Multi-LLM Text Summarization

Jiangnan Fang[1], Cheng-Tse Liu[1], Jieun Kim[1], Yash Bhedaru[1], Ethan Liu[1], Nikhil Singh[1],
Nedim Lipka[2], Puneet Mathur[2], Nesreen K. Ahmed[2], Franck Dernoncourt[2],
Ryan A. Rossi[2], and Hanieh Deilamsalehy[2]

[1]University of California, Santa Cruz
[2]Adobe Research

Abstract

In this work, we propose a Multi-LLM summarization framework, and investigate two different multi-LLM strategies including centralized and decentralized. Our multi-LLM summarization framework has two fundamentally important steps at each round of conversation: generation and evaluation. These steps are different depending on whether our multi-LLM decentralized summarization is used or centralized. In both our multi-LLM decentralized and centralized strategies, we have k different LLMs that generate diverse summaries of the text. However, during evaluation, our multi-LLM centralized summarization approach leverages a single LLM to evaluate the summaries and select the best one whereas k LLMs are used for decentralized multi-LLM summarization. Overall, we find that our multi-LLM summarization approaches significantly outperform the baselines that leverage only a single LLM by up to 3x. These results indicate the effectiveness of multi-LLM approaches for summarization.

1 Introduction

Large language models (LLMs) have been shown to have the potential to produce high-quality summaries (Chowdhery et al., 2022; Zhang et al., 2023; Goyal et al., 2023; Pu et al., 2023b). However, despite the remarkable progress in LLM-based summarization, limitations still exist for documents where useful information may be sparsely distributed throughout the text. Research by (Liu et al., 2023) highlights that a naive application of LLMs may overlook critical details or fail to grasp the holistic meaning of a document, indicating the need for more refined methods.

To address this, recent efforts have explored prompt-engineering techniques to guide LLMs towards producing better summaries (Adams et al., 2023). These techniques, while promising, still face limitations in consistently delivering high-quality summaries across different document types and structures. Instead of relying solely on a single model or simple prompt-engineering methods, we propose an approach novel to the summarization domain that focuses on aggregating the collective strengths of multiple LLMs. By combining the capabilities of multiple models with a diverse set of knowledge bases, we show it's possible to achieve more robust summaries across domains.

Summary of Main Contributions. The main contributions of this work are as follows:

- We propose the first framework for multi-LLM text summarization and investigate two topologies: centralized and decentralized.
- We find that multi-LLM text summarization often performs better than using a single LLM for summarization, and we show that the best performing method in the framework aligns with human judgments.
- We conduct experiments on how prompting, number of LLMs, and various combinations of generating and evaluating LLMs can affect quality of summaries in the multi-LLM setup.

2 Related Work

2.1 Summarization

Recent advancements in summarization have increasingly leveraged large language models (LLMs), moving beyond fine-tuned transformer models like Pegasus, BART, and T5. Studies consistently show that LLMs can generate summaries with higher coherence, relevance, and factual accuracy, often rivaling or surpassing human-written summaries (Goyal et al., 2023; Zhang et al., 2023; Pu et al., 2023b).

For example, Goyal et al. (2023) demonstrated that GPT-3 (text-davinci-002) produced summaries preferred by human evaluators over fine-tuned models like Pegasus and BRIO on structured datasets such as CNN/DM (Nallapati et al., 2016) and

XSUM (Narayan et al., 2018). Similarly, Zhang et al. (2023) emphasized the importance of instruction tuning in achieving superior zero-shot performance for summarization tasks. Pu et al. (2023b) further highlighted improved factual consistency and reduced hallucinations when using LLMs.

While these studies validate the potential of LLMs in summarizing well-structured texts, they may falter for inputs lacking clear structural cues and exhibiting greater complexity. Research focusing on longer text summarization, such as Keswani et al. (2024), employed semantic clustering and multi-stage summarization with LLaMA2 to manage lengthy inputs. However, such approaches often rely on predefined hierarchical processing strategies that may oversimplify the nuanced relationships within the text. Moreover, as Liu et al. (2023) noted, LLMs tend to neglect content from the middle sections of longer documents, resulting in incomplete or unbalanced summaries.

Our work aims to improve performance for both long and short text summarization, and it builds upon aforementioned foundations by proposing a multi-LLM framework designed to overcome these shortcomings through information exchange and collaborative synthesis.

2.2 Multi-LLM

The concept of leveraging multiple LLMs collaboratively has gained traction in recent research, particularly for tasks requiring complex reasoning and factual accuracy. For instance, Liang et al. (2024) introduced the Multi-Agent-Debate (MAD) framework, where LLMs engage in iterative debates to refine their reasoning. This framework demonstrated that a multi-agent GPT-3.5-Turbo setup outperformed GPT-4 on reasoning datasets. Similarly, Chen et al. (2024) proposed RECONCILE, a framework where LLMs collaboratively refine answers and explanations, achieving significant improvements over single-agent systems. Li et al. (2024) extended this line of research by optimizing agent connections, showing that sparse networks can maintain performance while reducing computational overhead.

Although these studies reveal the potential of multi-LLM approaches, their focus remains on structured reasoning tasks, such as question answering and fact-checking. They have not been adequately explored in the context of synthesizing distributed information, addressing content imbalances, and preserving the coherence of summaries across extended texts.

We hope to bridge this gap by adapting multi-LLM frameworks to the domain of document summarization, addressing limitations of both single LLM and traditional hierarchical techniques, and positioning multi-LLM summarization as a promising solution.

3 Multi-LLM Summarization Framework

In this work, we propose a novel multi-LLM summarization framework that leverages multiple large language models to enhance summarization quality of long document input. Through the distribution of generation and evaluation of candidate summaries across multiple models, our framework aims to provide better summaries than single LLM methods, leveraging expertise from different models. We present two interaction topologies, **centralized** and **decentralized**, to guide the collaboration, evaluation, and refinement of summaries between LLMs. Visually these two methods can be represented at a high level in Figure 1. In the datasets we test, articles are typically tens of thousands of words long and exceed the context window of most standard LLMs. To handle this, we establish a two stage process that involves chunking the source document, independently summarizing each chunk of the source document, and then applying a second round of chunking and summarization on the concatenated intermediate results. Throughout both these stages, both frameworks allow multiple LLMs to collaborate and converge on a single final high quality summary of the entire original reference document. Table 1 provides an overview of our framework's four main variations.

4 Centralized Multi-LLM Summarization

The steps for centralized summarization can be found in Algorithm 1. This method leverages multiple LLMs to generate candidate summaries and uses a central LLM to evaluate their quality and guide iterative refinements.

4.1 Single Round

In the simplest case, we prompt each LLM once, gather their summaries, and then perform a single evaluation step to select the best final summary. This is the initial process before we extend it to multiple rounds.

Multi-LLM Summarization Framework	General Mechanism	Stage
CENTRALIZED (Sec. 4)	Single-Round (Sec. 4.1)	Generation (§ 4.1.1) Evaluation (§ 4.1.2)
	Conversational (Sec. 4.2)	Generation (§ 4.2.1) Evaluation (§ 4.2.2)
DECENTRALIZED (Sec. 5)	Single-Round (Sec. 5.1)	Generation (§ 5.1.1) Evaluation (§ 5.1.2)
	Conversational (Sec. 5.2)	Generation (§ 5.2.1) Evaluation (§ 5.2.2)

Table 1: Overview of Multi-LLM Summarization Framework (Sections 4-5).
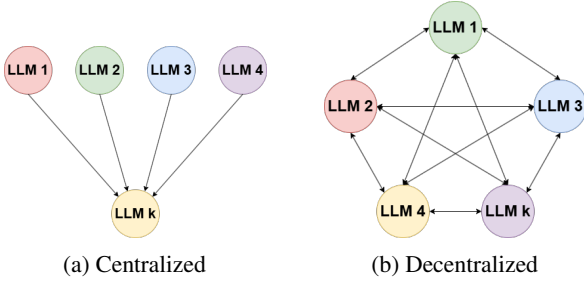

(a) Centralized   (b) Decentralized

Figure 1: Centralized and Decentralized approaches using a 5-LLM example. Similar topologies can be applied to any ("k") number of LLMs. In centralized interactions, all models communicate with a central model; in decentralized interactions, each model communicate with every other model and also itself.

4.1.1 Generation Phase

In the single-round setting, each LLM from the list of participating models $\mathcal{M} = \{M_1, \ldots, M_k\}$ independently generates a summary of the same input text using a common prompt P. The prompt P is illustrated in Figure 2. Formally, for each LLM $M_j \in \mathcal{M}$, the output is

$$S_j = M_j(P, S)$$

where S represents the input text. Running this step for all M_j yields a set of summaries $\mathcal{S} = \{S_1, \ldots, S_k\}$.

This initial generation stage corresponds to line 4 of Algorithm 1. Conceptually, each model contributes its unique perspective, leading to a diverse pool of candidate summaries, which is important for robust summary selection in the following evaluation phase.

4.1.2 Evaluation Phase

After collecting the set of candidate summaries $\mathcal{S}$, we select a central agent $C \in \mathcal{M}$ to evaluate these summaries. The central LLM C uses an evaluation prompt P_{ec}, as shown in Figure 5, to

assess the quality of each summary. To reduce potential bias arising from authorship attribution, we use anonymized identifiers for summaries like agent_1, agent_2, etc. during evaluation.

Formally, we obtain $E = C(P_{ec}, \mathcal{S})$, where E is the central LLM's evaluation of all candidate summaries. This includes the choice for the best summary (expressed as its anonymized identifier) and a confidence score for that evaluation (expressed as an integer from 0 to 10), denoted together as $\mathbf{r} = \text{AGGRRESULTS}(E)$ in Algorithm 1. We de-anonymize the identifier to recover the text of the selected summary S_j and set this as our final output S^*. In the single-round regime, this terminates the process as no further iterations are performed.

In the evaluation prompt, we include the prompt to output a confidence score so there is a variable on which to impose a stopping condition. This allows us to extend the centralized process to multiple rounds of generation and evaluation using that condition. This process is explained in subsequent sections.

4.2 Conversational

In the conversational approach, we repeat the generation and evaluation phases multiple times. We define each generation-evaluation process as one round and define conditions under which the process ends or a new round should begin, up to a maximum number of rounds.

4.2.1 Generation Phase

The first round of the conversational approach mirrors the single-round procedure (Section 4.1.1). Each LLM M_j generates an initial summary $S_j^{(1)}$ from the original input text S using the prompt P:

$$S_j^{(1)} = M_j(P, S).$$

If the evaluation result from the previous round

has a confidence score less than the threshold or, if the LLM fails to output a readable confidence score, the pipeline proceeds to the next round. For the second and subsequent rounds, we use the prompt $P^{(i)}$, shown in Figure 3. LLMs in the second and subsequent rounds have access to both the text to be summarized and summaries from the previous round. Concretely, in round $i > 1$:

$$S_j^{(i)} = M_j(P^{(i)}, S).$$

The hope is that LLM is able to iteratively improve summarization based upon previous outputs from itself and other models.

4.2.2 Evaluation Phase

The evaluation phase in round $i > 1$ is conceptually similar to the single-round setting (Section 4.1.2), but now operates on candidate summaries generated immediately before in the generation phase $\mathcal{S}_i = \{S_1^{(i)}, \ldots, S_k^{(i)}\}$. The central LLM C evaluates these candidates using P_{ec}:

$$E^{(i)} = C(P_{ec}, \mathcal{S}_i),$$

If the confidence level meets the threshold, the process terminates, and the summary chosen by the central LLM is accepted as S^*. Otherwise, we proceed to the next round of summary generation and evaluation. For the confidence scores we have chosen the range 0-10 as it is fine-grained but also is one of the most common rating scales.

4.3 Analysis of Complexity

The centralized approach uses k models for generation and 1 central model for evaluation; other than text length, the number of input tokens scale linearly with the number of models and with the number of rounds. Output tokens also scale linearly with number of models and number of rounds, but since we instruct the model to output a fixed number of words for summary (and in our experiments the models are largely compliant), and output only the anonymous identifier for a chosen summary, we ensure bounded runtime and cost. Further analysis can be found at Appendix B.1.

5 Decentralized Multi-LLM Summarization

Previously we introduced the summarization procedure for centralized approach (Section 4), which diversifies the knowledge base for summarization.

Algorithm 1 Centralized Multi-LLM Summary

Require: ordered set $\mathcal{S} = \{S_1, \ldots, S_m\}$ of summaries, set $\mathcal{M} = \{M_1, \ldots, M_k\}$ of k LLMs, a central agent $C \in \mathcal{M}$, max number of conversational rounds $t_{\max}$, initial summarization prompt P (*e.g.*, Figure 2), evaluation prompt $P_e c$ (*e.g.*, Figure 5) for centralized version
Ensure: summary S^* of the text
1: $S = \text{CREATESUMMARY}(\mathcal{S})$
2: **for** $i = 1$ to $t_{\max}$ **do** ▷ conversation rounds
3:     **for each** model $M_j \in \mathcal{M}$ **do**
4:         $S_j^{(i)} = M_j(P, S)$
5:     Let $\mathcal{S}_i = \{S_1^{(i)}, S_2^{(i)}, \ldots, S_k^{(i)}\}$
6:     $E^{(i)} = C(P_{ec}, \mathcal{S}_i)$
7:     $\mathbf{r} = \text{AGGRRESULTS}(E^{(i)})$
8:     $j \leftarrow \text{argmax}_{M_j \in \mathcal{M}} r_j$
9:     Set $S^* \leftarrow S_j^{(i)}$
10:     **if** $\text{CONVERGED}(\mathbf{r})$ **then return** S^*
11:     Set P to prompt in Figure 3.

```
Provide a concise summary of the text in
around 160 words. Output the summary text
only and nothing else.
                [text]
```

Figure 2: Prompt for generating the initial summary in the first round.

```
Given the original text below, along with
the summaries of that text by [k] LLMs,
please generate a better summary of the
original text in about 160 words.

ORIGINAL:
                [text]
Summary by M_1:
        [LLM 1's summary]
                  ⋮
Summary by M_k:
        [LLM k's summary]
```

Figure 3: Generation prompt that is used after the initial round of conversation among the multiple LLMs. Note that the above prompt is for generating the final summary, however, for the chunk-level generation, it would just be the actual chunk.

We extend the paradigm for the evaluator as well. In the decentralized approach, multiple LLMs also participate in the evaluation process with the hope that a best summary decided on consensus is more robust compared to a single model's decision.

5.1 Single Round

5.1.1 Generation Phase

Generation procedure is the same as that in the centralized approach described in Section 4.1.1. As before, multiple LLMs independently generate

```
Given the original text below, along with
the summaries of that text by [k] agents,
please evaluate the summaries and output
the name of the agent that has the best
summary.  Output the exact name only and
nothing else.

ORIGINAL:
[chunk or concatenated chunk summaries S]
Summary by agent_1:
            [LLM 1's summary]
                    ⋮
Summary by agent_k:
            [LLM k's summary]
```

Figure 4: Evaluation prompt for evaluating the summaries generated by different LLMs using our conversational (decentralized) multi-LLM framework. "k" is a parameter reflecting the number of LLMs that generate summaries.

```
Given the initial text below, along with
the summaries of that text by [k] LLMs,
please evaluate the generated summaries
and output the name of the LLM has the
best summary. On a separate line indicate
a confidence level between 0 and 10.
ORIGINAL:
                [text]
Summary by M_1:
            [LLM 1's summary]
                    ⋮
Summary by M_k:
            [LLM k's summary]
Remember, on a separate line indicate a
confidence level between 0 and 10
```

Figure 5: Evaluation prompt for evaluating the summaries generated using our conversational (centralized) multi-LLM framework. More specifically, we have added an instruction for centralized multi-LLM summarization approach that in addition to providing the best summary, it also outputs the confidence level between 0 and 10. "k" is a parameter reflecting the number of summary-generating LLMs.

summaries for the input text, obtaining the list of summaries $\mathcal{S} = \{S_1, \ldots, S_k\}$.

5.1.2 Evaluation Phase

For evaluation, each model that authored a summary is prompted with a new evaluation prompt (Figure 4) which does not include a confidence level and receives the text to be summarized along with summaries authored by all agents including itself. More formally, model preferences $E_1^{(i)}, \ldots, E_k^{(i)}$ are collected, where each $E_j^{(i)}$ represents model M_j's choice of the best summary

```
Provide a concise summary of the text in
around 160 words. Output the summary text
only and nothing else.
    [concatenated chunk summaries S]
```

Figure 6: Generation prompt for generating the final summary from the summarized chunks using our conversational (decentralized) multi-LLM framework. This prompt is the same as the one for the initial summary.

among $S_1^{(i)}, \ldots, S_k^{(i)}$. These preferences are aggregated into a result vector $\mathbf{r} \in 1, \ldots, k^k$, where each element r_j indicates which model's summary was chosen by model M_j. Convergence is achieved when a majority of models select the same summary, formally expressed as $\exists m \in 1, \ldots, k : |j : r_j = m| > \frac{k}{2}$. [1] When no majority choice emerges, the single-round approach ($t_{\max} = 1$) the algorithm selects the summary from a designated tie-breaker model M_t, where $t \in 1, \ldots, k$. Since the tie-breaker model can be any model in the multi-LLM setup, we run experiments with different choices of evaluator and tie-breaking models. Formally, the final summary S^* is determined as:

$$S^* = \begin{cases} S_m^{(1)} & \text{if } \exists m : |\{j : E_j^{(1)} = m\}| > \frac{k}{2} \\ S_t^{(1)} & \text{if } \max_l |\{j : E_j^{(1)} = l\}| \leq \frac{k}{2} \end{cases}$$

where $m \in 1, \ldots, k : |j : r_j = m| > \frac{k}{2}$. We test the different choices of tie-breaker model in the experiment Appendix C.1.

5.2 Conversational

The conversational approach extends the decentralized framework by introducing multiple rounds of generation and evaluation phases. Each generation-evaluation cycle constitutes a round, with iterations continuing until either consensus is achieved or a maximum number of rounds ($t_{\max}$) is reached.

5.2.1 Generation Phase

Generation follows the methodology in Section 4.1.1, producing the set of summaries $\mathcal{S} = S_1, \ldots, S_k$. A key distinction from the single-round approach lies in the conditional regeneration mechanism: when consensus fails in the first round, subsequent rounds use a new prompt (Figure 3) which includes generated summaries from previous evaluations.

---

[1]Here our implementation requires votes exceeding absolute majority for a summary to be immediately selected. In the case of 2 LLMs, this is equivalent to a unanimous decision because one vote does not satisfy absolute majority.

Algorithm 2 Decentralized Multi-LLM Summary

Require: ordered set $\mathcal{S} = \{S_1, \ldots, S_m\}$ of summaries, set $\mathcal{M} = \{M_1, \ldots, M_k\}$ of k LLMs, max number of conversational rounds $t_{\max}$, initial summarization prompt P (*e.g.*, Figure 2), evaluation prompt P_e (*e.g.*, Figure 4)
Ensure: summary S^* of the text
1: $S = \textsc{CreateSummary}(\mathcal{S})$
2: **for** $i = 1$ to $t_{\max}$ **do**   ▷ conversation rounds
3:     **for each** model $M_j \in \mathcal{M}$ **do**
4:         $S_j^{(i)} = M_j(P, S)$
5:     Let $\mathcal{S}_i = \{S_1^{(i)}, S_2^{(i)}, \ldots, S_k^{(i)}\}$
6:     **for each** model $M_j \in \mathcal{M}$ **do**
7:         $E_j^{(i)} = M_j(P_e, S_1^{(i)}, \ldots, S_k^{(i)})$
8:     Set $\mathcal{E}_i = \{E_1^{(i)}, E_2^{(i)}, \ldots, E_k^{(i)}\}$
9:     $\mathbf{r} = \textsc{AggrResults}(E_1^{(i)}, \ldots, E_k^{(i)})$
10:     $j \leftarrow \mathrm{argmax}_{M_j \in \mathcal{M}} \, r_j$
11:     Set $S^* \leftarrow S_j^{(i)}$
12:     **if** $\textsc{Converged}(\mathbf{r})$ **then return** S^*
13:     Set P to prompt in Figure 3.

5.2.2 Evaluation Phase

The first round of evaluation is identical to that in the single-round approach, but enters additional rounds with new generation prompts. Formally, let $E_j^{(i)}$ represent model M_j's choice in round i. In the single-round case, non-consensus (when $\max_m |\{j : E_j^{(i)} = m\}| \leq \frac{k}{2}$) triggers an immediate fallback to a tie-breaker model. In contrast, the conversational approach initiates a new generation-evaluation round with an updated prompt (Figure 3). This process continues until either a majority consensus emerges or $t_{\max}$ rounds are exhausted. After $t_{\max}$ rounds without a consensus, the algorithm defaults to the tie-breaker mechanism described in Section 5.1.2.

5.3 Analysis of Complexity

The decentralized approach uses k models for both generation and evaluation. For this reason the input and output tokens scale quadratically with number of models. As before, we instruct the model to output a fixed number of words for summary and an identifier only for evaluation and so ensure bounded runtime and cost. Further analysis can be found at Appendix B.2.

6 Experiments

To investigate the proposed multi-LLM summarization framework, we conduct extensive experiments to evaluate its effectiveness.

6.1 Experimental Setup

We use ArXiv (Cohan et al., 2018) and GovReport (Huang et al., 2021) to evaluate our summarization methods. We assess the quality of LLM-generated summaries using ROUGE-1, ROUGE-L, BLEU-1, and BLEU-4 metrics. For comparison with our multi-LLM approach, unless otherwise mentioned, we leverage GPT-3.5, GPT-4o, GPT-4o mini, and LLaMA3-8B as baselines. For these models, we perform the same chunking across all models, and the summarization prompt is identical to that in the first round of the multi-LLM process (Figure 6). Unless otherwise mentioned, all models use 4K-char chunk-size, and the final summary represents a concatenation of the generated summaries. Finally, unless otherwise mentioned, we set $W = 160$ for all the models.

6.2 Main Results

Our multi-LLM framework outperforms single-LLM baselines by up to $3\times$, as seen in Table 2. The fact that both precision- and recall-focused metrics improved means the multi-LLM approach is robust. On average the centralized method improves the scores by 73%, and the decentralized method outperforms baselines by 70%. In our theoretical cost analysis (Section B.1 and B.2) we show that the input cost (in number of tokens) for the decentralized multiplies by the the number of agents participating in the evaluation, and with the more cost-effective centralized method our system is able to perform better than the single-LLM setup. This demonstrates the effectiveness of our proposed method under decentralized and decentralized frameworks.

We see that additional LLMs do not improve upon the 2-LLM setup (see Appendix C.3), and additional rounds of generation and evaluation do not further improve scores. This shows that even with just 2 LLMs and a single round of generation and evaluation we observe performance gains, meaning that the least costly version of the multi-LLM system is still able to deliver better summaries compared to single-LLM approaches.

In Table 2 we use GPT-3.5 as the evaluator and tie-breaking choice in our multi-LLM. We also run the multi-LLM system with GPT-4o mini as the evaluator and the tie-breaker, and the results are shown in Table 5. Again, the multi-LLM framework outperformed single-LLM baselines, averaging 64% improvement for the decentralized variant and 63% for the centralized variant. In some

		ArXiv				GovReport			
		ROUGE-1 ↑	ROUGE-L ↑	BLEU-1 ↑	BLEU-4 ↑	ROUGE-1 ↑	ROUGE-L ↑	BLEU-1 ↑	BLEU-4 ↑
	LLaMA3-8B	0.180	0.106	0.084	0.021	0.403	0.177	0.242	0.079
	GPT-3.5	0.193	0.114	0.093	0.026	0.390	0.178	0.226	0.084
	GPT-4o mini	0.217	0.118	0.108	0.020	0.384	0.156	0.224	0.058
	GPT-4o	0.165	0.095	0.073	0.015	0.372	0.155	0.211	0.059
Decentralized	Multi-LLM 3 round max	0.313	0.163	0.200	0.029	0.447	0.180	0.458	0.098
	Multi-LLM 1 round max	**0.339**	**0.180**	**0.224**	**0.043**	0.468	0.190	0.477	0.112
Centralized	Multi-LLM 3 round max	0.329	0.168	0.217	0.031	0.468	0.189	0.470	0.109
	Multi-LLM 1 round max	0.333	0.173	0.219	0.036	**0.479**	**0.197**	**0.485**	**0.121**

Table 2: Results for the **decentralized** and **centralized** Multi-LLM approaches. For the multi-LLM pipelines participating models are GPT-3.5 and GPT-4o mini. The results use GPT-3.5 for the evaluator in the centralized approach, and summaries from GPT-3.5 are chosen in tie-breaking for both centralized and de-centralized approaches.

individual scores, our framework improves upon single-LLM setups by up to 3×. These improvements are competitive to those we obtain from the multi-LLM setup in Table 2, which means our proposed framework works well for different central models and different tie-breaking models.

We also perform additional experiments with other variables. More specifically, we assess the performance of the multi-LLM framework with alternative combinations of models, with three models contributing to the summarization and evaluation, and with models receiving fine-grained prompts instead of the same prompt. In all of these experiments, we obtain competitive results compared to the first decentralized and centralized setup, and the scores are higher than single-LLM baselines, showing that our proposed framework performs consistently under different setups.

6.3 Ablation Studies

Varying Model Combinations In Table 2 we use GPT-3.5 and GPT-4o mini as the participating models in the multi-LLM framework. We further experiment with alternative combinations of models in the framework. As shown in Table 3 we again observe improvements across the board compared to the single-LLM baselines in Table 2, regardless of default model and number of rounds and type of interaction (decentralized vs. centralized). The improvements are competitive with those seen in the GPT-3.5 and GPT-4o mini combination. Further results are provided in Appendix C.2.

Varying the Number of LLMs In this experiment we use 3 LLMs in the setup instead of 2. We observe a 54% improvement for the decentralized method and 59% for the centralized method on average over single-LLM summaries, and for individual scores we see improvements of up to 2.9×. More detailed results are presented in Table 6 and

in Appendix C.3.

Specialized Prompting In all previous experiments we have kept the generation prompt identical for all LLMs. With multi-LLM approaches, this need not be the case. In this experiment we choose different prompts for different models when generating summaries, aiming to have unique knowledge bases of different models complement each other. As seen in Table 7, the centralized method results in a 66% performance increase over single-LLM baselines in Table 2, and the decentralized method has a 58% increase over the single-LLM baselines. For experimental details and further analysis see Section C.4 in the Appendix

Short vs. Long-text Multi-LLM Summarization In this experiment, we use only the introduction section as the basis for summarization in the ArXiv dataset. Since the introduction typically shorter than the context window of LLMs, we refer to these as "short-text" summarization, in contrast to the "long-text" summarization we explore previously. The results in Table 8 shows that the centralized approach provides the most performance gains over single-LLM baselines – up to 2.4× on average, and the decentralized method sees a 2.3× increase. Further details can be found in Appendix C.5.

6.4 Cost Analysis

Table 4 presents the cost analysis for both decentralized and centralized methods based on the results in Table 2, highlighting key trends in input and output tokens across various stages of the summarization process. We observe that for evaluation stages the input and output token counts for the decentralized method are twice those for the centralized method, which reflect the number of LLMs in the setup.

	Max Rounds	Multi-LLM Model Combination	ROUGE-1 ↑	ROUGE-L ↑	BLEU-1 ↑	BLEU-4 ↑
Decentralized	**3 Rounds**	GPT-3.5 & GPT-4o mini	**0.313**	**0.163**	**0.200**	**0.029**
		GPT-4o & GPT-3.5	**0.313**	0.159	0.197	0.025
		GPT-4o & GPT-4o mini	0.302	0.152	0.185	0.022
	1 Rounds	GPT-3.5 & GPT-4o mini	**0.339**	**0.180**	**0.224**	**0.043**
		GPT-4o & GPT-3.5	0.328	0.170	0.212	0.033
		GPT-4o & GPT-4o mini	0.305	0.153	0.189	0.023
Centralized	**3 Rounds**	GPT-3.5 & GPT-4o mini	**0.329**	**0.168**	**0.217**	**0.031**
		GPT-4o & GPT-3.5	0.325	0.166	0.214	0.029
		GPT-4o & GPT-4o mini	0.304	0.153	0.188	0.022
	1 Rounds	GPT-3.5 & GPT-4o mini	0.333	**0.173**	0.219	**0.036**
		GPT-4o & GPT-3.5	**0.339**	0.177	**0.228**	0.039
		GPT-4o & GPT-4o mini	0.306	0.155	0.190	0.022

Table 3: Varying the combination of models in our Multi-LLM approaches. Note rounds is the max number of rounds allowed and all results are for ArXiv. Bolded numbers are best scores for each round-model combination. Underlined numbers are overall best scores for each metric in this table. Furthermore, the central LLM is highlighted in blue and for the decentralized multi-LLM approaches, we highlight the LLM used for tie-breaking in green.

		Input Tokens	Output Tokens	Average Tokens	Total Tokens
Decentralized	Multi-LLM 3 round max	383.73M	25.63M	14.62M	409.37M
	Multi-LLM 1 round max	129.36M	11.89M	11.77M	141.25M
Centralized	Multi-LLM 3 round max	216.65M	19.55M	14.76M	236.2M
	Multi-LLM 1 round max	77.69M	6.77M	10.56M	84.46M

Table 4: Cost Analysis of our Multi-LLM Decentralized and Centralized Summarization Methods. Note $M=$ millions of tokens.

6.5 Human evaluation

In addition to the ablation studies, we perform human evaluation of summaries similar to the last step of the multi-LLM framework, i.e. the evaluation phase (Section 4.2.2). The human raters are prompted with two sets of summaries from the generation phase (Section 4.1.1), and are instructed to evaluate these two sets of summaries for Coherence, Conciseness, and Fluency on 5-point Likert scales (Conroy and Dang, 2008) with rating guidelines for each possible score (Figure 10). The summaries are randomized and anonymized to reduce bias attributable to knowledge of authorship. We drop the Relevance criteria since no original text is provided to the human raters due to length.

We obtain 420 ratings from 7 raters, and find that humans generally prefer summaries produced by GPT-4o mini, which aligns with preferences by our multi-LLM framework. Human preferences also align with machine preferences for all three evaluation criteria to some degree. Conciseness has the highest agreement with multi-LLM evaluations ($\kappa = 0.6$). More details for human evaluations can be found in Appendix D.

7 Conclusion

This paper presented a multi-LLM framework for text summarization, and proposed two strategies, decentralized and centralized multi-LLM summarization. We demonstrated that the proposed multi-LLM summarization techniques lead to better generated summaries. Our results indicate that multi-LLM approaches are useful for improving text summarization. Future work should continue to investigate multi-LLM approaches for summarization.

8 Limitations

This work demonstrated the effectiveness of both our centralized and decentralized multi-LLM summarization approaches. Future work should further investigate various aspects, including more diverse LLMs, and explore other topologies beyond the two extremes we proposed. Furthermore, while we investigated a variety of datasets, future work can explore other domains. We believe there are many approaches that lie between the two extreme multi-LLM strategies we investigated empirically in this work. Finally, we did not optimize the prompts, as such we believe there is huge opportunity to achieve significantly better results by engineering better prompts to consider other important aspects

of summarization. We leave these and other important directions for future work.

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.

Joshua Ainslie, Santiago Ontanon, Chris Alberti, Vaclav Cvicek, Zachary Fisher, Philip Pham, Anirudh Ravula, Sumit Sanghai, Qifan Wang, and Li Yang. 2020. Etc: Encoding long and structured inputs in transformers.

Lochan Basyal and Mihir Sanghvi. 2023. Text summarization using large language models: A comparative study of mpt-7b-instruct, falcon-7b-instruct, and openai chat-gpt models.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer.

Yapei Chang, Kyle Lo, Tanya Goyal, and Mohit Iyyer. 2024. Booookscore: A systematic exploration of book-length summarization in the era of llms.

Justin Chih-Yao Chen, Swarnadeep Saha, and Mohit Bansal. 2024. Reconcile: Round-table conference improves reasoning via consensus among diverse llms.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2022. Palm: Scaling language modeling with pathways.

Janara Christensen, Stephen Soderland, Gagan Bansal, and Mausam. 2014. Hierarchical summarization: Scaling up multi-document summarization. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 902–912. 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.

John M. Conroy and Hoa Trang Dang. 2008. Mind the gap: Dangers of divorcing evaluations of summary content from linguistic quality. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 145–152, Manchester, UK. Coling 2008 Organizing Committee.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc Le, and Ruslan Salakhutdinov. 2019. Transformer-xl: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2978–2988, Florence, Italy. Association for 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.

Alexios Gidiotis and Grigorios Tsoumakas. 2020. A divide-and-conquer approach to the summarization of long documents. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:3029–3040.

Yihong Gong and Xin Liu. 2001. Generic text summarization using relevance measure and latent semantic analysis. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 19–25. ACM.

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

Mandy Guo, Joshua Ainslie, David Uthus, Santiago Ontanon, Jianmo Ni, Yun-Hsuan Sung, and Yinfei Yang. 2021. Longt5: Efficient text-to-text transformer for long sequences.

Karl Moritz Hermann, Tomáš Kočiský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend.

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.

Emma Järvinen. 2024. Long-input summarization using large language models.

Gunjan Keswani, Wani Bisen, Hirkani Padwad, Yash Wankhedkar, Sudhanshu Pandey, and Ayushi Soni. 2024. Abstractive long text summarization using

large language models. *International Journal of Intelligent Systems and Applications in Engineering*, 12(12s):160–168.

Anastassia Kornilova and Vladimir Eidelman. 2019. BillSum: A corpus for automatic summarization of US legislation. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pages 48–56, Hong Kong, China. Association for Computational Linguistics.

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.

Irene Li, Aosong Feng, Dragomir Radev, and Rex Ying. 2023. Hipool: Modeling long documents using graph neural networks. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 161–171, Toronto, Canada. Association for Computational Linguistics.

Yunxuan Li, Yibing Du, Jiageng Zhang, Le Hou, Peter Grabowski, Yeqing Li, and Eugene Ie. 2024. Improving multi-agent debate with sparse communication topology.

Tian Liang, Zhiwei He, Wenxiang Jiao, Xing Wang, Rui Wang, Yujiu Yang, Zhaopeng Tu, and Shuming Shi. 2024. Encouraging divergent thinking in large language models through multi-agent debate.

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

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

Yixin Liu, Pengfei Liu, Dragomir Radev, and Graham Neubig. 2022. Brio: Bringing order to abstractive summarization.

S. Mallick, A. Ghosh, et al. 2019. A survey on extractive text summarization. *Journal of Artificial Intelligence Research*, 65:123–143.

Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into texts. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 404–411.

Ramesh Nallapati, Bowen Zhou, Cicero Nogueira dos santos, Caglar Gulcehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond.

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.

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

Dongqi Pu, Yifan Wang, and Vera Demberg. 2023a. Incorporating distributions of discourse structure for long document abstractive summarization.

Xiao Pu, Mingqi Gao, and Xiaojun Wan. 2023b. Summarization is (almost) dead.

Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 379–389.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H. Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Fevry, Jason Alan Fries, Ryan Teehan, Tali Bers, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M. Rush. 2022. Multi-task prompted training enables zero-shot task generalization.

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.

Sam Shleifer. 2020. Distilbart-cnn-12-6. https://huggingface.co/sshleifer/distilbart-cnn-12-6. Accessed: 2024-05-29.

Sainbayar Sukhbaatar, Edouard Grave, Piotr Bojanowski, and Armand Joulin. 2019. Adaptive attention span in transformers. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 331–335, Florence, Italy. Association for Computational Linguistics.

Wen Xiao and Giuseppe Carenini. 2019. Extractive summarization of long documents by combining global and local context. In *Proceedings of the*

2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 3011–3021, Hong Kong, China. Association for Computational Linguistics.

Manzil Zaheer, Guru Guruganesh, Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. 2021. Big bird: Transformers for longer sequences.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. 2020. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization.

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

A Detailed Experimental Setup

Datasets: We use the test sets of ArXiv (Cohan et al., 2018) (first 20%, or 1,288 documents) and GovReport (Huang et al., 2021) (all, or 973 documents) as document input for our summarization methods. They cover a range of genres, providing diverse texts for evaluation. In ArXiv, the main article excluding the abstract is the target for summarization, and the abstract is used as the ground truth reference summary; for GovReport, the text is the main report and the ground truth is the human-written summary. ArXiv articles range from 241 to 44,489 space-delimited words long, with an average of 5,950 words; their summaries range from 46 to 290 words, averaging 164 words. GovReport main texts range from 396 to 31,371 words, averaging 7,379 words; their summaries range from 67 to 1,363 words, averaging 571 words.

Evaluation Metrics: We assess the quality of LLM-generated summaries using ROUGE-1, ROUGE-L, BLEU-1, and BLEU-4 metrics. ROUGE scores emphasizes recall while BLEU scores emphasize precision.

Baselines: For comparison with our multi-LLM approach, unless otherwise mentioned, we leverage GPT-3.5, GPT-4o, GPT-4o mini, and LLaMA3-8B as baselines. For these models, we perform the same chunking across all models, and the summarization prompt is identical to that in the first round of the multi-LLM summarization process (Figure 2). Unless otherwise mentioned, all models use 4K-char chunk-size, and the final summary for each document is concatenation of the generated summaries for each chunk in that document.

Finally, unless otherwise mentioned, we set $W = 160$ for all models.

B Theoretical Analysis & Discussion

B.1 Centralized Apporach

Cost and Complexity per Round Let I denote the number of input tokens in the original text and $O_{\max}$ represent an upper bound on the output tokens (i.e., maximum summary length). We consider k distinct LLMs and a maximum of $t_{\max}$ conversational rounds. In each round i, we prompt all k LLMs with approximately $I + \delta_i$ input tokens, where δ_i denotes additional tokens introduced in that round (e.g., references to previously generated summaries). Each LLM then produces up to $O_{\max}$ output tokens. Since input and output tokens often incur different costs, we consider them separately. For the generation phase, the input token cost per round is on the order of $\mathcal{O}(k \cdot (I + \delta_i))$, and the output token cost is on the order of $\mathcal{O}(k \cdot O_{\max})$. For evaluation, the central LLM processes k candidate summaries and I_{ec} instructions, resulting in an input token cost of about $\mathcal{O}(k \cdot O_{\max} + I_{ec})$. By directing the central LLM to output only an anonymous identifier for the chosen summary, we reduce output token length in evaluation, thereby minimizing the chance of hallucination and enabling more straightforward cost accounting.

Multi-Round Overhead Over $t_{\max}$ rounds, the total input token usage for generation is $\mathcal{O}(t_{\max} \cdot k \cdot (I + O_{\max}))$, and the evaluation input token usage is $\mathcal{O}(t_{\max} \cdot (k \cdot O_{\max} + I_{ec}))$. Although this complexity may appear large, $t_{\max}$ is typically small (e.g., 2 or 3), and $O_{\max}$ is usually constrained (e.g., a brief 160-word summary). Moreover, careful prompt engineering can curtail δ_i growth, ensuring that the number of tokens per round remains bounded.

Convergence and Quality Gains The iterative generation-evaluation mechanism aims to converge within a small number of rounds. With each iteration, models refine their outputs guided by previous results, potentially improving summary quality. This iterative refinement, while incurring additional steps, offers a practical trade-off between computation and quality, as the improved summaries can justify the limited number of extra rounds.

B.2 Decentralized Approach

Multi-Round Complexity Let I denote the number of input tokens in the original text and $O_{\max}$

represent an upper bound on the output tokens (i.e., maximum summary length). We consider k distinct LLMs and a maximum of t_{max} conversational rounds.

Over t_{max} rounds, the worst-case token cost from generation is:

$$\mathcal{O}(t_{max} \cdot k \cdot (I + O_{max})).$$

The evaluation cost scales to:

$$\mathcal{O}(t_{max} \cdot (k \cdot I_e + k^2 \cdot O_{max})).$$

Combined, we have a total complexity per round of approximately:

$$\mathcal{O}(k \cdot I + k \cdot O_{max} + k \cdot I_e + k^2 \cdot O_{max}).$$

Thus, for t_{max} rounds, the overall complexity becomes:

$$\mathcal{O}(t_{max} \cdot (k \cdot I + k \cdot I_e + k \cdot O_{max} + k^2 \cdot O_{max})).$$

Since $k^2 \cdot O_{max}$ may dominate for large k, this term can become the bottleneck. However, in practical scenarios, k (the number of LLMs) is often small (e.g., 2–5), making the decentralized evaluation overhead manageable.

Trade-Offs and Practical Considerations The decentralized evaluation approach increases computational overhead compared to the centralized model, as it requires every model to evaluate all candidate summaries. However, this additional cost is justified by the potential gains in robustness and reliability of the final output, but also by the flexibility to rely on multiple, potentially weaker models rather than a single, highly capable central evaluator. By employing a form of consensus voting, the system can arrive at a more stable decision even when no single model is individually strong.

While the added complexity of multi-round conversation can be non-trivial, it may lead to improved summary quality, especially when dealing with contentious or ambiguous source texts. Multiple rounds allow the system to refine the summaries and converge on a stable solution. If consensus emerges quickly, the number of rounds t_{max} can be effectively reduced, thereby decreasing the total computational cost. Conversely, if no consensus is reached, the algorithm ultimately defaults to a tie-break mechanism after t_{max} rounds, ensuring bounded time and cost. As with the centralized approach, prompt engineering and careful parameter selection (e.g., choosing O_{max}, t_{max}, and the number of participating models k) we can mitigate undue complexity.

C Ablation Study

C.1 Varying Evaluation LLM

In this section, we compare the scores of the centralized and decentralized approaches when the evaluator model and the tie-breaker models are GPT-4o or GPT-4o mini (instead of GPT-3.5 as in Table 2). These results are presented in Table 5. The sections where GPT-3.5 is the evaluator are reproduced from Table 2.

In these experiments, the summary-generating models remain the same as those in Table 2. In rows (in Table 5) where GPT-4o is listed as the evaluator, however, the decentralized method would have required GPT-4o to be the default choice for a tie-breaking summary as well when the model has not generated summaries. To remain maximally consistent with previous methodology, we modify the process here so that GPT-4o receives the final-round summaries from the decentralized method where GPT-3.5 is the tie-breaking choice and evaluator and performs a **centralized** evaluation on top of the decentralized results. The reason the GPT-3.5-default results are chosen as the basis instead of GPT-4o mini is because as an evaluator and default choice GPT-3.5 produced better final summaries compared to GPT-4o mini for both centralized and decentralized methods.

The multi-LLM framework outperformed single-LLM baselines, averaging 64% improvement for the decentralized variant and 63% for the centralized variant. In some individual scores, our framework improves upon single-LLM setups by up to 3×. GPT-3.5 emerged as the best-scoring evaluator and the best-scoring tie-breaker choice: for the centralized method, GPT-3.5 as an evaluator and tie-breaking choice outperforms other evaluators and tie-breakers, and for the decentralized method, GPT-3.5 turned out to be the best tie-breaking choice. Furthermore, GPT-3.5 as a centralized evaluator and tie-breaking choice separately outperform **both the decentralized and centralized** methods using other models as the evaluator and tie-breaking choice. As with results in Table 2, additional rounds of evaluation and regeneration do not improve summary scores.

C.2 Varying Model Combinations

In Table 2 we present the results with GPT-3.5 and GPT-4o mini as the models in the combination; we now investigate the performance of our approaches for alternative combinations of LLMs (in Table 3).

We use the following combinations for the 2-LLM framework: GPT-3.5 and GPT-4o mini, with GPT-3.5 as the evaluator and default, GPT-4o and GPT-3.5, again with GPT-3.5 as evaluator and default, and finally GPT-4o and GPT-4o mini, with GPT-4o mini as the evaluator and default.

These alternative combinations all outperform single-LLM baselines. We see a 54% improvement in the decentralized variant and a 59% for the centralized variant. Combinations with GPT-3.5 as a member and the evaluator/default choice offer larger improvements compared to those without GPT-3.5. Since we have used GPT-4o mini as the evaluator and tie-breaker where GPT-3.5 is absent, a possible reason the improvements for these pairings are less than those where GPT-3.5 is present is that GPT-3.5 is a larger model than GPT-4o mini.

C.3 Varying the Number of LLMs

In this experiment, we increase the number of LLMs in our multi-LLM system to ascertain the effects on summary quality, and present the results in Table 6. Here we use GPT-3.5, GPT-4o mini, and GPT-4o in the multi-LLM system. We see that while the 3-LLM system still outperform the single-LLM baseline, increasing the number of LLMs from 2 to 3 does not improve performance upon the 2-LLM system, contrary to the trend observed in the previous sections where 2-LLM system outperform single-LLM baselines.

We offer two possible explanations for this finding. First, adding an additional LLM increases the complexity of the pipeline, which may lead to propagation of noise or redundancy in intermediate summaries. This added complexity could dilute the strengths of individual LLMs and reduce overall coherence and relevance in the final output. Second, the integration of a third LLM introduces a greater risk of inconsistencies in summarization styles, which may negatively affect evaluation metrics like ROUGE that rely on lexical overlap.

C.4 Specialized Prompting

We now investigate using a single LLM to generate multiple different summaries of the text, and then using our framework to obtain the best summary. We explore the efficacy of varying prompt formulations and model parameters in regards to our framework. This experiment is grounded in the intuition that long documents contain very diverse sections within their content which may benefit from different summarization strategies. For example, differ-

```
Generate a summary that enhances coherence
of the text in around 160 words.  Output
the summary text only and nothing else.
                [text]
```

Figure 7: Prompt 1 for generating the initial summary in the first round.

ent chunks of a long document may cover distinct topics, serve various purposes, have diverse writing styles, and/or contain differing density. Given this diversity, a simple uniform summarization prompt is less likely to actually capture the required essential information from each chunk. With this, we propose a form of specialized prompting as a way to leverage the distinctive capabilities and specializations of each model for specific chunks specifically in regards to our framework. We hypothesize that the use of specialized prompting can help further leverage LLM capabilities within our suggested multi-LLM framework to produce higher quality summaries which are more suitable for subsequent evaluation by multiple LLMs.

We begin by generating four initial summaries using two sets of specialized prompts designed for GPT-3.5 and GPT-4o mini, ensuring that each model receives two distinct prompts. One prompt focuses on enhancing the coherence of the resulting summary (see Figure 7), while the second prompt aims to maximize precision in conveying the key facts (see Figure 8). After producing these four baseline summaries, we feed them into our multi-LLM framework, which incorporates two agents — GPT-3.5 and GPT-4o mini—working collaboratively. GPT-3.5 and GPT-4o mini are used for the initial generation of summaries, and GPT-3.5 also serves as the evaluator. The framework and methodology following the generation of the four baseline summaries, as well as their inclusion as input, mirror the procedures used to obtain decentralized and centralized results on ArXiv and GovReport in Table 2, with GPT-3.5 functioning as the evaluator. Results for this experiment are provided in Table 7.

This experiment demonstrates that employing specialized prompting strategies within both decentralized and centralized multi-LLM frameworks significantly enhances the quality of generated summaries. These results show the importance of prompt engineering and strategic framework design in multi-LLM summarization tasks and we leave this for future work.

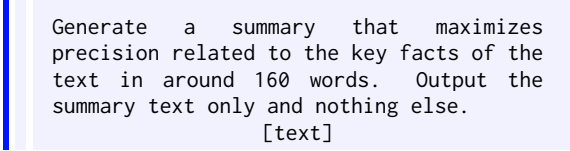|			ArXiv				GovReport			
			ROUGE-1 ↑	ROUGE-L ↑	BLEU-1 ↑	BLEU-4 ↑	ROUGE-1 ↑	ROUGE-L ↑	BLEU-1 ↑	BLEU-4 ↑
GPT-4o mini Evaluator	**Decentralized**	Multi-LLM 3 round max	0.317	0.160	0.206	0.026	**0.445**	**0.178**	**0.452**	**0.094**
		Multi-LLM 1 round max	**0.326**	**0.163**	**0.221**	**0.027**	0.438	0.175	0.446	0.089
	Centralized	Multi-LLM 3 round max	0.315	0.158	0.201	0.027	**0.441**	**0.176**	**0.447**	**0.092**
		Multi-LLM 1 round max	**0.330**	**0.165**	**0.222**	**0.028**	0.439	0.175	0.446	0.090
GPT-3.5 Evaluator	**Decentralized**	Multi-LLM 3 round max	0.313	0.163	0.200	0.029	0.447	0.180	0.458	0.098
		Multi-LLM 1 round max	**0.339**	**0.180**	**0.224**	**0.043**	0.468	0.190	0.477	0.112
	Centralized	Multi-LLM 3 round max	0.329	0.168	0.217	0.031	0.468	0.189	0.470	0.109
		Multi-LLM 1 round max	**0.333**	**0.173**	**0.219**	**0.036**	**0.479**	**0.197**	**0.485**	**0.121**
GPT-4o Evaluator	**Decentralized**	Multi-LLM 3 round max	**0.326**	**0.166**	**0.214**	0.030	0.446	0.179	0.456	0.098
		Multi-LLM 1 round max	0.325	0.165	0.211	0.030	**0.456**	**0.183**	**0.461**	**0.100**
	Centralized	Multi-LLM 3 round max	0.318	0.162	0.206	0.027	0.449	0.181	0.452	0.096
		Multi-LLM 1 round max	**0.327**	**0.167**	**0.215**	**0.031**	**0.461**	**0.186**	**0.467**	**0.105**

Table 5: Results for different evaluating and tie-breaking models for Multi-LLM approaches. The choice of the tie-breaker models is the same as the choice of evaluator model. We bold the best results for each combination of the experimental variables, and we underline the best results overall. For ease of comparison, we reproduce the best-performing 2-LLM results obtained in Table 2

			ArXiv				GovReport			
			ROUGE-1 ↑	ROUGE-L ↑	BLEU-1 ↑	BLEU-4 ↑	ROUGE-1 ↑	ROUGE-L ↑	BLEU-1 ↑	BLEU-4 ↑
2-LLMs GPT-3.5 Evaluator	**Decentralized**	3 rounds	0.313	0.163	0.200	0.029	0.447	0.180	0.458	0.098
		1 rounds	**0.339**	**0.180**	**0.224**	**0.043**	**0.468**	**0.190**	**0.477**	**0.112**
	Centralized	3 rounds	0.329	0.168	0.217	0.031	0.468	0.189	0.470	0.109
		1 rounds	**0.333**	**0.173**	**0.219**	**0.036**	**0.479**	**0.197**	**0.485**	**0.121**
3-LLMs GPT-4o mini Evaluator	**Decentralized**	3 rounds	**0.301**	**0.154**	0.184	**0.024**	**0.445**	0.178	**0.449**	**0.095**
		1 rounds	0.299	0.152	0.184	0.023	0.442	0.178	0.447	0.094
	Centralized	3 rounds	0.300	**0.153**	0.185	0.023	**0.443**	0.178	0.447	**0.094**
		1 rounds	0.300	0.152	**0.186**	0.023	0.442	0.178	**0.449**	0.093
3-LLMs GPT-3.5 Evaluator	**Decentralized**	3 rounds	0.300	0.154	0.184	0.024	0.446	0.179	0.443	0.094
		1 rounds	**0.309**	**0.159**	**0.193**	**0.027**	**0.451**	**0.182**	**0.459**	**0.099**
	Centralized	3 rounds	0.294	0.151	0.177	0.023	0.451	0.181	0.440	0.095
		1 rounds	**0.329**	**0.172**	**0.214**	**0.036**	**0.460**	**0.189**	**0.451**	**0.104**

Table 6: Multi-LLM framework with three models. We bold the best results for each combination of the experimental variables, and we underline the best results overall. For ease of comparison, we reproduce the best-performing 2-LLM results obtained in Table 2

			ArXiv				GovReport			
			ROUGE-1 ↑	ROUGE-L ↑	BLEU-1 ↑	BLEU-4 ↑	ROUGE-1 ↑	ROUGE-L ↑	BLEU-1 ↑	BLEU-4 ↑
Baseline Prompts	**Decentralized**	3 round max	0.313	0.163	0.200	0.029	0.447	0.180	0.458	0.098
		1 round max	**0.339**	**0.180**	**0.224**	**0.043**	**0.468**	**0.190**	**0.477**	**0.112**
	Centralized	3 round max	0.329	0.168	0.217	0.031	0.468	0.189	0.470	0.109
		1 round max	**0.333**	**0.173**	**0.219**	**0.036**	0.479	**0.197**	0.485	**0.121**
Specialized Prompts	**Decentralized**	3 round max	0.300	0.155	0.201	0.025	0.464	0.174	0.441	0.093
		1 round max	**0.338**	**0.175**	**0.236**	**0.040**	**0.469**	**0.181**	**0.486**	**0.104**
	Centralized	3 round max	0.316	0.162	0.215	0.032	0.473	0.177	0.452	0.101
		1 round max	**0.355**	**0.181**	**0.251**	**0.049**	**0.482**	**0.185**	**0.494**	**0.115**

Table 7: Results on the use of 2 specialized prompts on where the only change in the pipeline is that 4 total specialized baseline summaries are fed in initially instead of the 2 simple prompts fed in the methodology used to curate Table 2. Note that these results use GPT-3.5 for the evaluator in the centralized approach, and for breaking ties in the decentralized multi-LLM approaches. This is for a 15 sample size for both datasets. Refer to Figure 7 and Figure 8 for the prompts used for initial generation. We bold the best results for each combination of the experimental variables, and we underline the best results overall.

```
Generate  a  summary  that  maximizes
precision related to the key facts of the
text  in  around  160  words.   Output  the
summary text only and nothing else.
                    [text]
```

Figure 8: Prompt 2 for generating the initial summary in the first round.

C.5   Short-text vs. Long-text Summarization

In this section, we investigate the effectiveness of our approach for shorter text summarization. For this experiment, we leverage the ArXiv dataset and only use the introduction of the paper as input for summarization and evaluate against the same ground-truth. The introduction subsections of papers are typically rich in content yet contain enough brevity to serve as quality standardized reference bases for our goal of long and short text experimentation. With this experiment we present results that showcase the trade offs and performance differences of our methodologies on shorter text summarization compared to that of long document summarization. Generally, ArXiv papers contain detailed markers and section titles to distinguish introduction sections. However, using the Hugging Face dataset of ArXiv papers for our experimentation the format in which the article is represented is a string containing the "body" of the paper which contains little to no explicit markers for section identification. Thus, we present a simple heuristic to distinguish the introduction text from the rest of the article text. We manually went through 5 randomized example articles, with an assumption that the beginning of the article text starts with the introduction section, and found at which inflection point the introduction section concludes. After averaging the word count of the introduction sections and including an extension buffer to capture certain articles which may have slightly longer introduction sections we establish a benchmark for the using the first 1,500 words in ArXiv articles as our reference introduction section. We algorithmically consider a word as a break between the article string wherever there is whitespace. Refer to Figure 9 for more detailed explanation. We ultimately curate 20% of the examples from the test set using this strategy for performance testing on our metrics. Full results are provided in Table 8.

We highlight several key aspects of our multi-LLM summarization methodology using both the centralized and decentralized approaches, showcasing distinct performance across both long and short text summarization tasks. As evident by our results, short articles consistently show better performance compared to long articles, showcasing the inherent complexities and nuances of longer texts that plague LLMs in terms of capturing and summarizing relevant content. The similar performance on metrics like ROUGE-1 and BLEU-4 in our centralized approach across different text lengths might indicate a consistency in how our methodology is able to capture the essential content and has the ability to reproduce the core narrative elements of the text regardless of length. Furthermore, we posit the difference in performance across long and short text for BLEU-1 is based primarily on the metrics itself as it measures the unigram overlap between the generated summary and the reference text. In the case of short texts, the decentralized approach and especially the 3 round performs best as each round and each model provides an opportunity to focus more accurately on and determine crucial unigrams that are significant within the context of a compact introduction section. This iterative refinement likely leads to a higher precision in capturing key terms and phrases, directly contributing to better BLEU-1 scores than in the case of the centralized approach which performs best as the context length is scaled up as shown in the results for long text.

D   Human Evaluation

In the human evaluation, we select the first 10 pairs of summaries generated before the final evaluation step by the decentralized, one-round maximum framework (the best-performing setup for ArXiv; see Table 2), and prompt human raters to rate each summary according to Coherence, Conciseness, and Fluency metrics, each represented by a 5-point Likert scale. The goal is to compare human rater preferences with preferences of the LLM performing the final evaluation and therefore producing the final result of the multi-LLM pipeline. Each summary pair consists of texts generated by GPT-3.5 and GPT-4o mini randomized in order of presentation and anonymized such that the raters do not know which model produced which summaries. We do not prompt raters with the corresponding original text as in the multi-LLM method due to the original text's length and technicality, and therefore we remove the Relevance criterion used in Conroy and Dang (2008). For the remaining criteria we

		ArXiv			
		ROUGE-1 ↑	ROUGE-L ↑	BLEU-1 ↑	BLEU-4 ↑
Long Text	**Decentralized**	Multi-LLM 3 round max	0.329	0.168	0.217	0.031
		Multi-LLM 1 round max	**0.333**	**0.173**	**0.219**	**0.036**
	Centralized	Multi-LLM 3 round max	0.313	0.163	0.200	0.029
		Multi-LLM 1 round max	**0.338**	**0.180**	**0.224**	**0.043**
Short Text	**Decentralized**	Multi-LLM 3 round max	0.360	0.188	**0.328**	0.038
		Multi-LLM 1 round max	**0.369**	**0.198**	0.309	**0.044**
	Centralized	Multi-LLM 3 round max	0.367	0.194	**0.321**	0.041
		Multi-LLM 1 round max	**0.379**	**0.206**	0.305	**0.049**

Table 8: Results on short summarization tasks using the ArXiv dataset for the decentralized and centralized Multi-LLM approaches. Note that these results use GPT-3.5 for the evaluator in the centralized approach, and for breaking ties in the decentralized multi-LLM approaches.

provide guidelines for each possible point value to improve reproducibility. Instructions provided to human raters and rating guidelines can be found in Figure 10.

At the end of human evaluation, we collect 420 ratings from 7 raters (6 men, 1 woman; ages 23-31; 4 East Asian, 2 South Asian, 1 White).[2] To determine preferences for the human raters, we average the rating for each model and each summary in each evaluation criterion (Table 9). Thus, for each summary and for each criterion we obtain two averaged scores, one for GPT-3.5 and GPT-4o mini. We then determine the human preference by choosing the model with the higher score, and if the scores are the same, we fallback on the default choice GPT-3.5, consistent with the fallback default in the evaluation step in our multi-LLM framework (Table 10). We note that in all three criteria our framework show some agreement with the human raters, as measured by Cohen's kappa. For conciseness, we observe an agreement of $\kappa = 0.6$.

---

[2]Raters are authors of this paper and close associates who have consented to submitting evaluations, and no rater has prior knowledge of authors of the set of summaries being evaluated.

Summary Average scores by raters for:	Coherence		Conciseness		Fluency		Averaged	
	GPT-3.5	GPT-4o mini	GPT-3.5	GPT-4o mini	GPT-3.5	GPT-4o mini	GPT-3.5	GPT-4o mini
1	<u>3.57</u>	3.57	<u>**3.85**</u>	3.57	<u>**4.42**</u>	4.28	<u>**3.95**</u>	3.80
2	<u>**4.28**</u>	4.00	<u>**4.42**</u>	3.85	<u>**4.85**</u>	4.85	<u>**4.52**</u>	4.23
3	3.42	<u>**4.57**</u>	3.57	<u>**4.42**</u>	3.85	<u>**4.57**</u>	3.61	<u>**4.52**</u>
4	3.71	<u>**4.57**</u>	3.42	<u>**4.57**</u>	4.28	<u>**4.85**</u>	3.80	<u>**4.66**</u>
5	3.71	<u>**4.14**</u>	3.71	<u>**3.85**</u>	4.00	<u>**4.14**</u>	3.80	<u>**4.04**</u>
6	4.00	<u>**4.71**</u>	3.57	<u>**4.14**</u>	<u>**4.57**</u>	4.42	4.04	<u>**4.42**</u>
7	4.00	<u>**4.71**</u>	<u>**4.00**</u>	4.00	4.28	<u>**4.71**</u>	4.09	<u>**4.47**</u>
8	4.00	<u>**4.57**</u>	4.28	4.28	<u>**4.57**</u>	4.57	4.28	<u>**4.47**</u>
9	4.00	<u>**4.42**</u>	3.85	<u>**4.14**</u>	4.00	<u>**4.42**</u>	3.95	<u>**4.33**</u>
10	<u>**4.14**</u>	3.85	<u>**4.42**</u>	4.00	<u>**4.71**</u>	4.57	<u>**4.42**</u>	4.14

Table 9: Averaged scores (out of 5) given by human raters for each evaluation criterion for the first 10 summaries from the ArXiv dataset. The raters are asked to rate each summary on coherence, conciseness, and fluency on a 5-point Likert scale. We additionally show the score averaged from the scores for the three criteria. We bold the higher average score for each criterion, and underline the choice of the human raters between GPT-3.5 and GPT-4o mini summaries. When two summaries in a particular criterion have the same average score, we fallback on the default choice GPT-3.5, consistent with the evaluation step in our multi-LLM framework.

	Human raters				Multi-LLM (Ours)
Summary	**Coherence**	**Conciseness**	**Fluency**	**Averaged**	
1	GPT-3.5	GPT-3.5	GPT-3.5	GPT-3.5	GPT-3.5
2	GPT-3.5	GPT-3.5	GPT-3.5	GPT-3.5	GPT-4o mini
3	GPT-4o mini	GPT-4o mini	GPT-4o mini	GPT-4o mini	GPT-4o mini
4	GPT-4o mini	GPT-4o mini	GPT-4o mini	GPT-4o mini	GPT-3.5
5	GPT-4o mini	GPT-4o mini	GPT-4o mini	GPT-4o mini	GPT-4o mini
6	GPT-4o mini	GPT-4o mini	GPT-3.5	GPT-4o mini	GPT-4o mini
7	GPT-4o mini	GPT-3.5	GPT-4o mini	GPT-4o mini	GPT-3.5
8	GPT-4o mini	GPT-3.5	GPT-3.5	GPT-4o mini	GPT-3.5
9	GPT-4o mini	GPT-4o mini	GPT-4o mini	GPT-4o mini	GPT-4o mini
10	GPT-3.5	GPT-3.5	GPT-3.5	GPT-3.5	GPT-3.5
Cohen's κ	0.2	0.6	0.1	0.2	−

Table 10: Human choices obtained from choosing the model with the higher average score, done separately for each evaluating criterion. At the right-most column we show the choices made in the final evaluation step by our multi-LLM framework, specifically the decentralized one-round-max setup with GPT-3.5 as the evaluator. At the bottom row we show the inter-rater agreement (as measured by Cohen's kappa) between the human choices and the machine choices for each criterion

heisneberg - like sg magnets is regarded essentially as a chiral - glass transition `` revealed '' via the random magnetic anisotropy . weak but finite random magnetic anisotropy inherent to real magnets `` recouples '' the spin to the chirality , and the chiral - glass transition shows up as an experimentally observable _ spin_-glass transition in real heisneberg - like sg magnets . an interesting outcome of this picture is that the experimental sg transition is dictated by the chiral - glass transition of the fully isotropic system , _ not by the spin - glass transition of the fully isotropic system _ , which has been separated from the chiral one . very recently , the present authors discussed some of the possible consequences of the chirality scenario of ref.@xcite on the finite - field properties of the fully isotropic 3d heisenberg sg@xcite . it was argued there that the chiral - glass transition , essentially of the same character as the zero - field one , occurred also in finite fields . in the weak field regime , the transition line was predicted to behave as @xmath13 where @xmath14 and @xmath10 are constants . generally , the coefficient @xmath14 could be either positive or negative . an interesting observation here is that the chiral - glass transition line ([eqn : phaseline]) aparrently has a form similar to the gt line of the mean - field model . we emphasize , however , that their physical origin is entirely different . the quadratic dependence of the chiral - glass transition line is simply of regular origin , whereas that of the gt - line in the sk model can not be regarded so . in the present paper , we report on our results of large - scale monte carlo simulations on the 3d isotropic heisenberg sg , performed with the aim to reexamine the sg ordering in magnetic fields in light of the chirality scenario . in particular , by means of extensive numerical simulations , we wish to clarify in detail how the spin and the chirality order in applied fields . part of the mc results have been reported in ref.@xcite . the present paper is organized as follows . in [secmodel] , we introduce our model and explain some of the details of our numerical method . various physical quantities calculated in our mc simulations are defined in [secphysq] . the results of mc simulations are presented in [secresult] . the results for the chirality- and spin - related quantities are presented in [subsecchiral] and [subsecspin] , respectively . it is found that the chiral - glass transition , essentially of the same character as the zero - field one , occurs under magnetic fields . the chiral - glass ordered state exhibits a one - step - like peculiar replica - symmetry breaking in the chiral sector , while it does not accompany the spin - glass order perpendicular to the applied field . critical properties of the chiral - glass transition are analyzed in [subseccritical] . the analysis suggests that the universality class of both the zero - field and finite - field chiral - glass transitions might be common , which , however , differs

Figure 9: Here we showcase an example of how we choose at which point an introduction ends. The total word count of this example article was 7,671 and the word count of the reference summary was 172. We highlight the inflection sentence which most serves as the transition from the actual background and theoretical setup of the paper to the actual methodologies which are then detailed in later text. From here we gather the word count of everything before the inflection sentence and classify it as our reference introduction text for experimentation, including the inflection sentence. In this case, the resulting introduction section had a total word count of 1203.

#	Column 1	Summary 1	1: Coherence	1: Conciseness	1: Fluency	Summary 2	2: Coherence	2: Conciseness	2: Fluency
0		The study of short-term periodicities in solar	2	2	4	The periodicities of solar activity have been s	4	4	4
1		The text discusses the identification of perio	2	4	5	The paper discusses the conditions required	4	5	5
2		The text discusses advancements in detectin	5	5	5	This article discusses the potential for pulsar	2	3	2
3		The text presents mathematical equations ar	3	3	5	The text discusses the detection and analysis	5	5	5
4		The list of references covers gravitational wa	1	1	2	The references encompass a range of subjec	2	4	3
5		Researchers have developed a new formula f	4	3	5	Quantum tunneling is crucial in nuclear proce	4	2	2
6		Numerical time integration schemes are esse	2	2	4	Efficient numerical time integration schemes	4	2	4
7		The article explores the incorporation of a for	5	4	5	The article compares the accuracy and comp	1	4	4
8		The paper presents novel methods for derivir	2	2	2	The paper proposes new methods for derivin	2	1	1
9		The method discussed focuses on transform	1	3	3	Researchers have outlined new methods for	1	4	5

Read the two summaries (column C and H) and grade them on the following aspects Give a grade from 1 - 5 for each category, 1 being the lowest score and 5 being the highest score, as described below. For summary 1, fill in columns E-G; for summary 2, fill in columns J-L.

Coherence

Evaluate the logical flow of sentences and organization of information within the summary.

1: Ideas do not logically follow from one to another, even if they are relevant.

2: There are some phrases to connect one idea to another.

3: Main ideas connect logically with each other. A few arguments/points are out of order.

4: Most ideas follow logically from one to the next. There may be minor points or words/phrases that seem out of place.

5: All ideas, even if irrelevant, are well connected and organized in a clear overall structure.

Conciseness

Penalize summaries that are overly verbose.

1: Ideas are repeated multiple times, or are described in excessive detail or are verbose, even if ideas are logically organized or are relevant.

2: Ideas/arguments/phrases are sometimes repeated. Most ideas are described in too much detail, or, the text is generally verbose with jargon.

3: Ideas are generally described in appropriate detail. Parts of texts may be unnecessarily verbose or use unnecessary jargon.

4: Most ideas are described in appropriate detail. There may be occasional verbosity.

5: All ideas are described with appropriately complex or simple sentences.

Fluency

Evaluate the grammatical correctness of the generated summary.

Check for any awkward phrasing or grammatical errors that might hinder comprehension. 1: Text is grammatically incorrect or very difficult to understand. There are incomplete sentences or incorrectly used punctuations.

2: The text is generally difficult to understand, but some sections convey meaningful ideas.

3: The text is generally grammatically correct. Some sentences may have incorrect grammar.

4: The text is grammatically correct and sentences have understandable structure. There are occasional incorrectly phrased sentences.

5: Summary is easy to follow and understand. No grammatical errors.

Figure 10: Screenshot of the interface (with scores already filled in) and instructions given to raters for human evaluation. Instructions include the column indices (not shown in the screenshot) for easier reference. Each summary is rated according to the criteria listed in the instructions (i.e. Coherence, Conciseness, and Fluency). We provide guidelines for each criterion and each possible score for that criterion.