Principled Content Selection to Generate Diverse and Personalized Multi-Document Summaries

Anonymous ACL submission

Abstract

While large language models (LLMs) are increasingly capable of handling longer contexts, recent work has demonstrated that they exhibit the *"lost in the middle"* phenomenon (Liu et al., 2024) of unevenly attending to different parts of the provided context. This hinders their ability to cover diverse source material in multi-document summarization, as noted in the DIVERSESUMM benchmark (Huang et al., 2024). In this work, we contend that principled content selection is a simple way to increase source coverage on this task. As opposed to prompting an LLM to perform the summarization in a single step, we explicitly divide the task into three steps—(1) reducing document collections to atomic key points, (2) using determinantal point processes (DPP) to perform select key points that prioritize diverse content, and (3) rewriting to the final summary. By combining prompting steps, for extraction and rewriting, with principled techniques, for content selection, we consistently improve source coverage on the DIVERSESUMM benchmark across various LLMs. Finally, we also show that by incorporating relevance to a provided user intent into the DPP kernel, we can generate *personalized* summaries that cover *relevant* source information while retaining coverage.

1 Introduction

Recent advances in language modeling have enabled contemporary models to handle very long contexts (Reid et al., 2024; Anthropic, 2024b), spurring new evaluations of their capabilities in these settings (Tay et al., 2021; Pang et al., 2022; Shaham et al., 2022; Kamradt, 2023; Karpinska et al., 2024). As it becomes possible to process these longer inputs, Zheng et al. (2024) observe that a common use case of LLMs involves the summarization of dense information from collections of documents. A key challenge in providing reliable output for the users in this setting is ensuring high coverage of the source material when multiple documents present diverse viewpoints on the same issue—a problem formalized by the DIVERSESUMM benchmark (Huang et al., 2024) as Multi-Document Diversity Summarization (MDDS).

While contemporary models are highly capable, their attention mechanisms tend to prioritize content at the start and end of the context (Liu et al., 2024). This bias is particularly problematic for MDDS, where key details may be spread across multiple documents. As a result, even state-of-the-art LLMs like GPT-4 struggle when prompted to complete the MDDS task (Huang et al., 2024) despite performing well on single-document summarization (Goyal et al., 2022), where clear introductions and conclusions provide natural focal points. Furthermore, deploying LLMs in public-facing interfaces highlights another important facet of the MDDS problem—ensuring reliable coverage of all *relevant* information in a collection of documents to user intents, essentially an instance of *Query-focused* summarization (Daumé III and Marcu, 2006; Vig et al., 2022). There exists an open question to investigate how the attention biases of LLMs interact with information relevance to user intents when generating summaries.

Our research question is: How does content selection impact the source coverage of LLMs in MDDS? (Section 3). We observe that prompting an LLM for the task involves implicitly selecting relevant content and generation into a coherent summary in a single step. Instead, we decouple this single prompting step into principled content selection to prioritize diversity, defending against the aforementioned attention bias, followed by a rewriting step to produce a coherent, high-coverage summary (Figure 1).

In order to select content, we draw inspiration from recent work which shows that LLMs reliably break down individual documents into atomic claims or key points (Kim et al., 2024; Padmaku-

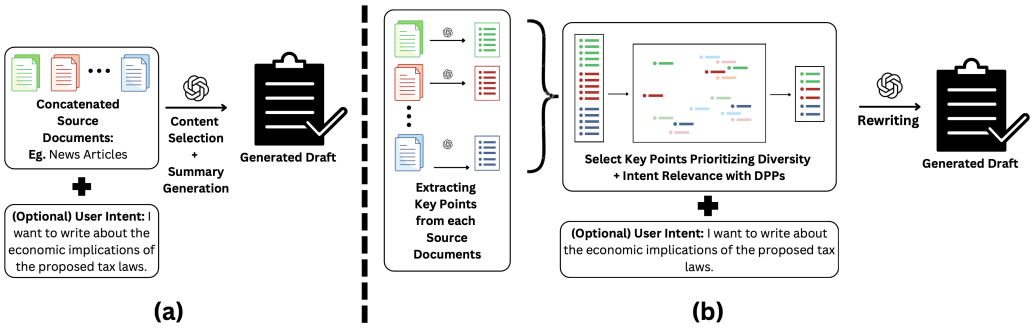


Figure 1: Overview of the MDDS task (Section 2), which aims to generate a summary from a set of source documents with an optional user intent. Compared to (a) prompting an LLM to perform MDDS in a single step (*Naive LLM*) and other baselines, (b) our method (*LLM + DPP*) first extracts atomic key points from each document, then explicitly selects content using DPPs to ensure diversity and relevance before rewriting them into a summary (Section 3). *LLM + DPP* improves source coverage and produces summaries more aligned with user intent (Section 5).

mar and He; Krishna et al., 2023). After extracting key points from each source document, we use *determinantal point processes* (DPPs) (Kulesza et al., 2012) to select the subset of key points used to generate the summary. DPPs are a statistical model that are used to select subsets of items prioritizing diversity.[1] Finally, we rewrite the selected key points into the desired output by prompting an LLM.

We show that using DPPs for diverse content selection consistently improves coverage on the DIVERSESUMM benchmark, compared to both a naive prompting baseline and a multi-step LLM-prompt pipeline, robustly across multiple LLMs—GPT-3.5, GPT-4o, Claude-3-Sonnet, and Llama 3.1 (Section 5.1). Content selection via DPPs can also be tuned to incorporate a relevance matrix generating summaries that are better aligned with user intents (Section 5.2). As LLMs are increasingly deployed in sequential, agentic pipelines for complex tasks, our findings show the value of complementing LLM prompting steps—such as extracting and rewriting key points—with principled techniques like DPPs for content selection, where appropriate, to achieve stronger performance.

2 Problem Formulation

Multi-document diversity summarization (MDDS) The MDDS task, as formulated by Huang et al. (2024), focuses on generating a summary s from a set of articles, $D = \{d_{1...k}\}$, covering the same news story. Each set D is paired with a set of questions $Q = \{q_1, \ldots, q_m\}$, which contain diverse answers drawn from multiple source documents. The objective is to model $p(s|D)$ such that the summary s is faithful to the source content and achieves high coverage, as measured by correctly answering a large number of questions $q_i \in Q$ based on the summary s.

Query-focused Multi-document diversity summarization Building on the MDDS framework, we also explore a variation known as *query-focused* summarization (Daumé III and Marcu, 2006). In this task, the input consists of the set of articles D and a user-specified query q_{user}. The goal is to model $p(s|D, q_{\text{user}})$, where the summary s has high coverage of content *relevant* to the q_{user}. Relevance is determined using a scoring function $f_{\text{rel}}(q_i|q_{\text{user}})$, which identifies the subset of relevant questions $Q_{\text{user}} \subset Q$. We evaluate the summary based on coverage of relevant questions $q_i \in Q_{\text{user}}$.

3 Constructing Documents With Principled Key Point Selection

A typical LLM pipeline for summarizing long contexts involves either concatenating multiple source documents and performing summarization via a single zero-shot prompt (Huang et al., 2024), or hierarchically summarizing the collection, using prompting to process individual documents (Chang et al., 2024). We hypothesize that LLMs might not be well suited to perform the content selection aspect of summarization. To test this, we design a three-step pipeline (Figure 1) that constructs a summary by extracting atomic key points from each document (Section 3.1), selects the key points to be included in the summary in a principled manner, prioritizing diversity of content (Section 3.2.2) as well as relevance to a user intent (Section 3.2.3)

[1]We detail related work that uses DPPs in recommender systems (Section 6.3) and as well as previous approaches to single document summarization (Section 6.2)

and then rewrites the selected key points into the summary (Section 3.3). We then evaluate the coverage of summaries generated with our method (Section 4.2) to various baselines (Section 4.3).

3.1 Key Point Extraction

Given a set of documents $D = \{d_{1...k}\}$, we use an LLM to decompose each document d_i into a set of key points $K_i = \{k_{i,1}, k_{i,2} \dots k_{i,n}\}$ that represent distinct pieces of information within the text. Prior work has demonstrated that LLMs reliably break down individual documents into atomic claims or key points via a zero-shot prompt for various applications (Kim et al., 2024; Padmakumar and He; Krishna et al., 2023). We aim to generate a summary s that allows for high coverage of Q associated with D. Each extracted key point captures an atomic claim or distinct piece of information, so we hypothesize that selecting diverse key points would lead to better coverage of Q.

3.2 Principled Key Point Selection

Given the set of all key points from all documents, $K = \bigcup_i K_i$, the next step involves selecting a subset of key points, K_{sel}, prioritizing coverage of source material for MDDS, additionally incorporating relevance for *query-focused* summarization.

3.2.1 Background on DPPs

Determinantal Point Processes (DPPs) model the probability of selecting subsets from a set of items emphasizing diversity among the chosen elements (Kulesza et al., 2012). DPPs construct a kernel matrix L using a similarity function between pairs of items. The kernel matrix may also be weighted by a diagonal matrix that scores the absolute quality or a task-specific property such as the relevance of the items (Kulesza et al., 2012). Inference from DPPs is formulated as a combinatorial optimization problem, where the goal is to find the subset of items with the highest likelihood under the kernel L. This can be efficiently approximated using greedy algorithms Chen et al. (2018). Our work uses DPP inference out of the box, noting that this allows the number of selected items to vary according to the similarity of items in the kernel matrix rather than a pre-specified number of distinct items. We provide more extensive coverage of prior work connecting DPPs with NLP tasks in Section 6.2.

3.2.2 Selecting Key Points Prioritizing Diversity

To achieve high source coverage in the MDDS task, we use a DPP to select a subset of key points from $K = \bigcup_i K_i$ that prioritizes diversity. Each key point k_{ij} is first embedded into a high-dimensional vector v_{ij} via a transformer-based encoder. These embeddings are then used to construct a kernel matrix L, where each entry $L_{(i_1,j_1),(i_2,j_2)}$ represents the similarity between pairs of key points, computed through a kernel function $f_k(v_{i_1 j_1}, v_{i_2 j_2})$. We then run DPP-inference on L to obtain the selected key points, K_{sel} as detailed in Section 3.2.1.

3.2.3 Selecting Relevant Key Points Prioritizing Diversity

In the *query-focused* MDDS task, we incorporate relevance to q_{user} into the key point selection objective, using a modified DPP approach. After embedding each key point k_{ij} into a vector v_{ij}, we construct the similarity matrix L as above. We then create the relevance vector R, where each entry R_i represents the relevance score of $k_i \in K$ calculated as $f_{\text{rel}}(k_i|q_{\text{user}})$.[2] The relevance-weighted matrix to $L' = RLR^T$ thus balances both key point similarity and relevance to q_{user}. where each entry in $L'_{(i_1,j_1),(i_2,j_2)} = f_{rel}(v_{i_1 j_1}|q_{user}) \times f_k(v_{i_1 j_1}, v_{i_2 j_2}) \times f_{rel}(v_{i_2 j_2}|q_{user})$. DPP inference is then applied to L' (Section 3.2.1), selecting a diverse yet query-relevant subset K_{sel}.

3.3 Rewriting

The final step involves synthesizing the selected key points into a coherent summary s. We use an LLM to rewrite the chosen subset, ensuring that the output is coherent and well-structured.

4 Experimental Setup

4.1 Datasets

4.1.1 DiverseSumm Benchmark

The DIVERSESUMM benchmark consists of 245 examples, each of which is a set 10 articles covering different aspects of the same news event. Each example is accompanied by 1 to 10 questions, with each question linked to a set of articles that provide answers. These articles offer diverse perspectives on the questions, and the objective is to produce a summary that captures the range of perspectives.

[2]We note here that the dimensionality of R is equal to the total number of key points across *all* source documents, the same as that of L.

4.1.2 Augmenting DiverseSumm with more questions

We observe that 78.3% of news stories in the original dataset have 3 or fewer associated questions. Thus not all articles are associated with questions in each example. To better evaluate the coverage of individual articles by the different methods, we use GPT-4o to generate 10 additional questions per article for each news story. This results in a synthetically augmented version of DIVERSESUMM with 100 questions per news story, sourced from the different articles.[3] The prompt to obtain these questions is provided in Appendix A.1. Unlike the original dataset, we do not expect these questions to have coverage across multiple articles, but this helps improve the statistical power of our comparison across methods. We report results on both the original, as well as augmented versions of the DIVERSESUMM dataset.

4.1.3 Augmenting DiverseSumm with synthetic user intents

Finally, to adapt DIVERSESUMM for a *query-focused* multi-document summarization task, we synthetically generate user intents to accompany each news story. These user intents reflect varied information needs, making certain perspectives from the source articles more or less relevant based on the intent. We prompt an LLM, again GPT-4o, to produce 5 distinct user intents for each news story given the concatenated set of 10 articles.[4] The prompt details for generating these user intents are provided in Appendix A.1.

4.2 Evaluation

Automatic Evaluation of Source Coverage To evaluate the coverage of generated summaries, we measure how many questions $q_i \in Q$ can be correctly answered based on the summary s. We evaluate if a question is answered using an LLM-as-judge evaluation with GPT-4o to (a) check whether a given question q_i is answerable from s, and (b) verify whether the answer from s aligns with the content in the corresponding article $d_j \in D$. A question q_i is *covered* by s if q_i is answerable from s and if the answer for q_i obtained from s matches the answer from d_j. We report the average coverage of examples from DIVERSESUMM and DIVERSESUMM Augmented (Section 4.1). Prior work has demonstrated the effectiveness of evaluation of question-answering tasks by prompting an LLM (Li et al., 2024; Balepur et al., 2024a). We select the prompt format per the recommendations of Huang et al. (2024) to evaluate the coverage of each question individually from the summary and the faithfulness of the answer to the original article, each via binary answers from an LLM. We provide the prompt used in Appendix A.6.

Correlation with Human Judgments To validate the reliability of our automatic evaluation, we cross-check a random sample of LLM-as-judge outputs from GPT-4o against human annotations collected via Amazon Mechanical Turk. We sample 100 outputs, equally split between cases where q_i is answerable from s and cases where it is not, obtaining 3 human annotations for each. The agreement between the LLM-as-judge and human annotations is 86.4% for answerability and 95.3% for correctness, demonstrating the robustness and reliability of the automatic evaluation method.[5]

4.3 Models Used

We perform experiments using four LLMs: GPT-3.5, GPT-4o (OpenAI, 2024), Claude-3-Sonnet (Anthropic, 2024a), and LLaMA 3.1 70B (Dubey et al., 2024).

Our Method (*LLM + DPP*) We first perform key point extraction from each article using the respective LLM (Section 3.1) with the prompt detailed in Appendix A.2. We then select the key points to be included in the summary (Section 3.2) using the DPPy library (Gautier et al., 2019).[6] We create the Gaussian kernel matrix L, using BertScore (Zhang* et al., 2020) with `Deberta-V3` embeddings (He et al.) as the similarity function between pairs of key points—we ablate aspects of the DPP kernel in Table 2. Additionally, we score the relevance of different key points to q_{user} using an instruction-tuned retrieval model, `intfloat/e5-mistral-7b-instruct` (Wang et al., 2023) model due to its strong performance on the MTEB leaderboard. The selected key points are then rewritten into the final summary using the

[3] To verify the quality of the augmented questions, we conduct a human annotation in Appendix B.1.

[4] We detail the method in which we filter generated user intents for quality as well as identify the set of relevant questions for evaluation in Appendix B.2. We also conduct a human annotation to verify the quality of the intents in Appendix B.2.

[5] We note that this agreement matches is in line with the reported performance of GPT-4 in an LLM-as-judge setting in MT-Bench (Zheng et al., 2023). However, we acknowledge the limitations of LLM-as-judge evaluation in Section 8.

[6] We perform exact sampling via the spectral method, the default inference technique via DPPy

same LLM (Section 3.3) with the prompt detailed in Appendix A.3.

Baselines (1) *Naive LLM*- A simple baseline where we prompt the LLM to generate the summary from the concatenated set of articles, performing content selection and text generation in one step. The prompt for *Naive LLM* is provided in Appendix A.4, (2) *All KPs*- To ablate the effect of our content selection methods, we compare to a baseline where we prompt the LLM to generate the summary from the set of all key points extracted from the articles. This uses the same prompt as *LLM + DPP* for rewriting (Appendix A.3), just without the selection step, and (3) *LLM-Selected KPs*- Finally, to demonstrate the effectiveness of the DPP-based key point selection method over an entirely LLM-prompting pipeline, we compare to a baseline that performs key point selection with an LLM (GPT-4o) before rewriting. The prompt used for LLM-based key point selection is provided in Appendix A.7. This uses the same prompt as *LLM + DPP* for rewriting (Appendix A.3).

5 Results

5.1 Evaluating Source Coverage in Multi-document Summarization

Content selection with DPPs results in better source coverage From Table 1, we observe that *LLM + DPP* consistently achieves the highest source document coverage across all evaluated LLMs, outperforming all baselines on both the DIVERSESUMM and DIVERSESUMM Augmented datasets. The baselines that explicitly select key points (*All KPs* and *LLM-Selected KPs*) generally outperform the naive approach of concatenating articles and prompting the LLM for a summary (*Naive LLM*) for all LLMs. Additionally, the consistent improvement of *LLM-Selected KPs* and *LLM + DPP* over *All KPs* indicates that simply reducing context length by extracting all key points is insufficient, explicit key point selection is important in order to obtain better coverage.

Encoded model representations of key points provide useful signal for key point selection From Table 2, we also observe that *LLM + DPP*, using variants of the DPP-kernel applied to high-dimensional encoder embeddings, outperforms *LLM-Selected KPs*, which performs on explicit key point selection in the text space through LLM prompting. This finding shows the value of using principled techniques, such as diversity-aware key point selection (Section 3.2), to perform individual steps in a pipeline instead of performing every step via an LLM prompt.

While LLMs selecting content have uneven coverage, key point selection is more uniform Prior work (Liu et al., 2024) has shown that LLMs have systematic biases in how well they attend to context, better answering questions when relevant information appears at the start or end of the context. Huang et al. (2024) also observe similar 'lost-in-the-middle' biases on the multi-document summarization task. To study this, we plot the coverage of the generated summaries from *LLM + DPP* and *Naive LLM* per article on DIVERSESUMM Augmented in Figure 2.[7] We observe that the *Naive LLM* approach exhibits systematic positional biases. Llama 3.1 has better coverage of documents at the end of the context, an *end* bias. Similarly, GPT-4o has a *start* bias, and GPT-3.5 and Claude exhibit mild biases to not sufficiently cover documents in the middle. *LLM + DPP* improves coverage on all documents, particularly alleviating the positional biases on Llama 3.1 and GPT-4o, highlighting the efficacy of key point selection in the multi-document summarization task.

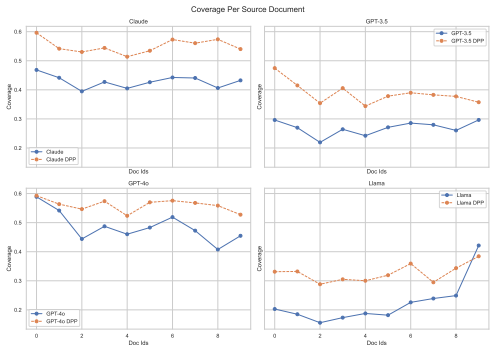


Figure 2: Studying the 'lost-in-the-middle' phenomenon by plotting coverage of different source articles by index with *Naive LLM* and *LLM + DPP*. While *Naive LLM* exhibits biases to better cover the articles at the *start* (GPT-4o, GPT-3.5) or *end* (Llama) of the context, *LLM + DPP* has higher and more uniform coverage of all source documents—mitigating these biases.

Key points selected in *LLM + DPP* better covers the source documents than *LLM-Selected KPs* To investigate the improved coverage of *LLM + DPP* over *LLM-Selected KPs*, we plot the distribu-

[7] We selected the augmented version of DIVERSESUMM since the synthetic question generation ensures that each article has at least 10 associated questions. This ensures we have statistical power on our results.

	DIVERSESUMM				DIVERSESUMM Augmented			
	GPT 3.5	GPT 4o	Claude	Llama	GPT 3.5	GPT 4o	Claude	Llama
Naive LLM	0.3324	0.5516	0.4776	0.2427	0.2667	0.4807	0.4248	0.2187
All KPs	0.3472	0.5443	0.5683	0.3458	0.2573	0.4620	0.4114	0.2368
LLM-Selected KPs	0.4370	0.5747	0.5369	0.3376	0.3849	0.5409	0.5142	0.3087
LLM + DPP	0.4706	0.5805	0.5923	0.3653	0.3845	0.5535	0.5469	0.3227

Table 1: Source coverage evaluation (Section 4.2) on DIVERSESUMM (Section 4.1.1) and DIVERSESUMM Augmented (Section 4.1.2). We report coverage of the source material as the fraction of questions correctly answered from the generated summaries (Section 4.2) from 4 different LLMs—GPT3.5, GPT-4o, Claude-3-Sonnet and Llama-3.1, and compare the performance of our method, *LLM + DPP*, with three relevant baselines (Section 4.3). Selecting key points to prioritize diversity via DPPs (Section 3.2) results in better source coverage for all 4 LLMs.

	DIVERSESUMM			DIVERSESUMM Augmented		
	GPT 3.5	GPT 4o	Claude	GPT 3.5	GPT 4o	Claude
LLM-Selected KPs	0.4370	0.5747	0.5369	0.3849	0.5409	0.5142
LLM + DPP (Gaussian Kernel, $\sigma = 0.1$)	0.4494	0.6145	0.6347	0.3728	0.6145	0.6037
LLM + DPP (Gaussian Kernel, $\sigma = 1$)	0.4706	0.5805	0.5923	0.3845	0.5535	0.5469
LLM + DPP (Gaussian Kernel, $\sigma = 10$)	0.4342	0.5906	0.5198	0.3752	0.5258	0.4699
LLM + DPP (Linear Kernel)	0.4653	0.5893	0.5863	0.3674	0.5518	0.5450

Table 2: We report 4 ablations of the DPP kernel used for keypoint selection (Section 3.2) for our method, *LLM + DPP*. We evaluate 3 LLMs on 4 different kernels for source coverage (Section 4.2).

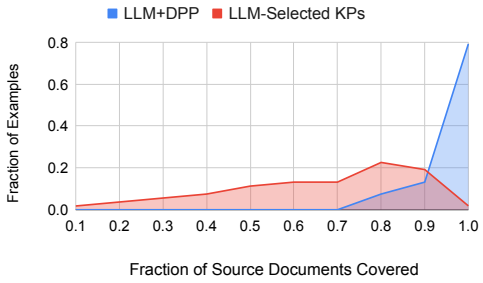


Figure 3: Distribution of source documents covered by key points when selected with *LLM + DPP* and *LLM-Selected KPs*. *LLM + DPP* exhibits consistently higher coverage of source documents.

tion of the fraction of source documents covered in the selected subsets of key points in Figure 3. While *LLM-Selected KPs* has a much higher variance of documents covered, *LLM + DPP* consistently achieves high coverage of the diverse source documents.[8]

DPP-based key point selection improves coverage without increasing summary length To investigate whether the improved source coverage achieved by *LLM + DPP* stems from better content selection rather than simply generating longer summaries—a potential confounder—we compare the average summary lengths across *LLM + DPP*, *Naive LLM*, and *LLM-Selected KPs* for each of the four LLMs analyzed (Table 3). We calculate the statistical significance of the differences in mean lengths using a two-tailed t-test. We observe no significant differences in average summary lengths for GPT-4o, GPT-3.5, and Llama indicating that the higher source coverage reported in Table 1 is not attributable to longer summaries in these.[9]

	GPT-4o	GPT-3.5	Llama	Claude
LLM + DPP	925.34	448.77	296.28	890.37
LLM-Selected KPs	929.33	414.13	290.71	706.50
Naive LLM	914.05	418.15	298.40	601.77

Table 3: Average length of summaries, in words, from *LLM + DPP*, *LLM-Selected KPs* and *Naive LLM* with various LLM. For GPT-4o, GPT-3.5, and Llama, we observe no significant difference across methods.

5.2 Evaluation of Coverage of Relevant Source Material in *Query-Focused* Multi-Document Summarization

Adapting DPPs to select relevant content to a user intent leads to better relevant coverage

[8]We perform tests for significance in Appendix C.2.

[9]For Claude, the differences in summary lengths across the various methods, at odds with the other LLMs, potentially stems from differences in model training—we note that Claude was specifically tuned for long contexts (Anthropic, 2024b). We believe that this differing behavior when interacting with different inputs for the rewriting step presents a direction for future exploration.

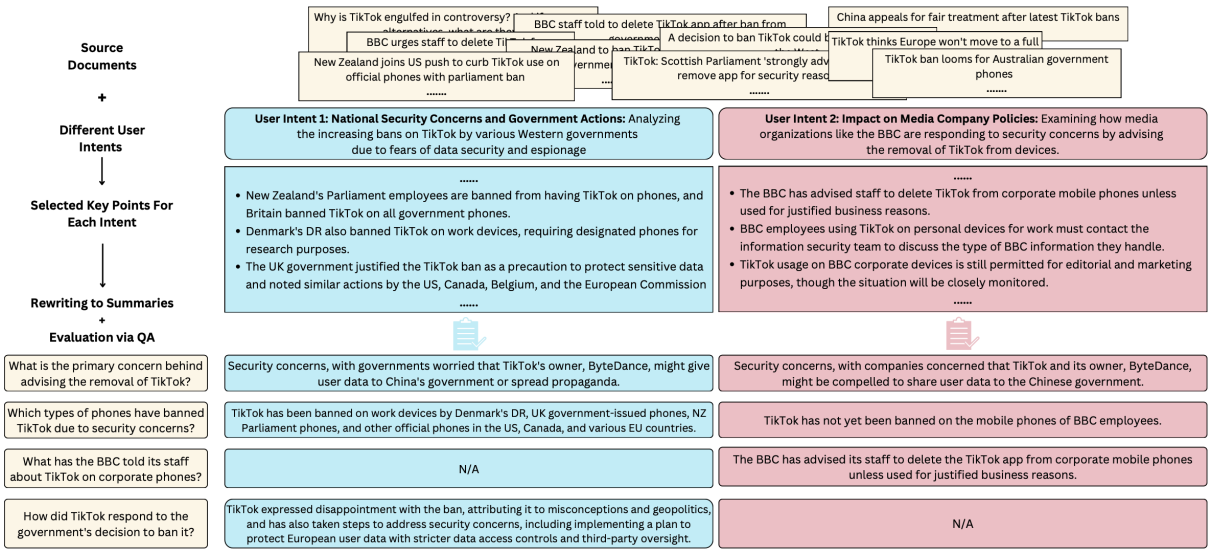
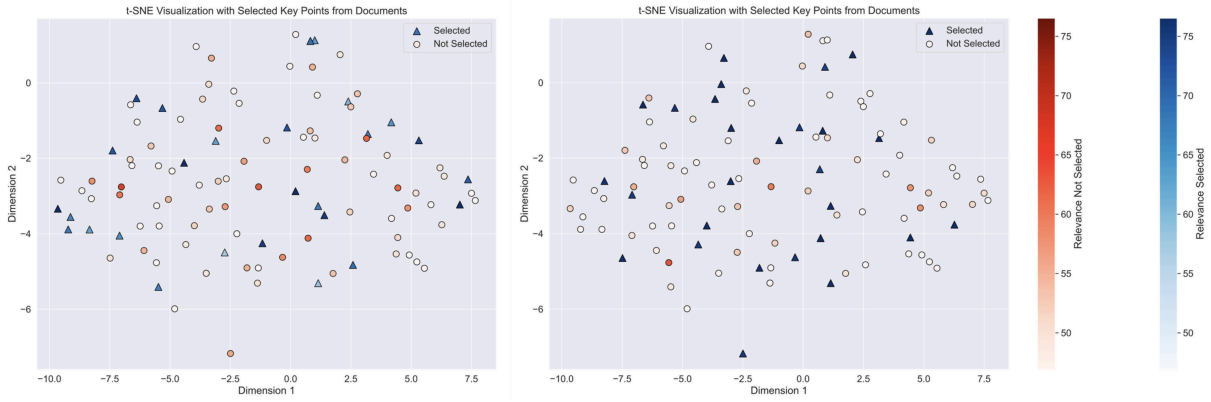


Figure 4: Case study of *LLM + DPP* (Section 5.2) selecting key points that are diverse and yet relevant to two different user intents (Section 3.2.3) and evaluation of the summaries via question-answering (Section 4.2).



(a) Intent 1 (b) Intent 2

Figure 5: TSNE visualization of the key points selected for the two user intents in Figure 4 from the document set. Blue triangles represent selected key points, while red circles denote unselected points. Color intensity reflects relevance to the respective user intent. *LLM + DPP* is able to select relevant key points while also prioritizing diverse coverage of the source material.

too In addition to ensuring better coverage of the source material, we also evaluate the effectiveness of our proposed method in covering content relevant to specific user intents using the DIVERSESUMM Relevance dataset (Section 4.1.3). This problem requires balancing diversity of content selected along with relevance to user intent. From Table 4, we find that adapting the DPP kernel to incorporate relevance (Section 3.2.3) leads to the highest performance compared to the various baselines. While prompting an LLM to directly generate summaries tailored to user intents (*LLM-Selected KPs*) yields improved relevance coverage compared to the naive summarization baseline (*Naive LLM*), our approach, which combines principled key point selection with relevance-aware DPPs, consistently outperforms both baselines.[10]

To further illustrate the effectiveness of selecting diverse yet relevant key points, we provide a qualitative case study. Figure 4 is an example of two distinct user intents associated with the same set of source documents, along with corresponding representative key points selected by *LLM + DPP*. As a result, the answers to evaluation questions (Section 4.2) differ based on the summaries rewritten from these selected key points. In Figure 5, we present a t-SNE visualization of key points from the source documents, embedded using the `intfloat/e5-mistral-7b-instruct` model (Section 4.3), that also highlights their relevance to the two user intents and marks those selected

[10]We use prompts Appendix A.5 and Appendix A.8 for *Naive LLM* and *LLM-Selected KPs* in Section 5.2.

	DIVERSESUMM Relevance			
	GPT 3.5	GPT 4o	Claude	Llama
Naive LLM	0.4080	0.6410	0.5843	0.3182
LLM-Selected KPs	0.5292	0.6443	0.6180	0.3603
LLM + DPP	0.5229	0.6605	0.6672	0.4224
LLM + DPP-Relevance	0.5409	0.6972	0.6937	0.4501

Table 4: Evaluation of coverage of *relevant* source material on DIVERSESUMM Relevance (Section 4.1.3). We compare the performance of *LLM + DPP* with two relevant baselines (Section 4.3) across various LLMs. Incorporating relevance into the DPP-kernel (Section 3.2.3) results in the highest coverage, improving over *LLM-Selected KPs* prompted to select relevant key points and *LLM + DPP* prioritizing diversity alone.

by *LLM + DPP*. We observe that *LLM + DPP* effectively balances diverse coverage across the latent space while maintaining high relevance to user queries.

6 Background and Related Work

6.1 Multi-Document Summarization

Our work builds on foundational multi-document summarization methods that extract information at various granularities (Radev et al., 2004; Hong and Nenkova, 2014; Cheng and Lapata, 2016) and abstractively summarize documents with specialized neural networks (McKeown and Radev, 1995; Radev and McKeown, 1998; Barzilay et al., 1999; Zhang et al., 2018; Fabbri et al., 2019; Song et al., 2022). This has been aided by various datasets (Over and Yen, 2004; Dang, 2005; Owczarzak and Dang, 2011; Fabbri et al., 2019; Lu et al., 2020), most recent of which is DIVERSESUMM (Huang et al., 2024). More recently, Bhaskar et al. (2023); Chang et al. (2024) prompt LLMs to hierarchically generate summaries. To et al. (2024) generate an extractive summary using K-means clustering of sentence embeddings and then rewrite it as an abstractive summary using a fine-tuned T5 model. With LLMs able to process longer contexts, Huang et al. (2024) primarily evaluate a version of the *Naive LLM* baseline reporting results on various models. Our work extends this line of research by integrating a prompting pipeline with a principled content selection mechanism using Determinantal Point Processes (DPPs). This approach allows us to combine the strong off-the-shelf generative capabilities of LLMs on the extraction and rewriting subtasks with a robust content selection strategy.

6.2 DPPs for Summarization

Earlier works that use DPPs for summarization tend to be extractive in nature. Kulesza et al. (2012) propose a method to use DPPs for selection of sentences to construct a summary that best resembles the reference in training data, computing the similarity kernel between sentences via TF-IDF scores. Cho et al. (2019b) propose to use DPPs to select sentences to construct an extractive summary based on a BERT-based similarity measure. Cho et al. (2019a) propose an enhanced similarity metric to further refine extractive summaries. Moving beyond sentence-level extraction, Perez-Beltrachini and Lapata (2021) introduced DPPs into the attention mechanisms of LSTMs and transformers for abstractive summarization, encouraging diversity in attending to input tokens during generation. Our method requires no additional fine-tuning, as we make no changes to the model architecture or objective function, unlike previous abstractive methods, allowing us to reap the benefits from further advancements in language modeling. Unlike existing extractive methods, which focus on selecting context-dependent sentences from the documents, we operate on context-independent key points to ensure more high-quality content selection.

6.3 Further applications of DPPs

DPPs are used in recommender systems when diversity in retrieved items is desirable (Gan et al., 2020; Wilhelm et al., 2018). DPPs are also used to select diverse and high-quality in-context learning examples leading to improved performance when prompting LLMs (Wang et al., 2024; Ye et al., 2023; Yang et al., 2023). Finally, DPPs have also been used to help search the prompt space, thereby eliciting jailbreaks of LLMs (Zhang et al., 2024).

7 Conclusion

In this work, we demonstrate the utility of explicit content selection for improving the coverage of diverse sources on the DIVERSESUMM benchmark. Creating a pipeline that uses LLM prompting steps, for extracting and rewriting information, combined with principled key point selection with DPPs yields summaries that cover diverse source material as well as can be personalized to different user intents. As agentic workflows are increasingly deployed for complex tasks, our findings highlight the need to identify and incorporate principled techniques and tools as a complement to powerful LLMs in order to best suit user needs.

8 Limitations

Firstly, we note the limitations of automatically evaluating coverage on DIVERSESUMM with an LLM. While ultimately the gold standard, conducting human evaluations for all ablations is prohibitively expensive, particularly as our task would require annotators to review entire news articles. We followed the evaluation recommendations from Huang et al. (2024) and supplemented our automatic evaluation with human validation of the metrics in Appendix B. Another limitation of this project is that we run experiments on only one dataset, with synthetic augmentations. The main reason for this is that we are intentionally looking for datasets that involve long documents with diverse source material. The challenge with many other summarization datasets is that LLMs already obtain fairly high performance when compared against the references (Goyal et al., 2022). It is yet unclear if our findings would generalize beyond the news domain, and to other languages. We do not make an exhaustive comparison with all possible prompting pipelines for multi-document summarization. Our research question in this project is about evaluating the role of principled content selection in improving coverage so we compare to baselines that do this implicitly (*Naive LLM*) or via an LLM prompt (*LLM-Selected KPs*). It is unclear if this is the maximum performance that can be obtained on the task with a multi-step LLM pipeline. One potential risk from our pipeline is that in Section 3.2.2, we select key points purely based on diversity—we do not incorporate any information about the reliability of the particular news articles. Since our work is purely academic, with publicly available datasets, this is not as much an issue but incorporating reliability into systems is important if deployed with real users.

Acknowledgments

References

Anthropic. 2024a. Claude 3.5 Sonnet. Technical report, Anthropic. Accessed: 2024-06-23.

Anthropic. 2024b. The Claude 3 Model Family: Opus, Sonnet, Haiku. Technical report, Anthropic. Accessed: 2024-05-23.

Nishant Balepur, Feng Gu, Abhilasha Ravichander, Shi Feng, Jordan Boyd-Graber, and Rachel Rudinger. 2024a. Reverse question answering: Can an llm write a question so hard (or bad) that it can't answer? *arXiv preprint arXiv:2410.15512*.

Nishant Balepur, Vishakh Padmakumar, Fumeng Yang, Shi Feng, Rachel Rudinger, and Jordan Boyd-Graber. 2024b. Whose boat does it float? improving personalization in preference tuning via inferred user personas.

Regina Barzilay, Kathleen McKeown, and Michael Elhadad. 1999. Information fusion in the context of multi-document summarization. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics*, pages 550–557.

Adithya Bhaskar, Alex Fabbri, and Greg Durrett. 2023. Prompted opinion summarization with gpt-3.5. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 9282–9300.

Yapei Chang, Kyle Lo, Tanya Goyal, and Mohit Iyyer. 2024. Booookscore: A systematic exploration of book-length summarization in the era of LLMs. In *The Twelfth International Conference on Learning Representations*.

Laming Chen, Guoxin Zhang, and Eric Zhou. 2018. Fast greedy map inference for determinantal point process to improve recommendation diversity. *Advances in Neural Information Processing Systems*, 31.

Jianpeng Cheng and Mirella Lapata. 2016. Neural summarization by extracting sentences and words. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 484–494, Berlin, Germany. Association for Computational Linguistics.

Sangwoo Cho, Logan Lebanoff, Hassan Foroosh, and Fei Liu. 2019a. Improving the similarity measure of determinantal point processes for extractive multi-document summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1027–1038, Florence, Italy. Association for Computational Linguistics.

Sangwoo Cho, Chen Li, Dong Yu, Hassan Foroosh, and Fei Liu. 2019b. Multi-document summarization with determinantal point processes and contextualized representations. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pages 98–103, Hong Kong, China. Association for Computational Linguistics.

Hoa Trang Dang. 2005. Overview of duc 2005. In *Proceedings of the document understanding conference*, volume 2005, pages 1–12. Citeseer.

Hal Daumé III and Daniel Marcu. 2006. Bayesian query-focused summarization. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 305–312.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Alexander Richard 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.

Lu Gan, Diana Nurbakova, Léa Laporte, and Sylvie Calabretto. 2020. Enhancing recommendation diversity using determinantal point processes on knowledge graphs. SIGIR '20, page 2001–2004, New York, NY, USA. Association for Computing Machinery.

Guillaume Gautier, Guillermo Polito, Rémi Bardenet, and Michal Valko. 2019. Dppy: Dpp sampling with python. *Journal of Machine Learning Research*, 20(180):1–7.

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

Pengcheng He, Jianfeng Gao, and Weizhu Chen. Debertav3: Improving deberta using electra-style pretraining with gradient-disentangled embedding sharing. In *The Eleventh International Conference on Learning Representations*.

Kai Hong and Ani Nenkova. 2014. Improving the estimation of word importance for news multi-document summarization. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 712–721.

Kung-Hsiang Huang, Philippe Laban, Alexander Richard Fabbri, Prafulla Kumar Choubey, Shafiq Joty, Caiming Xiong, and Chien-Sheng Wu. 2024. Embrace divergence for richer insights: A multi-document summarization benchmark and a case study on summarizing diverse information from news articles. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 570–593.

Gregory Kamradt. 2023. Needle In A Haystack - pressure testing LLMs. *Github*.

Marzena Karpinska, Katherine Thai, Kyle Lo, Tanya Goyal, and Mohit Iyyer. 2024. One thousand and one pairs: A" novel" challenge for long-context language models. *arXiv preprint arXiv:2406.16264*.

Yekyung Kim, Yapei Chang, Marzena Karpinska, Aparna Garimella, Varun Manjunatha, Kyle Lo, Tanya Goyal, and Mohit Iyyer. 2024. Fables: Evaluating faithfulness and content selection in book-length summarization. In *Conference on Language Modeling*.

Kalpesh Krishna, Erin Bransom, Bailey Kuehl, Mohit Iyyer, Pradeep Dasigi, Arman Cohan, and Kyle Lo. 2023. Longeval: Guidelines for human evaluation of faithfulness in long-form summarization. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1650–1669.

Alex Kulesza, Ben Taskar, et al. 2012. Determinantal point processes for machine learning. *Foundations and Trends® in Machine Learning*, 5(2–3):123–286.

Zongxia Li, Ishani Mondal, Huy Nghiem, Yijun Liang, and Jordan Lee Boyd-Graber. 2024. PEDANTS: Cheap but effective and interpretable answer equivalence. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 9373–9398, Miami, Florida, USA. Association for Computational Linguistics.

Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173.

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.

Kathleen McKeown and Dragomir R Radev. 1995. Generating summaries of multiple news articles. In *Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 74–82.

OpenAI. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Paul Over and James Yen. 2004. An introduction to duc-2004. *National Institute of Standards and Technology*.

Karolina Owczarzak and Hoa Trang Dang. 2011. Overview of the tac 2011 summarization track: Guided task and aesop task. In *Proceedings of the Text Analysis Conference (TAC 2011), Gaithersburg, Maryland, USA, November*.

Vishakh Padmakumar and He He. Does writing with language models reduce content diversity? In *The Twelfth International Conference on Learning Representations*.

Richard Yuanzhe Pang, Alicia Parrish, Nitish Joshi, Nikita Nangia, Jason Phang, Angelica Chen, Vishakh Padmakumar, Johnny Ma, Jana Thompson, He He, et al. 2022. Quality: Question answering with long input texts, yes! In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5336–5358.

Laura Perez-Beltrachini and Mirella Lapata. 2021. Multi-document summarization with determinantal point process attention. *Journal of Artificial Intelligence Research*, 71:371–399.

Dragomir R Radev, Hongyan Jing, Małgorzata Styś, and Daniel Tam. 2004. Centroid-based summarization of multiple documents. *Information Processing & Management*, 40(6):919–938.

Dragomir R Radev and Kathleen R McKeown. 1998. Generating natural language summaries from multiple on-line sources. *Computational Linguistics*, 24(3):470–500.

Machel Reid, Nikolay Savinov, Denis Teplyashin, Dmitry Lepikhin, Timothy Lillicrap, Jean-baptiste Alayrac, Radu Soricut, Angeliki Lazaridou, Orhan Firat, Julian Schrittwieser, et al. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*.

Uri Shaham, Elad Segal, Maor Ivgi, Avia Efrat, Ori Yoran, Adi Haviv, Ankit Gupta, Wenhan Xiong, Mor Geva, Jonathan Berant, et al. 2022. Scrolls: Standardized comparison over long language sequences. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 12007–12021.

Yun-Zhu Song, Yi-Syuan Chen, and Hong-Han Shuai. 2022. Improving multi-document summarization through referenced flexible extraction with credit-awareness. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1667–1681, Seattle, United States. Association for Computational Linguistics.

Yi Tay, Mostafa Dehghani, Samira Abnar, Yikang Shen, Dara Bahri, Philip Pham, Jinfeng Rao, Liu Yang, Sebastian Ruder, and Donald Metzler. 2021. Long range arena : A benchmark for efficient transformers. In *International Conference on Learning Representations*.

Huy Quoc To, Hung-Nghiep Tran, Andr'e Greiner-Petter, Felix Beierle, and Akiko Aizawa. 2024. Skt5scisumm-a hybrid generative approach for multi-document scientific summarization. *arXiv preprint arXiv:2402.17311*.

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

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2023. Improving text embeddings with large language models. *arXiv preprint arXiv:2401.00368*.

Peng Wang, Xiaobin Wang, Chao Lou, Shengyu Mao, Pengjun Xie, and Yong Jiang. 2024. Effective demonstration annotation for in-context learning via language model-based determinantal point process. *arXiv preprint arXiv:2408.02103*.

Mark Wilhelm, Ajith Ramanathan, Alexander Bonomo, Sagar Jain, Ed H. Chi, and Jennifer Gillenwater. 2018. Practical diversified recommendations on youtube with determinantal point processes. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, CIKM '18, page 2165–2173, New York, NY, USA. Association for Computing Machinery.

Zhao Yang, Yuanzhe Zhang, Dianbo Sui, Cao Liu, Jun Zhao, and Kang Liu. 2023. Representative demonstration selection for in-context learning with two-stage determinantal point process. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5443–5456, Singapore. Association for Computational Linguistics.

Jiacheng Ye, Zhiyong Wu, Jiangtao Feng, Tao Yu, and Lingpeng Kong. 2023. Compositional exemplars for in-context learning. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 39818–39833. PMLR.

Jianmin Zhang, Jiwei Tan, and Xiaojun Wan. 2018. Towards a neural network approach to abstractive multi-document summarization. *arXiv preprint arXiv:1804.09010*.

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

Xu Zhang, Dinghao Jing, and Xiaojun Wan. 2024. Enhancing jailbreak attacks with diversity guidance. *Preprint*, arXiv:2403.00292.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Tianle Li, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zhuohan Li, Zi Lin, Eric Xing, Joseph E. Gonzalez, Ion Stoica, and Hao Zhang. 2024. LMSYS-chat-1m: A large-scale real-world LLM conversation dataset. In *The Twelfth International Conference on Learning Representations*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36:46595–46623.

A Prompts Used

All prompting experiments were done by sampling from the LLM with temperature 0.7. We run inference on Claude-3-Sonnet, GPT-4o and GPT3.5 via their APIs and Llama 3.1 with model parallelism on three A100 GPUs.

A.1 Augmenting DIVERSESUMM with synthetic questions

```
Write down 10 factual questions that can
be answered from the article below. These
questions, and their answer should relate
the most important facts of the event
being reported in the article. Include
questions that require reasoning about
the facts in the document.
Make sure you create questions
such that all the important information
in the document appears in the answers.
Each question should be up to 14 words.
Return a numbered list of questions
with answers and nothing else.
Article:
<ARTICLE>
```

A.2 Generating key points from articles

```
Summarize all the content in this article
into a list of simple, one-sentence,
bullet points. Make sure that each bullet
point is atomic and can be understood
without any external context. Also, make
sure that all the information in the
article is covered in the list.
Article:
<ARTICLE>
```

A.3 Rewriting the set of selected key points into a coherent summary

```
Read the following set of key points
obtained from a set of news stories about
a specific topic. From the set, you have
a subset of selected key points. Rewrite
the selected key points into a coherent
report that includes all the details
present in the key points. Make sure the
summary is fluent and coherent. Elaborate
when you summarize diverse or conflicting
information. Make sure to include all of
the factual details from the key points
because we want to use the report to
answer questions. Remember, your output
should be a summary that discusses and
elaborates on the diverse and conflicting
information presented across the articles.
You need to elaborate on the differences
rather than only mentioning which topic
they differ. Don't worry about the summary
being too lengthy. You must give your
response in a structured format:
```Report: [your report]```, where
[your report] is your generated report.
--------
SELECTED KEY POINTS
--------
<SELECTED KEYPOINTS>
```

A.4 *Naive LLM* baseline prompt

We largely reuse the prompt as provided by Huang et al. (2024).

```
Read the following news articles. Produce
a summary that only covers diverse
and conflicting information across the
following articles, without discussing
the information all articles agree upon.
Elaborate when you summarize diverse or
conflicting information by stating what
information different sources cover and
how is the information diverse or
conflicting. You must give your answer in a
structured format: ```Report:
[your report]```, where [your report] is
your generated report.
---------
ARTICLES
<ARTICLES>
---------
Remember, your output should be a summary
that discusses and elaborates on the
diverse and conflicting information
presented across the articles. You need
to elaborate on the differences rather
than only mentioning which topic they
differ. Don't worry about the summary
being too lengthy.
```

A.5 *Naive LLM* baseline prompt with relevance

```
Read the following news articles and
associated user intent. Produce
a summary that only covers the diverse
and conflicting information across the
following articles relevant to the user
intent, without discussing
the information all articles agree upon.
Elaborate when you summarize diverse or
conflicting information by stating what
information different sources cover and
how is the information diverse or
conflicting. Balance diversity of content
```
```

with relevance to user intent. You must give your answer in a structured format: ```Report: [your report]```, where [your report] is your generated report.

ARTICLES
<ARTICLES>

USER INTENT
<USER INTENT>

Remember, your output should be a summary that is relevant to the user intent and discusses and elaborates on the diverse and conflicting information presented across the articles. You need to elaborate on the differences rather than only mentioning which topic they differ. Don't worry about the summary being too lengthy.

A.6 Evaluation of source coverage

Please act as an impartial judge and evaluate the quality of the response provided by an AI assistant. Your evaluation should consider coverage of the summary with regard to the question and answers (i.e. how much information in the question and answers is covered by the summary). Begin your evaluation by deciding if the question is answerable from the summary - this should be a true or false answer. Be as objective as possible. You next need to evaluate if the information to answer a question from the summary matches the reference answer. The answer to whether the answer matches should be “0” for insufficient coverage, and 1 indicates sufficient coverage. The output should strictly be in the format of a JSON with two keys, 'answerable' with the value true or false, and 'coverage' with the answer 0 or 1. Return nothing else.

Model Generated Response:
<SUMMARY>

Question:
<QUESTION>

Reference Answer:
<REFERENCE ANSWER>

A.7 *LLM-Selected KPs* baseline prompt

Read the following set of key points obtained from a set of news stories about a specific topic. From the set, you have a select a subset that ensure maximum coverage of the articles provided.
Make sure that all the important factual details from the articles are covered in the selected key points. Ensure that you cover all of the diverse viewpoints mentioned in the articles. Your output should be a list of selected key points where each selected one identically matches the corresponding key point
You must give your response in a structured format: ```Selected Key Points: [your list]```.

KEY POINTS
<ALL KEYPOINTS>

ARTICLES
<ARTICLES>

A.8 *LLM-Selected KPs* baseline prompt with relevance

Read the following set of key points obtained from a set of news stories about a specific topic and the associated user intent. From the set, you have a select a subset that are relevant to the user intent and ensure maximum coverage of the articles provided.
Make sure that all the important factual details from the articles that are relevant to the user intent are covered in the selected key points. Ensure that you cover all of the diverse viewpoints mentioned in the articles. Your output should be a list of selected key points where each selected one identically matches the corresponding key point
You must give your response in a structured format: ```Selected Key Points: [your list]```.

KEY POINTS

```
<ALL KEYPOINTS>
---------
ARTICLES
<ARTICLES>
---------
USER INTENT
<USER INTENT>
---------
```

B Validation of DIVERSESUMM Augmented and DIVERSESUMM Relevance

In order to confirm that our synthetic augmentations of DIVERSESUMM are valid, we perform an additional human annotation. The annotators for this task were volunteer PhD students recruited from our university in the US.

B.1 Confirming that the questions generated in Section 4.1.2 are valid

We randomly sample 100 LLM-generated question-answer pairs and the corresponding articles from which they were generated. Two separate human annotators independently provide a binary annotation that the question can indeed be answered by the article in question. Both annotators agree that the generated questions are answerable in 98% of cases. They are also asked to score if the provided answer correctly answers the question given the article. Agreement on the correctness of the provided answer is 93%.

B.2 Filtering of synthetic user intents generated in Section 4.1.3

To create the query-focused version of the dataset, we prompt the model to generate 5 distinct user intents. For each of the intents, we identify the set of relevant questions by scoring the relevance of all DIVERSESUMM-Augmented questions to that particular intent with the trained `intfloat/e5-mistral-7b-instruct` model. We select this model due to its strong performance on the MTEB leaderboard. We set the threshold as 0.6 above which a question is deemed relevant. We retain all user intents that contain at least 20 different relevant questions associated with them. As a result, the average number of user intents evaluated per example is 4.65 with a minimum of 2 and a mode of 5.

Confirming the validity of synthetic user intents generated in Section 4.1.3 We randomly sample 50 examples, and the associated user intents, out of those that maintain 5 intents after filtering (Appendix B.2). These are independently annotated by two separate human annotators. Each annotator provides a score from 1-5 to assess that each individual intent is valid given the set of input documents. The mean rating assigned to the generated user intents is 4.35 out of 5, with a Cohen's Kappa of 0.64 indicating moderate to high agreement. This value also corresponds with the score assigned for *Applicability* of LLM-generated user personas in (Balepur et al., 2024b). We then ask the annotators to score the effective number of distinct personas out of the provided 5, an integer value from 1 to 5. Annotators report an average value of 3.56 indicating that further exploration is necessary in order to synthetically create diverse user intents.

C Additional Results

C.1 The latent representations also contain useful information over selecting key points with uniform random sampling

From Section 5.1, we observe that prioritizing diversity when selecting key points leads to high coverage in summaries, more uniformly covering all the different source documents. However, uniform random sampling is another way in which we can, in theory, cover each source document. We concatenate key points from all the documents and then randomly sample k of them, before rewriting these into the summary using the prompt in Appendix A.3. We then compare this baseline with one that selects k key points using a k-DPP to represent these. From Figure 6, we see that, for the same number of key points, the k-DPP baseline fairly consistently outperforms uniform random sampling. This again highlights the value in using the learned representations to select key points as it allows our method to sample prioritizing the relative similarities of different key points. Finally, we note that both these methods are comfortably outperformed by the *LLM + DPP* baseline, essentially a DPP with exact sampling as detailed in Section 3.2.2. The main difference is that exact sampling sets the number of key points to be selected by considering the nature of the latent space of the key points, and not as a hyperparameter input to the method. This confirms the benefit of combining LLM-prompting with principled techniques as appropriate to achieve high performance on tasks such as DIVERSESUMM.

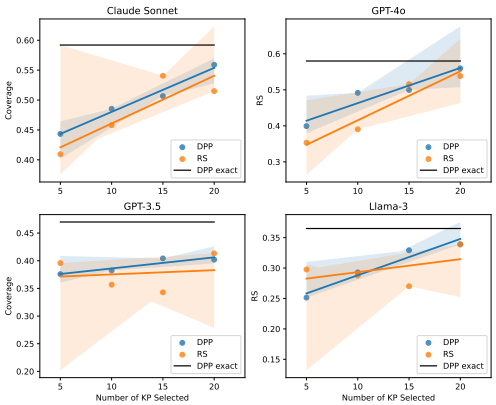


Figure 6: Comparing using a k-DPP with uniform random sampling for key point selection for DIVERSESUMM, varying k, across 4 different LLMs (Appendix C). The k-DPP consistently outperforms uniform random sampling, showing the value in sampling while considering the learned representations of key points. We also note that both are outperformed by *LLM + DPP* with exact sampling.

C.2 Tests for statistical significance

To evaluate the significance of the coverage improvements shown in Table 1, we perform a two-tailed t-test comparing the mean coverage of *LLM + DPP* to *LLM-Selected KPs*, the highest-performing baseline, for DIVERSESUMM and DIVERSESUMM-Augmented. We find that *LLM + DPP* achieves significantly higher coverage than *LLM-Selected KPs* for GPT-3.5, Claude, and Llama 3.1 at the 5% significance level ($p < 0.05$). For DIVERSESUMM-Augmented, the improvement is significant for all four LLMs, likely due to increased statistical power from the larger sample size. Similarly, for Table 4, we perform a two-tailed t-test comparing *LLM + DPP* with and without relevance in the DPP-kernel. Incorporating relevance leads to significantly higher coverage ($p < 0.05$) across all LLMs.