$, which is lower. For both methods and all models, we use a block size m of 1,500 words (roughly 2,000 tokens), and preserve coherence by ending blocks at the earliest end of sentence reaching the length.

Results are shown in Fig. 5 (blue and purple bars), with detailed numbers in Appendix F. We also compare to a baseline consisting in adding the prompt `Please also pay attention to the middle section of the input when constructing the summary,` which we refer to as the *Focus prompt* (brown bars). Both alternative methods show promising results on open-source LLMs, notably on Mistral-7B for which they improve performance significantly. However, they are not successful and lag behind

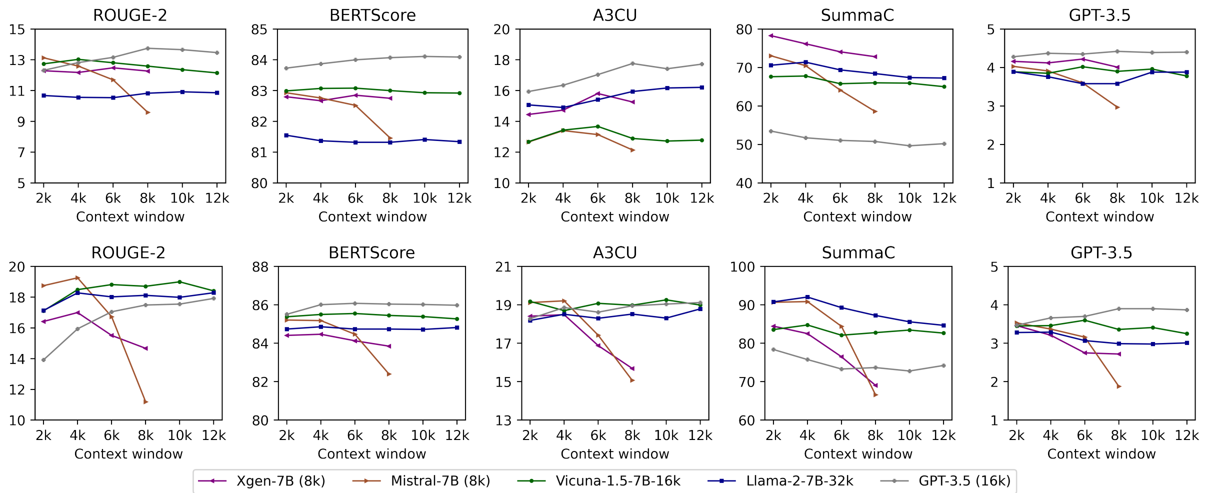

Figure 6: Long-input summarization performance on Arxiv (top) and GovReport (bottom) with 5 LLMs and all 5 metrics. X-axis represents the truncated maximum source length. Xgen-7B and Mistral-7B cannot infer beyond 8k tokens.

Focus Prompt with GPT-3.5. Across domains (see Table 8), *hierarchical* and *incremental* inference are very effective on scientific publications, which we hypothesize is due to the natural division in structured sections of such inputs. Yet, they seem to harm summaries on the other domains.

4.3 Is scaling context length *really* useful?

Experiments from §4.1 confirm that LLMs struggle to summarize information contained in the middle of their context window. This poses issues for long-input summarization: after the initial part with (usually) high saliency, important information becomes sparser, and at the same time LLMs processing capability weakens. To investigate this issue, we infer long-document summarization with length truncated at $m * 2k$ tokens, varying m from 1 to 6.[5] We use our longest context LLMs {Xgen-7B, Mistral-7B, GPT-3.5} ; as well as two open-source LLMs extending Llama-2-7B context window with **position interpolation** (Chen et al., 2023), a method gaining traction as an efficient way to scale LLMs' context window. We use Vicuna-7B-1.5-16k[6], and Llama-2-7B-32k[7], with context of 16k tokens and 32k tokens, respectively.

Results on Arxiv and GovReport in Fig. 6 confirm our intuition: all metrics plateau or even *decrease* (see Mistral-7B) from 4k context window upwards. Two conflicting forces are at play when increasing length: giving more information to the model helps it retrieve key elements further to make a richer summary, while at the same time reasoning over a longer context is more challenging. Yet, such a drop for Xgen-7B and Mistral-7B at 8k inference length is concerning. Both position interpolated models show more robustness ; while GPT-3.5 seems to plateau at 8k tokens. Our results suggest that ***in the current LLMs inference and evaluation framework, there is no need to exceed 4k tokens in the context window for open-source model.***

4.4 Does the decoding method impact the *middle curse*?

We now turn our attention to the process controlling summary generation. While we had sampled all summaries with **top-k** sampling with $k = 50$ and temperature $T = 0.3$ so far ; we now also experiment with **greedy** decoding, and **top-p** sampling (where we use $p = 0.95$ and temperature $T = 1.0$).

In Fig. 7, we reproduce the salient bigrams and sentences alignment experiments from §3.1 and §3.2, respectively, with the aforementioned decoding methods. As seen, the decoding method does not affect position bias: for all setups, the LLMs show similar patterns as with our previous default decoding method. We conclude that ***the middle curse is independent from the decoding method***.

5 Related Work

Summarization with LLMs It is widely acknowledged that LLMs have propelled forward abstractive summarization research (Pu et al., 2023), with their summaries being highly rated by human annotators (Goyal et al., 2022; Zhang et al.,

[5] Our hardware does not allow us to exceed 12k tokens.
[6] In HuggingFace: *lmsys/vicuna-7b-v1.5-16k*
[7] In HuggingFace: *togethercomputer/LLaMA-2-7B-32K*

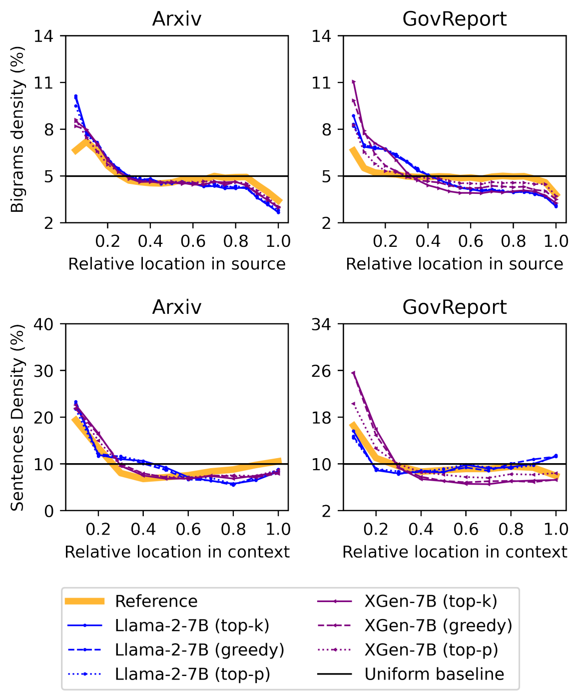

Figure 7: Summary bigrams (top) and aligned source sentences (bottom) distribution on Arxiv and GovReport for Llama-2-7B and XGen-7B, for several decoding strategies.

2023b). Liu et al. (2023d) proposes to train smaller models like BART (Lewis et al., 2020) or BRIO (Liu et al., 2022b) with contrastive learning using LLMs like ChatGPT as evaluator providing signal on which generated summary candidate is better. Summary chain-of-thought designs a custom chain-of-thought method which first prompts the LLM to list important facts, then integrates these facts into a coherent summary (Wang et al., 2023c). SummIt utilizes ChatGPT to iteratively write then refine summaries given feedback from an evaluator LLM (Zhang et al., 2023a). Chain-of-density gradually makes GPT-4 generated summaries contain more and more entities while keeping length budget constant, creating more informative albeit a bit less readable summaries (Adams et al., 2023). Ravaut et al. (2022) noticed that data points with higher compression are generally harder to summarize with pre-trained models.

Position bias in LLMs Sun et al. (2021) showed that for Transformer-based models, most recent tokens play a greater role compared to older tokens for next-token prediction. It was later found that for in-context learning, the order of examples within the prompt impacts GPT-3's performance (Liu et al., 2022a; Lu et al., 2021). Reliance on positional information affects LLMs capabilities in arithmetic (Shen et al., 2023b), in multiple-choice question-answering (Zheng et al., 2023; Pezeshkpour and Hruschka, 2023), and as text generation evaluators (Wang et al., 2023b) ; making it hard to rank LLMs (Alzahrani et al., 2024). Liu et al. (2023a) were the first to show that LLMs' performance weakens in the middle of the prompt (the **middle curse**), yet, how LLMs make use of their full context window remains poorly understood. The *passkey retrieval* evaluation, which consists in prompting the LLM to recall a complex string or long number inserted in its prompt, is becoming popular recently as a way of verifying LLM's processing capability at each position (Liu et al., 2023a; Jiang et al., 2024). However, this task does not measure position bias on complex, abstract reasoning tasks like summarization. A line of work attempts to solve the *middle curse* through compressing the prompt (Jiang et al., 2023b,c), with very promising results albeit at the cost of prompt fluency. Another approach marginalizes results over different permutations of the input to suppress dependency on input order (Tang et al., 2023). Concurrent work to ours also finds that in zero-shot summarization, LLMs tend to prefer lead content (Chhabra et al., 2024).

6 Conclusion

Behind the recent hype around LLMs and their amazing instruction following and content generation capabilities, our study showcases a major weakness in abstractive summarization: LLMs suffer from the *middle curse* and struggle to use information in the middle of their context window. LLMs do not make a consistent use of their context window as they mostly look at the beginning and (to a lesser extent) the end, which at first glance may be hidden by the prevalent *lead bias* in summarization datasets. Extending context window beyond 4k tokens, which has been an intense area of focus lately, is not justified in the current inference and evaluation setup in abstractive summarization. We benchmarked two alternative inference methods (*hierarchical* inference and *incremental* inference) on MiddleSum, an evaluation subset designed to showcase the *middle curse*. Despite encouraging results on scientific paper datasets, these methods are far from a silver bullet to the *middle curse*. We call for a better evaluation of LLMs, which accounts for the salient spans of the source which are effectively being processed.

Limitations

Our work only considers open-source LLMs for summary generation and ignores closed-source LLMs such as the popular OpenAI's GPT-3.5 and GPT-4, or Anthropic's Claude. We made this decision to advocate for openness in LLM research ; yet we acknowledge that it would be interesting to also investigate properties of summaries generated by these paying API LLMs.

Another limitation lays in the saliency estimation. We approximate salient content in the source through maximizing ROUGE-1 overlap with summary sentences. Other methods are also well-suited for this task, albeit at greater computational cost ; for instance semantic similarity through BERTScore or BARTScore ; or saliency estimation through a LLM in zero-shot.

Lastly, we can only evaluate a finite number of LLMs, and we settled for the evaluation of 5 recent and popular open-source LLMs. Findings may change as LLMs undergo changes and improvements in their training process.

Acknowledgements

This research was supported by the SINGA scholarship and partially supported by the National Research Foundation, Prime Minister's Office, Singapore under its Campus for Research Excellence and Technological Enterprise (CREATE) programme. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the author(s) and do not reflect the views of the National Research Foundation, Singapore.

We thank anonymous ARR Reviewers 1 and 3 for their great review, which have led us to include plenty more important results enriching our analysis. We also thank the AC for their constructive meta-review. We thank Florian Le Bronnec and Ruochen Zhao for their helpful proof-reading of the draft.

References

Griffin Adams, Alexander Fabbri, Faisal Ladhak, Eric Lehman, and Noémie Elhadad. 2023. From sparse to dense: Gpt-4 summarization with chain of density prompting. *arXiv preprint arXiv:2309.04269*.

Norah Alzahrani, Hisham Abdullah Alyahya, Yazeed Alnumay, Sultan Alrashed, Shaykhah Alsubaie, Yusef Almushaykeh, Faisal Mirza, Nouf Alotaibi, Nora Altwairesh, Areeb Alowisheq, et al. 2024. When benchmarks are targets: Revealing the sensitivity of large language model leaderboards. *arXiv preprint arXiv:2402.01781*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Yapei Chang, Kyle Lo, Tanya Goyal, and Mohit Iyyer. 2023. Booookscore: A systematic exploration of book-length summarization in the era of llms. *arXiv preprint arXiv:2310.00785*.

Mingda Chen, Zewei Chu, Sam Wiseman, and Kevin Gimpel. 2022. SummScreen: A dataset for abstractive screenplay summarization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8602–8615, Dublin, Ireland. Association for Computational Linguistics.

Shouyuan Chen, Sherman Wong, Liangjian Chen, and Yuandong Tian. 2023. Extending context window of large language models via positional interpolation. *arXiv preprint arXiv:2306.15595*.

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. *arXiv preprint arXiv:2401.01989*.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.

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.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Alexander Fabbri, Irene Li, Tianwei She, Suyi Li, and Dragomir Radev. 2019. Multi-news: A large-scale multi-document summarization dataset and abstractive hierarchical model. In *Proceedings of the 57th*

Annual Meeting of the Association for Computational Linguistics, pages 1074–1084, Florence, Italy. Association for Computational Linguistics.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 889–898, Melbourne, Australia. Association for Computational Linguistics.

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

Bogdan Gliwa, Iwona Mochol, Maciej Biesek, and Aleksander Wawer. 2019. SAMSum corpus: A human-annotated dialogue dataset for abstractive summarization. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pages 70–79, Hong Kong, China. Association for Computational Linguistics.

Tanya Goyal, Junyi Jessy Li, and Greg Durrett. 2022. News summarization and evaluation in the era of gpt-3. *arXiv preprint arXiv:2209.12356*.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. *Advances in neural information processing systems*, 28.

Luyang Huang, Shuyang Cao, Nikolaus Parulian, Heng Ji, and Lu Wang. 2021. Efficient attentions for long document summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1419–1436, Online. Association for Computational Linguistics.

Sameer Jain, Vaishakh Keshava, Swarnashree Mysore Sathyendra, Patrick Fernandes, Pengfei Liu, Graham Neubig, and Chunting Zhou. 2023. Multi-dimensional evaluation of text summarization with in-context learning. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8487–8495, Toronto, Canada. Association for Computational Linguistics.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023a. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

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

Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. 2023b. Llmlingua: Compressing prompts for accelerated inference of large language models. *arXiv preprint arXiv:2310.05736*.

Huiqiang Jiang, Qianhui Wu, Xufang Luo, Dongsheng Li, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. 2023c. Longllmlingua: Accelerating and enhancing llms in long context scenarios via prompt compression. *arXiv preprint arXiv:2310.06839*.

Byeongchang Kim, Hyunwoo Kim, and Gunhee Kim. 2019. Abstractive summarization of Reddit posts with multi-level memory networks. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2519–2531, Minneapolis, Minnesota. 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.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. 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.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2022a. What makes good in-context examples for GPT-3? In *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 100–114, Dublin, Ireland and Online. 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. *arXiv preprint arXiv:2307.03172*.

Yixin Liu, Alex Fabbri, Pengfei Liu, Yilun Zhao, Linyong Nan, Ruilin Han, Simeng Han, Shafiq Joty, Chien-Sheng Wu, Caiming Xiong, and Dragomir Radev. 2023b. Revisiting the gold standard: Grounding summarization evaluation with robust human evaluation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4140–4170, Toronto, Canada. Association for Computational Linguistics.

Yixin Liu, Alexander Fabbri, Yilun Zhao, Pengfei Liu, Shafiq Joty, Chien-Sheng Wu, Caiming Xiong, and Dragomir Radev. 2023c. Towards interpretable and efficient automatic reference-based summarization evaluation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 16360–16368, Singapore. Association for Computational Linguistics.

Yixin Liu, Alexander R Fabbri, Pengfei Liu, Dragomir Radev, and Arman Cohan. 2023d. On learning to summarize with large language models as references. *arXiv preprint arXiv:2305.14239*.

Yixin Liu, Pengfei Liu, Dragomir Radev, and Graham Neubig. 2022b. BRIO: Bringing order to abstractive summarization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2890–2903, Dublin, Ireland. Association for Computational Linguistics.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2021. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. *arXiv preprint arXiv:2104.08786*.

Yao Lu, Yue Dong, and Laurent Charlin. 2020. Multi-XScience: A large-scale dataset for extreme multi-document summarization of scientific articles. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8068–8074, Online. Association for Computational Linguistics.

Zheheng Luo, Qianqian Xie, and Sophia Ananiadou. 2023. Chatgpt as a factual inconsistency evaluator for text summarization.

Frank J Massey Jr. 1951. The kolmogorov-smirnov test for goodness of fit. *Journal of the American statistical Association*, 46(253):68–78.

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.

Erik Nijkamp, Tian Xie, Hiroaki Hayashi, Bo Pang, Congying Xia, Chen Xing, Jesse Vig, Semih Yavuz, Philippe Laban, Ben Krause, et al. 2023. Xgen-7b technical report. *arXiv preprint arXiv:2309.03450*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Bo Pang, Erik Nijkamp, Wojciech Kryściński, Silvio Savarese, Yingbo Zhou, and Caiming Xiong. 2022. Long document summarization with top-down and bottom-up inference. *arXiv preprint arXiv:2203.07586*.

Guilherme Penedo, Quentin Malartic, Daniel Hesslow, Ruxandra Cojocaru, Alessandro Cappelli, Hamza Alobeidli, Baptiste Pannier, Ebtesam Almazrouei, and Julien Launay. 2023. The refinedweb dataset for falcon llm: outperforming curated corpora with web data, and web data only. *arXiv preprint arXiv:2306.01116*.

Bowen Peng, Jeffrey Quesnelle, Honglu Fan, and Enrico Shippole. 2023. Yarn: Efficient context window extension of large language models. *arXiv preprint arXiv:2309.00071*.

Pouya Pezeshkpour and Estevam Hruschka. 2023. Large language models sensitivity to the order of options in multiple-choice questions. *arXiv preprint arXiv:2308.11483*.

Ofir Press, Noah A Smith, and Mike Lewis. 2021. Train short, test long: Attention with linear biases enables input length extrapolation. *arXiv preprint arXiv:2108.12409*.

Xiao Pu, Mingqi Gao, and Xiaojun Wan. 2023. Summarization is (almost) dead. *arXiv preprint arXiv:2309.09558*.

Mathieu Ravaut, Shafiq Joty, and Nancy Chen. 2022. Towards summary candidates fusion. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8488–8504, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, et al. 2022. Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada. Association for Computational Linguistics.

Uri Shaham, Maor Ivgi, Avia Efrat, Jonathan Berant, and Omer Levy. 2023. Zeroscrolls: A zero-shot benchmark for long text understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 7977–7989, Singapore. Association for Computational Linguistics.

Uri Shaham, Elad Segal, Maor Ivgi, Avia Efrat, Ori Yoran, Adi Haviv, Ankit Gupta, Wenhan Xiong, Mor Geva, Jonathan Berant, and Omer Levy. 2022. SCROLLS: Standardized CompaRison over long language sequences. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 12007–12021, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Chenhui Shen, Liying Cheng, Yang You, and Lidong Bing. 2023a. Are large language models good evaluators for abstractive summarization? *arXiv preprint arXiv:2305.13091*.

Ruoqi Shen, Sébastien Bubeck, Ronen Eldan, Yin Tat Lee, Yuanzhi Li, and Yi Zhang. 2023b. Positional description matters for transformers arithmetic. *arXiv preprint arXiv:2311.14737*.

Simeng Sun, Kalpesh Krishna, Andrew Mattarella-Micke, and Mohit Iyyer. 2021. Do long-range language models actually use long-range context? In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 807–822, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yutao Sun, Li Dong, Barun Patra, Shuming Ma, Shaohan Huang, Alon Benhaim, Vishrav Chaudhary, Xia Song, and Furu Wei. 2022. A length-extrapolatable transformer. *arXiv preprint arXiv:2212.10554*.

Raphael Tang, Xinyu Zhang, Xueguang Ma, Jimmy Lin, and Ferhan Ture. 2023. Found in the middle: Permutation self-consistency improves listwise ranking in large language models. *arXiv preprint arXiv:2310.07712*.

Yi Tay, Mostafa Dehghani, Vinh Q Tran, Xavier Garcia, Jason Wei, Xuezhi Wang, Hyung Won Chung, Dara Bahri, Tal Schuster, Steven Zheng, et al. 2022. Ul2: Unifying language learning paradigms. In *The Eleventh International Conference on Learning Representations*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023a. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

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

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

Peiyi Wang, Lei Li, Liang Chen, Dawei Zhu, Binghuai Lin, Yunbo Cao, Qi Liu, Tianyu Liu, and Zhifang Sui. 2023b. Large language models are not fair evaluators. *arXiv preprint arXiv:2305.17926*.

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.

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Wen Xiao, Iz Beltagy, Giuseppe Carenini, and Arman Cohan. 2022. PRIMERA: Pyramid-based masked sentence pre-training for multi-document summarization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5245–5263, Dublin, Ireland. Association for Computational Linguistics.

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.

Wenhan Xiong, Anchit Gupta, Shubham Toshniwal, Yashar Mehdad, and Scott Yih. 2023. Adapting pretrained text-to-text models for long text sequences. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5566–5578, Singapore. Association for Computational Linguistics.

Haopeng Zhang, Xiao Liu, and Jiawei Zhang. 2023a. Summit: Iterative text summarization via chatgpt. *arXiv preprint arXiv:2305.14835*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

Tianyi Zhang, Faisal Ladhak, Esin Durmus, Percy Liang, Kathleen McKeown, and Tatsunori B Hashimoto. 2023b. Benchmarking large language models for news summarization. *arXiv preprint arXiv:2301.13848*.

Chao Zhao, Faeze Brahman, Kaiqiang Song, Wenlin Yao, Dian Yu, and Snigdha Chaturvedi. 2022a. NarraSum: A large-scale dataset for abstractive narrative summarization. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages

182–197, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Yao Zhao, Misha Khalman, Rishabh Joshi, Shashi Narayan, Mohammad Saleh, and Peter J Liu. 2022b. Calibrating sequence likelihood improves conditional language generation. *arXiv preprint arXiv:2210.00045*.

Chujie Zheng, Hao Zhou, Fandong Meng, Jie Zhou, and Minlie Huang. 2023. Large language models are not robust multiple choice selectors. *arXiv e-prints*, pages arXiv–2309.

Qingyu Zhou, Nan Yang, Furu Wei, Shaohan Huang, Ming Zhou, and Tiejun Zhao. 2018. Neural document summarization by jointly learning to score and select sentences. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 654–663, Melbourne, Australia. Association for Computational Linguistics.

A Statistics

In Table 4, we include statistics on each of the abstractive summarization datasets under consideration. We use the non-anonymized version for CNN/DM (See et al., 2017). For Reddit-TIFU, we use the Long subset, and SummScreenFD, is the ForeverDreaming (FD) subset of SummScreen. GovReport and SummScreenFD are part of the long-input benchmarks Scrolls (Shaham et al., 2022) and ZeroScrolls (Shaham et al., 2023).

In Fig. 8, we illustrate how much of the source document(s) is visible with a 4k context window (Llama-2).

B GPT-3.5 Evaluation

To evaluate LLM-generated summaries with GPT-3.5, we use the following prompt template:

Score the following summary generated by another system given the source on a scale from 1 to 5 with regards to overall general summary quality. 1-point indicates a low quality summary, and 5 points a very high quality summary. A high quality summary is grammatical, fluent, informative, relevant, coherent and factually consistent with the source. Let's think step-by-step and just output the score.

Source:
[source]

Instruction:
Summarize the above text in [n] sentences.

Summary:
[summary generated by the LLM to evaluate]

Your score:

When evaluating for the specific aspect of ***informativeness***, the first paragraph becomes:

Score the following summary generated by another system given the source on a scale from 1 to 5 with regards to how informative the summary is. 1 point indicates a not informative summary, and 5 points a very informative summary. An informative summary captures the important information in the article and presents it accurately and concisely. Let's think step-by-step and just output the score.

When evaluating for the specific aspect of ***overall quality***, the first paragraph becomes:

Score the following summary generated by another system given the source on a scale from 1 to 5 with regards to its quality. 1 point indicates a low quality summary, and 5 points a very high quality summary. A high quality summary is comprehensible and understandable. Let's think step-by-step and just output the score.

When evaluating for the specific aspect of ***coherence***, the first paragraph becomes:

Score the following summary generated by another system given the source on a scale from 1 to 5 with regards to its coherence. 1 point indicates an incoherent summary, and 5 points a very coherent summary. A coherent summary is well-structured and well-organized. Let's think step-by-step and just output the score.

When evaluating for the specific aspect of ***attributability***, the first paragraph becomes:

Score the following summary generated by another system given the source on a scale from 1 to 5 with regards to how attributable it is. 1 point indicates a not very attributable summary, with many hallucinations, and 5 points a summary very attributable to the source, consistent with the source. In a very attributable summary, all the informa-

Dataset	Domain	# Docs	# Data points			# Sentences			# Words		# Tokens		
			Train	Val	Test	Doc.	Summ.	Inf.	Doc.	Summ.	Doc.	Summ.	Max gen.
CNN/DM (Hermann et al., 2015)	News	1.00	287,113	13,334	11,490	33.37	3.79	3	773.23	57.75	994.56	84.47	192
XSum (Narayan et al., 2018)	News	1.00	204,045	11,332	11,334	19.11	1.00	1	433.05	23.19	566.79	31.63	64
Reddit-TIFU (Long) (Kim et al., 2019)	Social Media	1.00	33,701	4,214	4,221	22.21	1.45	2	444.20	23.37	532.18	29.82	128
SAMSum (Gliwa et al., 2019)	Dialogue	1.00	14,732	818	819	8.96	2.01	2	126.93	23.12	175.54	29.69	128
Multi-XScience (Lu et al., 2020)	Science	5.06	30,369	5,066	5,093	30.55	4.86	5	773.36	120.65	965.99	157.77	384
Arxiv (Cohan et al., 2018)	Science	1.00	203,037	6,436	6,440	250.37	6.23	6	6,446.11	166.72	8,940.00	225.58	512
PubMed (Cohan et al., 2018)	Science (medical)	1.00	119,924	6,633	6,658	101.61	7.59	7	3,142.92	208.03	4,602.62	324.97	512
GovReport (Huang et al., 2021)	Legal	1.00	17,517	973	973	282.86	23.14	22	8,363.22	649.01	11,025.02	879.10	768
SummScreenFD (Chen et al., 2022)	TV Transcripts	1.00	3,673	338	337	727.06	5.26	5	7,618.20	123.34	10,067.39	157.44	512
Multi-News (Fabbri et al., 2019)	News	2.73	44,972	5,622	5,622	79.02	9.88	10	2,101.49	256.55	2,998.52	324.29	512

Table 4: Statistics on the 10 datasets used for experiments. **Doc.** is the source document, **Summ.** the ground-truth summary, **Inf.** refers to the number of desired sentences to be in the summary prompted to each LLM during inference. Statistics are computed on the entire test set. **# Tokens** is calculated with Llama-2's tokenizer. **Max gen.** is the maximum tokens size that we set when decoding summaries. Underlined test sizes correspond to datasets where we subsample randomly 1,000 test data points for evaluation.

Model	Metric	CNN/DM	XSum	Reddit-TIFU	SAMSum	Multi-X	AVG	Arxiv	PubMed	GovReport	SummScreenFD	Multi-N	AVG
SOTA	ROUGE-2	*24.17*	*27.09*	*11.13*	*29.88*	*4.60*	*19.37*	*21.93*	*23.26*	*30.90*	*10.70*	*13.60*	*20.07*
Flan-UL2	# sents	2.89	1.00	1.34	2.08	2.28	1.19	1.78	1.27	1.80	5.49	2.79	2.63
	ROUGE-2	**20.28**	**22.74**	**8.61**	**28.21**	3.04	**16.58**	9.37	7.42	4.76	4.33	7.79	6.73
	BERTScore	**88.05**	**91.94**	**87.42**	**92.29**	81.87	**88.31**	83.82	83.07	83.53	84.85	84.97	84.05
	A3CU	**32.69**	**32.11**	**16.89**	**49.48**	5.98	**27.43**	14.79	13.83	12.00	8.37	16.99	13.20
	SummaC	**69.96**	24.27	**35.76**	**30.19**	57.98	**43.63**	**67.56**	60.96	73.80	**56.00**	**76.80**	**67.02**
	GPT-3.5	3.16	3.52	1.61	2.92	3.23	2.89	3.11	3.16	2.46	2.11	3.36	2.84
Llama-2-7B	# sents	3.00	1.27	2.00	1.83	7.77	3.17	5.80	6.61	12.88	5.77	18.69	9.95
	ROUGE-2	14.16	7.27	4.17	15.53	4.87	9.20	**13.84**	12.89	16.22	5.36	12.37	12.14
	BERTScore	87.32	87.47	85.87	89.95	83.32	86.79	83.84	82.82	85.28	85.41	85.63	84.60
	A3CU	29.04	14.18	12.15	35.64	6.39	19.48	16.78	16.60	17.23	9.66	22.23	16.50
	SummaC	35.58	25.24	26.38	24.56	49.39	32.23	53.22	51.82	70.47	39.01	57.49	54.40
	GPT-3.5	4.10	4.24	2.83	3.61	4.42	3.84	4.21	4.19	3.43	2.71	3.91	3.69
Llama-2-13B	# sents	3.01	1.16	2.00	1.98	5.22	2.67	5.92	7.22	27.75	5.16	12.79	11.77
	ROUGE-2	14.10	8.61	4.22	14.16	5.29	9.28	13.52	**15.24**	17.28	5.62	12.58	12.85
	BERTScore	87.40	87.94	85.85	89.45	**83.58**	86.84	83.88	84.24	85.29	85.42	85.84	84.93
	A3CU	29.57	16.30	12.94	34.19	7.29	20.06	16.44	**19.21**	17.01	10.30	**23.09**	17.21
	SummaC	33.83	24.07	25.76	24.81	41.70	30.03	55.09	56.00	**76.44**	38.74	53.12	55.88
	GPT-3.5	**4.12**	4.34	2.91	3.45	4.45	3.85	4.06	4.13	3.66	2.69	3.89	3.69
Xgen-7B	# sents	3.93	2.24	2.62	2.34	5.94	3.41	8.07	13.50	22.46	10.60	9.05	12.74
	ROUGE-2	14.55	6.00	3.98	14.51	5.54	8.92	12.31	13.99	14.68	4.24	11.07	11.26
	BERTScore	87.07	87.12	85.84	89.53	83.42	86.60	83.07	82.87	83.94	83.91	84.95	83.75
	A3CU	27.88	12.75	12.56	33.18	7.45	18.76	15.28	18.79	15.65	8.44	21.33	18.90
	SummaC	52.25	**37.95**	28.40	25.63	44.36	37.72	53.28	60.56	65.22	42.03	56.45	55.51
	GPT-3.5	3.82	3.97	2.78	3.52	4.37	3.69	3.96	3.99	2.78	2.42	3.58	3.35
Mistral-7B	# sents	3.10	1.12	2.64	2.25	7.73	3.37	12.00	11.88	25.83	27.13	16.43	18.65
	ROUGE-2	16.37	7.05	4.34	14.66	**5.57**	9.60	9.77	14.32	11.36	3.11	**12.61**	10.23
	BERTScore	87.50	87.45	85.71	89.78	83.16	86.72	81.44	82.85	82.43	81.46	85.07	82.65
	A3CU	30.60	13.14	13.08	32.21	6.96	19.20	12.66	16.31	14.92	8.40	22.22	14.90
	SummaC	53.67	24.79	30.51	26.82	**62.49**	39.66	57.81	**69.03**	67.07	35.76	68.50	59.63
	GPT-3.5	3.92	4.30	2.73	3.63	4.47	3.81	2.83	3.63	2.01	1.88	3.60	2.79
GPT-3.5	# sents	3.00	1.00	2.00	1.98	4.99	2.59	5.60	6.46	15.82	5.01	9.27	8.43
	ROUGE-2	13.17	8.19	5.17	14.47	5.37	9.27	13.78	13.90	**17.94**	**6.55**	12.28	**12.89**
	BERTScore	87.26	87.90	86.38	89.79	83.82	87.03	**84.14**	**84.29**	**85.99**	**86.13**	**86.02**	**85.31**
	A3CU	26.72	15.21	13.37	35.85	**7.90**	19.81	**17.98**	18.76	**19.32**	**12.69**	22.95	**18.34**
	SummaC	35.14	23.22	26.00	24.92	37.56	29.37	50.11	47.36	73.23	40.58	48.91	52.04
	GPT-3.5	**4.12**	**4.58**	**3.08**	**3.81**	**4.61**	**4.04**	**4.41**	**4.36**	**4.03**	**3.39**	**4.14**	**4.07**

Table 5: Performance achieved by the LLMs on each dataset for all 5 metrics. **# sents** is the average number of sentences in generated summaries. **Multi-X** is short for Multi-XScience, **Multi-N** is Multi-News dataset, **AVG** columns represent the average over standard-length and long-input datasets, respectively. SOTA numbers are taken from (Xie et al., 2023) on CNN/DM, from (Zhao et al., 2022b) XSum, Reddit-TIFU and SAMSum, from (Pang et al., 2022) for Arxiv and PubMed, from (Xiong et al., 2023) for GovReport and SummScreenFD, from (Xiao et al., 2022) for Multi-News and Multi-XScience. Due to a lack of reported results for other metrics, we only include ROUGE-2 scores for SOTA models. Best scores (outside of SOTA) are in bold.

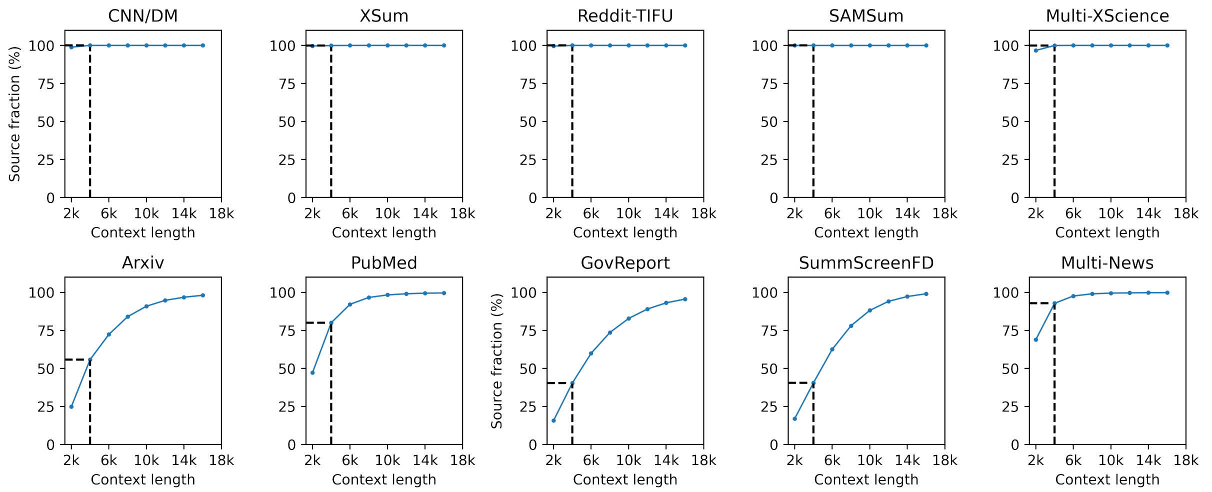

Figure 8: Fraction of the source which fits into the context window, for several context window lengths with Llama-2 tokenization. The black dashed lines correspond to Llama-2 context window length of 4k tokens. On standard-length datasets, 4k is enough to access 100% of all source documents ; but on the long-input datasets such as GovReport or SummScreenFD, such a context window may not even fit 50% of the source.

Model	CNN/DM	XSum	Reddit-TIFU	SAMSum	Multi-XScience	Arxiv	PubMed	GovReport	SummScreenFD	Multi-News
Flan-UL2	0.000	0.012	0.000	0.057	0.000	–	–	–	–	–
Llama-2-7B	0.000	0.000	0.000	0.490	0.000	0.000	0.000	0.000	0.000	0.000
Llama-2-13B	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Xgen-7B	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Mistral-7B	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
GPT-3.5	0.000	0.000	0.000	0.135	0.000	0.000	0.000	0.000	0.000	0.000

Table 6: P-value of a 2-sample Kolmogorov-Smirnov test between the position distribution of bigrams in LLM-generated summaries and bigrams in reference summaries. We round numbers to 3 decimals. Numbers in gray correspond to non-significant differences (p-value above 0.01).

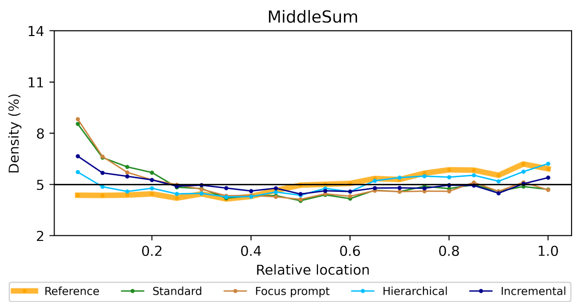

Figure 9: Relative location of summary bigrams within the source on MiddleSum for Llama-2-13B with different inference methods.

tion is fully attributable to the source. Let's think step-by-step and just output the score.

C Baseline Performance

In Table 5, we report zero-shot performance with the 6 LLMs described in §2. We note that for standard-length datasets, Flan-UL2 is dominating, perhaps due to its better instruction-tuning ; while for long-context ones, GPT-3.5 takes the lead. GPT-3.5 is always consistently ahead on the GPT-3.5 metric, showing the preference of the LLM for its own generation when used as an evaluator.

D Statistical Significance on RQ1

In Table 6, we run a 2-sample Kolmogorov-Smirnov statistical significance test to compare the bigrams position distribution of LLMs with the reference summaries.

E Results with *base* Models

In Fig. 10, we repeat the analysis from §3.1 (bigrams alignment) and §3.2 (sentences alignment) but this time using the *base* model versions (not instruction-tuned) of Llama-2-7B and Llama-2-13B. As we can see, the position bias is even stronger for base models than for chatbot models, suggesting that this bias is acquired during pre-training.

F More Results on MiddleSum

In Table 7, we show statistics on the MiddleSum evaluation dataset.

Dataset	Domain	# Docs	# Data points	# Sentences		# Words		# Tokens	
				Doc.	Summ.	Doc.	Summ.	Doc.	Summ.
Arxiv	Science	1.00	50	299.36	5.84	7,605.52	165.80	10,846.00	226.22
PubMed	Science (medical)	1.00	50	157.60	6.96	5,090.84	203.04	7,783.02	329.12
GovReport	Legal	1.00	50	445.64	22.80	13,308.70	656.10	17,462.90	883.10
SummScreen	TV transcripts	1.00	25	762.24	3.56	9,732.44	88.60	12,878.08	115.00
MultiNews	News	3.18	50	211.34	9.86	5,939.32	269.84	8,130.78	336.16
Overall	Mixed	1.48	225	332.24	10.50	8,180.13	297.57	11,258.16	407.15

Table 7: Statistics on the MiddleSum evaluation dataset, breaking down on each domain. **Doc.** is the source document, **Summ.** the ground-truth summary. **# Tokens** is calculated with Llama-2's tokenizer.

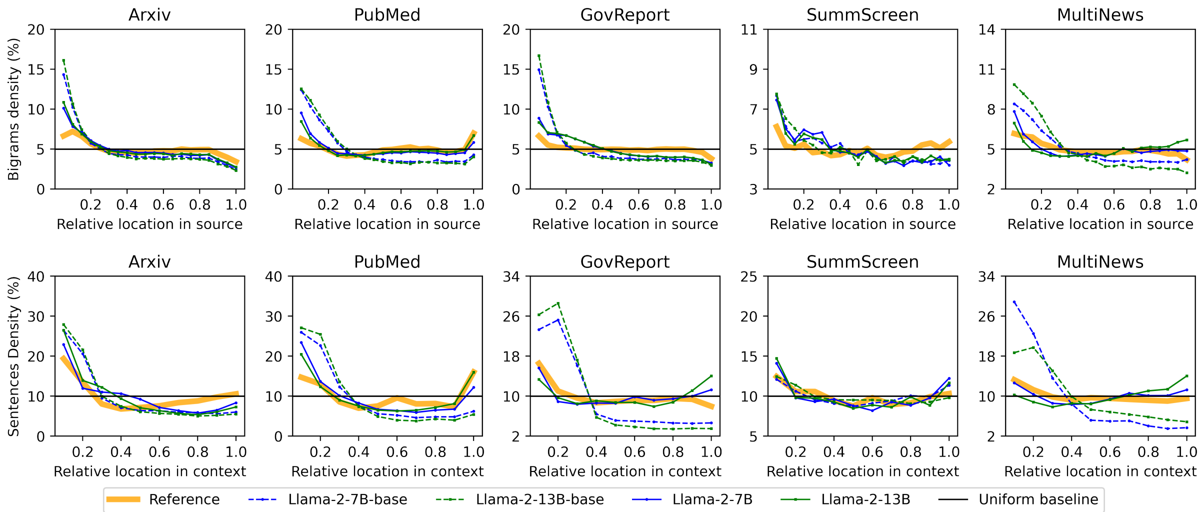

Figure 10: Summary bigrams (top) and aligned source sentences (bottom) distribution for Llama-2-7B and Llama-2-13B on the long-input datasets, both with *base* (dashed lines) and *chat* (full lines) models.

In Fig. 9, we repeat the salient bigrams analysis from Fig. 1 for Llama-2-13B on MiddleSum. We note that both *hierarchical* and *incremental* inference notably decrease reliance on lead bigrams compared to standard inference, while the simple *Focus Prompt* does not.

In Table 8, we show reference-based evaluation on each of the 5 subsets of MiddleSum.

G Software

We use the following Python libraries, all open source:

- *numpy*, version 1.24.3
- *torch*, version 2.0.1
- *scikit-learn*, version 1.0.2
- *sentencepiece*, version 0.1.97
- *nltk*, version 3.8.1
- *spacy*, version 3.6.0
- *scipy*, version 1.10.1
- *rouge-score*, version 0.1.2
- *bert-score*, version 0.3.13
- *summac*, version 0.0.03
- *tiktoken*, version 0.4.0
- *openai*, version 0.28.0
- *huggingface-hub*, version 0.17.2
- *datasets*, version 2.14.5
- *accelerate*, version 0.21.0
- *tokenizers*, version 0.14.1
- *transformers*, version 4.34.0

Model	Metric	Inference	MiddleSum (MS)	MS/Arxiv	MS/PubMed	MS/GovReport	MS/SummScreen	MS/Multi-News
Llama-2-7B	ROUGE-2	Standard	10.96	12.62	10.97	13.26	4.07	**10.43**
		Focus prompt	10.70	12.10	11.51	13.28	3.57	9.48
		Hierarchical	11.33	**14.63**	13.36	13.51	4.85	7.06
		Incremental	**12.56**	14.43	**13.54**	**17.26**	**5.19**	8.68
	BERTScore	Standard	84.19	83.43	83.13	84.63	85.33	**85.01**
		Focus prompt	84.07	83.33	83.00	**84.77**	85.27	84.59
		Hierarchical	84.13	**83.86**	83.74	84.26	85.23	84.10
		Incremental	**84.26**	83.73	**83.80**	84.67	**85.70**	84.14
	A3CU	Standard	12.81	13.55	12.12	11.71	7.31	**16.61**
		Focus prompt	12.41	13.54	13.39	10.84	7.15	14.52
		Hierarchical	**13.21**	**15.64**	**15.14**	**12.26**	9.71	11.57
		Incremental	12.88	15.34	14.45	10.51	**10.24**	12.55
Llama-2-13B	ROUGE-2	Standard	11.07	11.68	11.63	13.56	**5.09**	10.38
		Focus prompt	10.82	11.51	11.30	12.42	4.83	**11.03**
		Hierarchical	10.24	**13.45**	13.26	10.21	4.93	6.71
		Incremental	**11.90**	12.53	**13.84**	**17.34**	5.06	7.33
	BERTScore	Standard	84.04	83.29	83.15	84.43	**85.24**	84.70
		Focus prompt	**84.12**	83.26	83.06	84.49	85.08	**85.19**
		Hierarchical	83.17	**83.74**	**83.75**	80.07	85.03	84.20
		Incremental	83.43	82.45	83.45	**84.96**	83.24	82.95
	A3CU	Standard	12.94	13.01	13.42	10.92	8.47	16.65
		Focus prompt	**13.10**	13.17	11.76	11.91	7.06	**18.58**
		Hierarchical	12.85	**16.28**	14.11	10.40	8.36	12.86
		Incremental	12.47	14.64	**14.80**	**11.75**	**10.76**	9.54
Xgen-7B	ROUGE-2	Standard	8.92	11.94	9.34	10.02	**4.53**	6.57
		Focus prompt	9.79	12.87	10.79	11.46	4.15	**6.85**
		Hierarchical	**9.87**	**13.06**	**11.37**	11.77	2.79	6.83
		Incremental	9.11	10.97	10.55	**13.16**	3.71	4.44
	BERTScore	Standard	82.51	82.45	81.96	82.65	83.28	82.60
		Focus prompt	82.75	82.45	82.14	83.08	**83.81**	82.80
		Hierarchical	**82.86**	**82.73**	**82.56**	82.67	83.79	**83.03**
		Incremental	82.49	81.94	81.82	**83.11**	83.39	80.83
	A3CU	Standard	**12.50**	15.14	12.85	**11.99**	8.48	**12.05**
		Focus prompt	12.23	**15.82**	12.04	11.57	8.20	11.50
		Hierarchical	11.74	14.55	12.61	10.76	7.33	11.25
		Incremental	11.51	12.73	**13.15**	9.13	**8.97**	9.69
Mistral-7B	ROUGE-2	Standard	7.16	9.28	9.56	5.86	2.19	6.41
		Focus prompt	6.20	7.62	8.44	4.85	1.42	6.29
		Hierarchical	**10.78**	**11.67**	**13.71**	**13.36**	**4.76**	**7.39**
		Incremental	10.15	11.36	12.66	12.89	3.46	7.02
	BERTScore	Standard	81.07	80.98	81.19	80.41	80.10	82.20
		Focus prompt	80.72	80.31	80.81	79.91	80.12	82.13
		Hierarchical	**83.10**	**82.56**	**83.00**	**83.81**	**83.90**	**82.62**
		Incremental	82.25	81.92	82.39	83.67	83.14	80.56
	A3CU	Standard	10.35	11.55	11.46	7.94	**8.53**	11.34
		Focus prompt	9.96	10.13	10.65	7.81	7.59	**12.44**
		Hierarchical	**11.15**	12.15	**14.21**	**10.62**	7.48	9.47
		Incremental	10.43	**14.25**	11.00	8.90	7.44	9.05
GPT-3.5	ROUGE-2	Standard	**12.33**	**12.99**	**12.78**	16.16	6.05	**10.53**
		Focus prompt	12.17	12.61	12.56	15.95	**6.35**	10.46
		Hierarchical	8.59	9.58	10.27	10.30	4.33	6.36
		Incremental	12.04	11.85	12.17	**18.46**	5.38	9.03
	BERTScore	Standard	84.77	**83.81**	83.70	85.41	**86.32**	**85.39**
		Focus prompt	**84.80**	83.76	**83.81**	**85.50**	86.28	**85.39**
		Hierarchical	84.27	83.34	83.62	85.30	85.37	84.28
		Incremental	84.43	83.48	83.74	85.41	85.76	84.44
	A3CU	Standard	**15.40**	**15.22**	15.58	12.72	13.15	**19.21**
		Focus prompt	15.07	14.51	**15.59**	12.34	**13.63**	18.54
		Hierarchical	12.24	12.54	13.01	**13.29**	9.03	11.74
		Incremental	12.08	13.59	13.24	10.90	9.05	12.08

Table 8: Reference-based results for all models and inference methods on MiddleSum, breaking down by subset. The best number across inference methods is in bold.