Scaling Multi-Document Event Summarization: Evaluating Compression vs. Full-Text Approaches

Adithya Pratapa   **Teruko Mitamura**

Language Technologies Institute
Carnegie Mellon University
{vpratapa, teruko}@cs.cmu.edu

Abstract

Automatically summarizing large text collections is a valuable tool for document research, with applications in journalism, academic research, legal work, and many other fields. In this work, we contrast two classes of systems for large-scale multi-document summarization (MDS): compression and full-text. Compression-based methods use a multi-stage pipeline and often lead to lossy summaries. Full-text methods promise a lossless summary by relying on recent advances in long-context reasoning. To understand their utility on large-scale MDS, we evaluated them on three datasets, each containing approximately one hundred documents per summary. Our experiments cover a diverse set of long-context transformers (Llama-3.1, Command-R, Jamba-1.5-Mini) and compression methods (retrieval-augmented, hierarchical, incremental). Overall, we find that full-text and retrieval methods perform the best in most settings. With further analysis into the salient information retention patterns, we show that compression-based methods show strong promise at intermediate stages, even outperforming full-context. However, they suffer information loss due to their multi-stage pipeline and lack of global context. Our results highlight the need to develop hybrid approaches that combine compression and full-text approaches for optimal performance on large-scale multi-document summarization.[1]

1 Introduction

Summarizing events described in document collections has long interested the NLP community with shared tasks for event tracking (Allan et al., 1998) and summarization (Chieu and Lee, 2004; Dang and Owczarzak, 2009; Aslam et al., 2015). Given an input collection of hundreds of text documents, systems have to extract and summarize salient information about the event. The length and diversity of the input presents a challenge to recent large language models (LLMs). In this work, we contrast two classes of systems for large-scale multi-document summarization (MDS), compression-based, and full-text systems.[2]

Full-text systems promise a lossless approach by providing the summarizer access to the entire input. They are based on the long-context reasoning abilities of LMs, having already shown strong retrieval performance on long inputs (Hsieh et al., 2024). However, their capabilities on large-scale MDS are not as well understood. In a recent work, Laban et al. (2024) introduced a synthetic MDS benchmark that resembles the Needle in a Haystack evaluation (Kamradt, 2023). In addition to this dataset, we evaluate on two large-scale event summarization datasets: Background (Pratapa et al., 2023) and WCEP (Gholipour Ghalandari et al., 2020). We contrast the end-to-end full-context method[3] with three compression-based methods: retrieval, hierarchical, and incremental. Each method *compresses* the input in a multistage pipeline (§2.2). We evaluated the content selection aspects of the summary using the Atomic Content Unit (A3CU) metric (Liu et al., 2023b).

Our experiments show that full-context and retrieval perform best in most settings (§3). To better understand the performance of compression-based methods, we measure A3CU recall to track the salient information retention in their intermediate outputs (§3.4). Across all settings, we find that compression-based methods show high recall in intermediate stages but suffer information loss in their multistage pipeline. In particular, the intermediate recall is often much higher than the full-context system recall. We highlight two key takeaways: First, while iterative methods (hierarchical & incremental) were previously found effec-

---

[1] Our code and data are available at https://github.com/adithya7/scaling-mds.

[2] We use the term *scale* to refer to the large number of documents associated with each summary.

[3] We use full-text and full-context interchangeably.

tive for book summarization and small-scale MDS, they underperform on large-scale MDS. Second, full-context systems are suboptimal on large-scale MDS datasets. We advocate for hybrid methods that combine input compression and long-context models. Such hybrid approaches are also scalable to even larger MDS tasks that go far beyond the context window limits of current LLMs.

2 Experimental Setup

2.1 Datasets

Our three datasets provide different flavors of the multi-document summarization task (Table 1).

SummHay: A query-focused dataset that covers the news and conversation domains (Laban et al., 2024). Synthetically generated using GPT-3.5 and GPT-4o, each summary constitutes a set of insights. To keep our evaluation setup consistent across datasets, we concatenate these insights into a free-form summary. Following the original work, we include an oracle setting that only retains documents containing the reference insights.

Background: This dataset provides summaries of complex news events (Pratapa et al., 2023). The task is based on an event timeline. For a given day, the goal is to generate a background summary by summarizing past new articles related to the event. We expand the original dataset to use news articles instead of just news updates. The dataset includes three human-written background summaries.

WCEP: A newswire dataset collected from Wikipedia Current Events Portal (Gholipour Ghalandari et al., 2020). The summaries come from the portal and the documents include a combination of cited source articles and a retrieved collection of related articles from the Common Crawl archive.

Our choice of datasets collectively represents the real-world use-cases of multi-document summarization systems. Previous work has shown the effectiveness of full-context methods in retrieval tasks. To this end, we include the query-focused SummHay dataset. On the other hand, Background and WCEP provide different variants of the task. Background task requires accumulation of salient content units over the entire input. WCEP has high information redundancy, with many articles providing support for the salient units.

2.2 Methods

We now describe our long-context methods and transformers. The key difference between our meth-

Dataset	# Ex.	# Docs/Ex.	Avg. length	
			Doc.	Summ.
SummHay	92	100	884	185
Background	658	186	1033	174
WCEP	1020	76	468	34

Table 1: An overview of our multi-document summarization datasets. We report the number of examples in the test set, and average statistics for # documents per example, document and summary lengths (words).

ods is the length of the input passed to the summarization system (transformer) at any stage.

Full-context: The transformer has access to the full input and relies on its long context reasoning abilities to generate the summary.

Iterative: Multi-stage summarization where we iteratively pass chunks of the input to the transformer. We explore two methods, hierarchical and incremental. The hierarchical method summarizes each document and iteratively merges these to compile the final summary. The incremental method processes documents in order while maintaining a running summary of the input. Previous work explored these methods for book summarization (Chang et al., 2024) and small-scale multi-document summarization (Ravaut et al., 2024).

Retrieval: We rank the input documents according to their relevance to the query.[4] We then select the top-ranked documents (up to 32k tokens) and pass their concatenation to the transformer. We use SFR Embedding-2 (Meng* et al., 2024) for the retrieval task and order-preserving RAG following the recommendation from Yu et al. (2024). We set 32k as the limit because all of our transformers are effective at this context length (Hsieh et al., 2024).

2.3 Transformers

For our summarization systems, we experiment with three transformer-based models, Llama-3.1, Command-R, and Jamba-1.5. Each model supports a context window of at least 128k tokens. They rely on a different long-context methodologies, and represent the broad class of open-weight LLMs. All the three models show competitive performance on the RULER benchmark for long-context LMs (Hsieh et al., 2024).

Llama-3.1: Pretrained on 15T+ tokens, it supports long context by using a large base frequency

---

[4]If a query is unavailable, we default to using 'Generate a summary of the document' as the query.

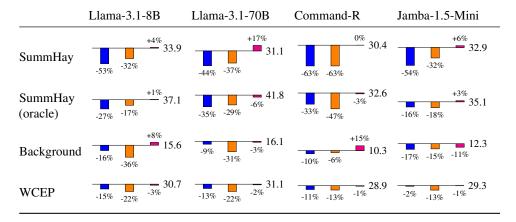|	Llama-3.1-8B	Llama-3.1-70B	Command-R	Jamba-1.5-Mini
SummHay	33.9 (+4%); −53%, −32%	31.1 (+17%); −44%, −37%	30.4 (0%); −63%, −63%	32.9 (+6%); −54%, −32%
SummHay (oracle)	37.1 (+1%); −27%, −17%	41.8 (−6%); −35%, −29%	32.6 (−3%); −33%, −47%	35.1 (+3%); −16%, −18%
Background	15.6 (+8%); −16%, −36%	16.1 (−3%); −9%, −31%	10.3 (+15%); −10%, −6%	12.3 (−11%); −17%, −15%
WCEP	30.7 (−3%); −15%, −22%	31.1 (−2%); −13%, −22%	28.9 (−1%); −11%, −13%	29.3 (−1%); −2%, −13%

Table 2: Performance of hierarchical, incremental and retrieval methods relative to the full-context baseline.

of 500,000 and non-uniform scaling of RoPE dimensions (Meta, 2024). We use both 8B and 70B variants to test the effect of model scaling.

Command-R: A transformer-based model that uses NTK-aware interpolation with a very large RoPE base frequency of 4M (Cohere For AI, 2024). We use the 32B variant.

Jamba-1.5: A hybrid architecture with interleaved Transformer and Mamba layers (Team et al., 2024). It involves both mid-training on long texts and post-training on (synthetic) long-context tasks. We use the 52B Jamba-1.5-Mini mixture-of-experts model with 12B active parameters.

For a fair comparison of above methods and transformers, we set the maximum input length to 128k across all settings. If the input is longer than 128k tokens, we first truncate the longest documents. In the case of Background, we also ensure equal representation from the past events by budgeting the token limit to each past timestamp. We also set a minimum document length (128 tokens) and drop documents if this cannot be achieved. To ensure that all methods see the same input, we adopt the same truncation strategy across full-text and compression-based methods. Theoretically, compression-based methods could work with even longer input (>128k), but we limit all settings to 128k tokens for a fair comparison.

See §A.2 in the Appendix for additional details about our experimental setup including our summarization prompt (Table 4). We sample summaries with a temperature of 0.5. We note that the summaries could be slightly different across different seeds. Vig et al. (2022) compared end-to-end and RAG for query-focused summarization, but limited to the short input setting.

3 Results

3.1 Metrics

We focus our analysis on the *content selection* aspect of summarization. Nenkova and Passonneau (2004) first studied the content selection evaluation using the pyramid method on summarization of content units. Follow-up efforts have automated various parts of this method (Shapira et al., 2019; Liu et al., 2023b). In this work, we use the reference-based Atomic Content Unit (A3CU) metric (Liu et al., 2023b) that is based on the definition of atomic content units of Liu et al. (2023a). This metric is trained to predict a score that measures the overlap of atomic content units between the reference and predicted summaries.

Recent works also studied faithfulness (Kim et al., 2024), coherence (Chang et al., 2024), and position bias (Huang et al., 2024; Ravaut et al., 2024; Laban et al., 2024). Although these evaluations are important, content selection remains a core issue for large-scale MDS.

3.2 Overall Results

Table 2 reports the A3CU F1 scores for compression-based methods relative to the full-context baseline.[5] Full-context and retrieval perform the best, being particularly effective on the query-focused SummHay dataset. The two iterative methods perform poorly in most settings. We also find that the performance of transformers and methods varies considerably across the datasets and even within examples in each dataset.[6] Below, we break down these results and analyze the effect of transformer and compression methods.

---

[5]We report ROUGE and A3CU precision, recall in §A.3.
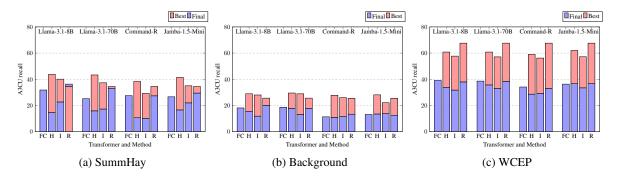[6]See Figure 3 in the Appendix for example-level trends.

Figure 1: Salient information retention in the intermediate and final summaries (A3CU *recall*). For each compression method, we report the best recall from the intermediate outputs and the recall of the final summary. (H: hierarchical, I: incremental, R: retrieval, FC: full-context)

Due to the high costs of running API-based models on long texts, we mostly limit our evaluation to open-weight LLMs. We report preliminary results using Gemini-1.5 on SummHay in Table 10 in the Appendix. We noticed trends similar to those of open-weight LLMs.

3.3 Analysis: Full-context & Transformer

In the full-context setting, we see mixed results across transformers, with none performing the best across all datasets. Interestingly, Llama-3.1-8B outperforms 70B on SummHay. This surprising result aligns with their relative performance on the RULER benchmark at 128k context length. The 70B model fares better in the oracle setting and shows similar performance on non-retrieval-style datasets. We believe that the 70B model needs additional post-training to improve its long-context retrieval performance.

Command-R underperforms the much smaller Llama-3.1-8B. This could be attributed to its use of RoPE (Su et al., 2021). Command-R increases the base frequency while Llama-3.1 additionally scales RoPE dimensions non-uniformly, likely leading to better long-context capabilities (Ding et al., 2024). However, without specific details on the mid- and post-training with long texts, it would be difficult to identify the exact cause. We direct the reader to Peng et al. (2023) and Lu et al. (2024) for a discussion on long-context methods.

3.4 Analysis: Full-context vs. Compression

With the exception of retrieval on query-focused SummHay dataset, compression-based methods generally underperform full-context (Table 2). To analyze this, we use A3CU *recall* to track the retention of salient information in intermediate outputs. These intermediate outputs correspond to the re-

trieved documents (retrieval) and intermediate summaries (hierarchical, incremental). Figure 1 reports the recall scores for the final summary and the best intermediate output (excl. final). For comparison, we also report the recall score for the full-context summary. Across datasets, the best intermediate recall is significantly higher than the final summary recall, even outperforming full-context.[7]

We highlight two key observations. First, iterative methods suffer catastrophic information loss in their multistage pipeline. Second, the best intermediate recall scores from compression methods show areas of improvement for full-context systems. As a control setting, we evaluated on SummHay-oracle and found full-context to be comparable to the best intermediate recall from compression methods (Figure 2 in the Appendix).

Retrieval: Relative performance of full-context and retrieval varies widely across examples and transformers. Karpinska et al. (2024) observed similar behavior for claim verification on books. In particular, for Llama-3.1-8B on SummHay, we find the final summary to be better than the best intermediate output (Figure 1). This is the optimal scenario, illustrating the system's effectiveness in aggregating information from the retrieved documents. We do not see this behavior in other settings.

Iterative: We qualitatively analyze the outputs from iterative methods. The hierarchical method tends to generate increasingly abstract summaries at higher levels. It often skips details such as entities and numerals in the summaries. We observe this behavior across all transformers. With the incremental method, we attribute poor performance

---

[7]Since recall is impacted by the summary length, we report average length of summaries for each system in Table 9 in the Appendix. We do not find any noticeable correlation.

Transformer	Method	Best	Worst
Llama-3.1-8B	Full-Context	28	10
Llama-3.1-8B	Hierarchical	13	44
Llama-3.1-8B	Incremental	18	21
Llama-3.1-8B	Retrieval	45	4

Table 3: Best-worst ratings from human evaluation on a random sample of 62 examples from SummHay. We report the counts for number of times a system was rated the best or worst amongst the four summaries. We compare each system summary against the reference.

to the large number of intermediate steps (# documents). Even though the system retrieves salient information at an intermediate stage, the model often gets distracted by non-salient information seen in documents thereafter. We provide examples in Table 15 and Table 16 in the Appendix.

In the Appendix (§A.5), we also experiment with short-context transformers such as Llama-3 (Table 11), varying chunk sizes for the hierarchical method, an alternative embedding method for retrieval (Table 13), and grounded generation templates for Jamba and Command-R.

3.5 Human Evaluation

To complement our automatic evaluation, we perform a reference-based human evaluation. We randomly sample 62 examples from the SummHay dataset (≈67%) and ask a human expert[8] to rate the system summaries. We follow recommendations from prior work (Kiritchenko and Mohammad, 2017; Goyal et al., 2022; Pratapa et al., 2023) to use the best-worst rating scale. For each example, the human evaluator picks the best and worst summaries (multiple allowed) among the four methods, full context, hierarchical, incremental, and retrieval (Llama-3.1-8B). They use reference summaries to perform content selection evaluation. We shuffle the presentation order of the system summaries in each example, and system labels are completely hidden from the human evaluator. The results of our human evaluation are presented in Table 3. Retrieval-based summaries are rated the best, followed by full-context, incremental, and hierarchical. These results strongly correlate with our automatic evaluation (Table 2).

---

[8]This task was done by the first author.

3.6 Recommendations for Future Work

Based on our analysis, we make two recommendations for future work on large-scale MDS. First, hybrid systems that combine input compression methods with long-context LLMs. Second, a reference-free content selection evaluation that facilitates further scaling of MDS.

Hybrid Methods: Our analysis using A3CU recall shows the scope for improvement of full-context systems (Figure 1). Recent studies have shown that long-context models are not as effective as claimed for retrieval tasks (Hsieh et al., 2024; Karpinska et al., 2024), and our results support this for large-scale MDS. Iterative methods were previously used for book summarization (Chang et al., 2024) and small-scale MDS (Ravaut et al., 2024). In large-scale MDS, they show a significant loss of salient information. Based on these observations, we advocate for a hybrid approach that utilizes selective input compression methods (Sarthi et al., 2024; Xu et al., 2024; Jiang et al., 2024) in conjunction with a long-context LLM. A hybrid approach could provide optimal performance while improving the runtime over full-context. It also allows for scaling to a very large-scale MDS that goes far beyond the model context window.

Reference-free evaluation: In our analysis, we used a reference-based A3CU metric. As we scale the MDS task to include hundreds or thousands of documents, obtaining high-quality human-written reference summaries will be infeasible. Therefore, reference-free content selection evaluation metrics are needed. Synthetic tasks such as SummHay present a promising alternative.

4 Conclusion

In this work, we contrast the full-context method against three compression-based methods for large-scale MDS. We evaluated on three datasets, SummHay, Background, and WCEP using the A3CU content selection evaluation metric. We find that the full-context and retrieval-based methods perform the best. Iterative methods suffer from significant information loss. Our analysis shows that full-context methods provide suboptimal performance, and we recommend future work to explore hybrid methods that combine the strengths of input compression methods with advances in long-context LLMs.

Limitations

In this work, we rely on high-quality reference summaries to measure the content selection aspects of system-generated summaries. We acknowledge that human evaluation is the gold standard for text summarization. However, for large-scale multi-document summarization ($\approx$100 docs per example), it is prohibitively expensive to perform human evaluation. Karpinska et al. (2024) reported that a human takes about 8-10 hours to read an average book (of similar length to our setting). We leave the extension of human evaluation of full-context and compression-based systems to future work. We also limit our evaluation to models with publicly available weights. We report preliminary results on SummHay using Gemini-1.5 (Table 10 in Appendix). Due to the high API costs of running Gemini on long inputs, we couldn't run them for other datasets. We did not conduct an extensive search for optimal prompts for the summarization task. So, it is possible that the performance of some system configurations could be improved with additional prompt tuning.

Ethics Statement

Hallucination is an important concern for text summarization systems and has been widely studied in the literature. We focus on the content selection aspects of text summarization and choose our evaluation metrics accordingly. However, we recognize the importance of faithfulness evaluation in providing a holistic evaluation of summarization systems. We leave this extension to future work.

Acknowledgments

References

James Allan, Jaime G. Carbonell, George R. Doddington, Jonathan Yamron, and Yiming Yang. 1998. Topic detection and tracking pilot study final report.

Javed Aslam, Fernando Diaz, Matthew Ekstrand-Abueg, Richard McCreadie, Virgil Pavlu, and Tetsuya Sakai. 2015. TREC 2015 Temporal Summarization Track Overview. In *TREC*.

Giang Binh Tran, Mohammad Alrifai, and Dat Quoc Nguyen. 2013. Predicting relevant news events for timeline summaries. In *Proceedings of the 22nd International Conference on World Wide Web*, WWW '13 Companion, page 91–92, New York, NY, USA. Association for Computing Machinery.

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*.

Hai Leong Chieu and Yoong Keok Lee. 2004. Query Based Event Extraction along a Timeline. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '04, page 425–432, New York, NY, USA. Association for Computing Machinery.

Cohere For AI. 2024. c4ai-command-r-08-2024.

Hoa Dang and Karolina Owczarzak. 2009. Overview of the TAC 2008 Update Summarization Task.

Yiran Ding, Li Lyna Zhang, Chengruidong Zhang, Yuanyuan Xu, Ning Shang, Jiahang Xu, Fan Yang, and Mao Yang. 2024. LongroPE: Extending LLM context window beyond 2 million tokens. In *Forty-first International Conference on Machine Learning*.

Demian Gholipour Ghalandari, Chris Hokamp, Nghia The Pham, John Glover, and Georgiana Ifrim. 2020. A large-scale multi-document summarization dataset from the Wikipedia current events portal. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1302–1308, Online. Association for Computational Linguistics.

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

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, and Boris Ginsburg. 2024. RULER: What's the real context size of your long-context language models? In *First Conference on Language Modeling*.

Kung-Hsiang Huang, Philippe Laban, Alexander 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, Mexico City, Mexico. Association for Computational Linguistics.

Huiqiang Jiang, Qianhui Wu, Xufang Luo, Dongsheng Li, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. 2024. LongLLMLingua: Accelerating and enhancing LLMs in long context scenarios via prompt compression. In *Proceedings of the 62nd Annual Meeting*

of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1658–1677, Bangkok, Thailand. Association for Computational Linguistics.

Greg Kamradt. 2023. Needle in a haystack - pressure testing llms.

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. 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 First Conference on Language Modeling.

Svetlana Kiritchenko and Saif Mohammad. 2017. Best-worst scaling more reliable than rating scales: A case study on sentiment intensity annotation. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 465–470, Vancouver, Canada. Association for Computational Linguistics.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles.

Philippe Laban, Alexander R. Fabbri, Caiming Xiong, and Chien-Sheng Wu. 2024. Summary of a haystack: A challenge to long-context llms and rag systems. Preprint, arXiv:2407.01370.

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

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

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

Yi Lu, Jing Nathan Yan, Songlin Yang, Justin T. Chiu, Siyu Ren, Fei Yuan, Wenting Zhao, Zhiyong Wu,

and Alexander M. Rush. 2024. A controlled study on long context extension and generalization in llms. Preprint, arXiv:2409.12181.

Rui Meng*, Ye Liu*, Shafiq Rayhan Joty, Caiming Xiong, Yingbo Zhou, and Semih Yavuz. 2024. Sfr-embedding-2: Advanced text embedding with multi-stage training.

Meta. 2024. Llama 3.1 model card.

Ani Nenkova and Rebecca Passonneau. 2004. Evaluating content selection in summarization: The pyramid method. In Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004, pages 145–152, Boston, Massachusetts, USA. Association for Computational Linguistics.

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

Adithya Pratapa, Kevin Small, and Markus Dreyer. 2023. Background summarization of event timelines. In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, pages 8111–8136, Singapore. Association for Computational Linguistics.

Mathieu Ravaut, Aixin Sun, Nancy Chen, and Shafiq Joty. 2024. On context utilization in summarization with large language models. In Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 2764–2781, Bangkok, Thailand. Association for Computational Linguistics.

Parth Sarthi, Salman Abdullah, Aditi Tuli, Shubh Khanna, Anna Goldie, and Christopher D Manning. 2024. RAPTOR: Recursive abstractive processing for tree-organized retrieval. In The Twelfth International Conference on Learning Representations.

Ori Shapira, David Gabay, Yang Gao, Hadar Ronen, Ramakanth Pasunuru, Mohit Bansal, Yael Amsterdamer, and Ido Dagan. 2019. Crowdsourcing lightweight pyramids for manual summary evaluation. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 682–687, Minneapolis, Minnesota. Association for Computational Linguistics.

Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. 2021. Roformer: Enhanced transformer with rotary position embedding. Preprint, arXiv:2104.09864.

Jamba Team, Barak Lenz, Alan Arazi, Amir Bergman, Avshalom Manevich, Barak Peleg, Ben Aviram, Chen Almagor, Clara Fridman, Dan Padnos, Daniel Gissin, Daniel Jannai, Dor Muhlgay, Dor Zimberg, Edden M

Gerber, Elad Dolev, Eran Krakovsky, Erez Safahi, Erez Schwartz, Gal Cohen, Gal Shachaf, Haim Rozenblum, Hofit Bata, Ido Blass, Inbal Magar, Itay Dalmedigos, Jhonathan Osin, Julie Fadlon, Maria Rozman, Matan Danos, Michael Gokhman, Mor Zusman, Naama Gidron, Nir Ratner, Noam Gat, Noam Rozen, Oded Fried, Ohad Leshno, Omer Antverg, Omri Abend, Opher Lieber, Or Dagan, Orit Cohavi, Raz Alon, Ro'i Belson, Roi Cohen, Rom Gilad, Roman Glozman, Shahar Lev, Shaked Meirom, Tal Delbari, Tal Ness, Tomer Asida, Tom Ben Gal, Tom Braude, Uriya Pumerantz, Yehoshua Cohen, Yonatan Belinkov, Yuval Globerson, Yuval Peleg Levy, and Yoav Shoham. 2024. Jamba-1.5: Hybrid transformer-mamba models at scale. *Preprint*, arXiv:2408.12570.

Giang Tran, Mohammad Alrifai, and Eelco Herder. 2015. Timeline summarization from relevant headlines. In *Advances in Information Retrieval*, pages 245–256, Cham. Springer International Publishing.

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

Lu Wang, Claire Cardie, and Galen Marchetti. 2015. Socially-informed timeline generation for complex events. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1055–1065, Denver, Colorado. Association for Computational Linguistics.

Fangyuan Xu, Weijia Shi, and Eunsol Choi. 2024. RECOMP: Improving retrieval-augmented LMs with context compression and selective augmentation. In *The Twelfth International Conference on Learning Representations*.

Tan Yu, Anbang Xu, and Rama Akkiraju. 2024. In defense of rag in the era of long-context language models. *Preprint*, arXiv:2409.01666.

Dawei Zhu, Liang Wang, Nan Yang, Yifan Song, Wenhao Wu, Furu Wei, and Sujian Li. 2024. Longembed: Extending embedding models for long context retrieval. *Preprint*, arXiv:2404.12096.

A Appendix

We use GitHub copilot and Claude-3.5 Sonnet for assistance with coding and editing.

A.1 Datasets

For background summarization, we use the news articles from the original timeline summarization datasets, Timeline17 (Binh Tran et al., 2013), Crisis (Tran et al., 2015) and Social Timeline (Wang et al., 2015). To constrain the input length, we use a maximum of five news articles from any given day. We also experimented with prefiltering the articles using the news update of the given day, but this did not show improvements in summary quality.

A.2 Experimental Setup

Transformers: We use weights from Huggingface for Llama-3.1-8B,[9] Llama-3.1-70B,[10] Command-R,[11] and Jamba-1.5-Mini.[12]

Compute: We run inference using vLLM on four 48G GPUs (Kwon et al., 2023). Given its large size, we load Llama-3.1-70B with fp8 precision. For the smaller Llama-3.1-8B, we use a single 48G GPU. Our setup includes a mix of Nvidia's A6000, L40, and 6000 Ada GPUs.

Iterative methods: For both iterative methods, we set the maximum chunk size to 4096 tokens. For the hierarchical method, we first generate summaries for each input document. Then, we pack consecutive document summaries into the maximum chunk size for the next summarization step. We stop the process when we only have one summary. For the incremental method, we start by generating the summary of the first document. Then, we concatenate this summary with the following document for the next summarization step. We iterate through every document in the input, in the order provided by the dataset. The document order is relevant for Background (event timelines), but might not be as relevant for SummHay and WCEP.

Retrieval: We limit each document to 1024 tokens and the post-retrieval input to 32k tokens.

Summary length: To set the maximum summary words for each dataset, we first tokenize the summaries in the validation split using NLTK. We use the 80th percentile as the maximum summary words for the systems. To account for the differences in tokenizers for Llama-3.1, Command-R, and Jamba-1.5, we set the maximum number of summary *tokens* by multiplying the maximum summary words with model-specific word-to-token ratios. The word-to-token ratios for Llama-3.1, Command-R, and Jamba-1.5-Mini are 1.145, 1.167, and 1.219 respectively. For iterative methods, we use the same maximum summary token limit at

---

[9]https://hf.co/meta-llama/Llama-3.1-8B-Instruct
[10]https://hf.co/meta-llama/Llama-3.1-70B-Instruct
[11]https://hf.co/CohereForAI/c4ai-command-r-08-2024
[12]https://hf.co/ai21labs/AI21-Jamba-1.5-Mini

```
{document}

Question: {question}

Answer the question based on the provided document. Be
concise and directly address only the specific question asked.
Limit your response to a maximum of {num_words} words.
```

Table 4: Prompt for our summarization task. We pass
the input documents concatenated together by a \n char-
acter. The number of words in the summary are deter-
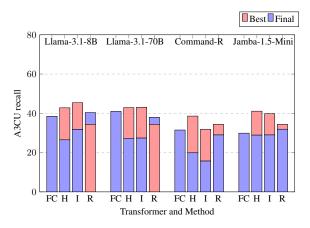mined by the dataset (Table 1).

Figure 2: Salient information retention in the intermedi-
ate and final summaries (A3CU *recall*) for SummHay
(oracle). For each compression method, we report the
best recall from the intermediate outputs and the recall
of the final summary. (H: hierarchical, I: incremental,
R: retrieval, FC: full-context)

each intermediate step. In Table 9, we report the
average length of system-generated summaries.

Prompt: Table 4 provides our prompt for the
text summarization task. We use the same prompt
for all transformers and methods. We follow the
recommendations from model providers and use
the model-specific chat templates from Hugging-
face tokenizers when prompting the instruction-
fine-tuned models.

A.3 Full Metrics

We report the precision, recall, and F1 scores for
A3CU and ROUGE scores (Lin, 2004) for each
dataset: SummHay (Table 5), SummHay oracle
(Table 6), Background (Table 7), and WCEP (Ta-
ble 8). We use Huggingface evaluate for ROUGE
and the original repo for A3CU.[13]

---

[13] https://github.com/Yale-LILY/AutoACU

A.4 Example-level Trends

Figure 3 shows the distribution of A3CU F1 scores
across examples. We notice a significant variance
in system performance across all datasets.

A.5 Ablations

We perform ablation studies to further study our
choice of models and hyperparameters. Given its
small size, we used SummHay for our ablation
experiments.

Gemini-1.5: We run some preliminary ex-
periments with Gemini-1.5 Flash and Pro (Ta-
ble 10). Across methods, we consistently found
that Gemini-1.5 models generate short summaries
and underperform open source models. It is possi-
ble that we could improve their summaries using
a different prompt, but we leave this extension to
future work. Due to the high costs associated with
Gemini API, we did not run experiments with our
larger Background and WCEP datasets.

Llama-3: Our iterative methods do not require
a long-context transformer, so we experiment with
short-context transformers to see if they are better
suited for this task. We run inference with Llama-3
8B and 70B (8k context window) in the SummHay
and SummHay oracle settings (Table 11). We
found that both models are either comparable or
underperform their Llama-3.1 counterparts. It is
likely that the Llama-3.1 models are better at short-
text summarization.

Chunk size: As we have highlighted earlier, the
hierarchical method exhibits a significant degrada-
tion in summary recall. We experiment with larger
chunk sizes that allow for packing more interme-
diate summaries into the transformer. Our results
using 8k, 16k and 32k chunk sizes show minimal
improvements over our default 4k chunk size.

Retriever: Following the setup of SummHay
(Laban et al., 2024), we experiment with the E5-
RoPE embedding for retrieval.[14] We report results
in Table 13. E5-RoPE performs slightly worse than
the SFR-Embedding-2 results from Table 5.

Grounded generation: Jamba provides a
grounded generation option in which the docu-
ments are passed as a separate object in the chat
template. We experiment with this chat template to
see if it provides any gains over our default setting
of concatenating documents in the message. We
report results in Table 14. Interestingly, this tem-
plate helps improve the performance of hierarchical

---

[14] https://huggingface.co/dwzhu/e5rope-base

Figure 3: A3CU F1 score distribution across examples.

and incremental methods and hurts performance in full-context and retrieval settings. This needs further investigation. Command-R also includes a grounded generation template, but it is recommended for documents (or chunks) that contain 100-400 words. We couldn't make it work with full documents from our datasets.

Filtered Background: Our results showed that Background is the most challenging of the three datasets. To simplify the task, we pre-filter the documents using the update summary from the event timeline. We use the E5RoPE model (Zhu et al., 2024) to prefilter up to 128k tokens in the input for each example. However, we did not observe any significant improvements with this filtered dataset.

Transformer	Method	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-Lsum	Recall	A3CU Precision	F1
Llama-3.1-8B	Full-Context	49.4	25.4	28.5	46.4	31.8	39.5	33.9
Llama-3.1-8B	Hierarchical	29.4	10.8	16.4	27.1	14.5	23.3	16.0
Llama-3.1-8B	Incremental	41.5	16.4	22.5	38.0	22.6	27.5	23.2
Llama-3.1-8B	Retrieval	51.8	27.0	29.3	48.9	36.3	36.7	35.3
Llama-3.1-70B	Full-Context	43.7	23.8	25.9	41.3	25.2	46.3	31.1
Llama-3.1-70B	Hierarchical	30.0	11.0	16.4	27.2	15.8	23.6	17.3
Llama-3.1-70B	Incremental	33.1	13.6	19.3	30.5	17.2	27.5	19.7
Llama-3.1-70B	Retrieval	50.2	26.7	29.3	47.1	33.1	43.8	36.3
Command-R	Full-Context	45.0	19.0	24.4	41.2	27.5	38.1	30.4
Command-R	Hierarchical	35.4	8.0	18.4	32.0	10.6	13.9	11.4
Command-R	Incremental	33.0	7.7	17.8	29.7	10.1	15.9	11.4
Command-R	Retrieval	45.0	19.6	24.9	41.8	27.3	38.3	30.4
Jamba-1.5-Mini	Full-Context	44.2	22.0	27.0	41.2	26.6	47.7	32.9
Jamba-1.5-Mini	Hierarchical	38.1	11.6	19.2	35.0	16.5	15.9	15.1
Jamba-1.5-Mini	Incremental	40.7	15.9	21.8	37.1	21.9	27.8	22.5
Jamba-1.5-Mini	Retrieval	46.4	22.8	27.6	42.8	29.4	46.4	34.7

Table 5: Results on SummHay.

Transformer	Method	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-Lsum	Recall	A3CU Precision	F1
Llama-3.1-8B	Full-Context	53.4	29.0	29.7	50.1	38.5	37.9	37.1
Llama-3.1-8B	Hierarchical	40.7	18.2	21.4	38.0	26.5	31.9	27.0
Llama-3.1-8B	Incremental	48.0	21.8	25.2	44.6	31.8	32.9	30.9
Llama-3.1-8B	Retrieval	53.7	28.8	29.8	50.5	40.4	37.2	37.5
Llama-3.1-70B	Full-Context	54.1	30.1	30.7	51.0	41.0	45.8	41.8
Llama-3.1-70B	Hierarchical	37.6	18.3	21.1	34.9	27.3	32.3	27.2
Llama-3.1-70B	Incremental	41.8	20.2	23.5	38.7	27.4	37.8	29.5
Llama-3.1-70B	Retrieval	53.3	28.7	30.1	50.3	38.0	44.0	39.3
Command-R	Full-Context	48.3	20.2	25.4	44.2	31.5	38.0	32.6
Command-R	Hierarchical	41.7	12.5	21.3	38.1	19.9	26.8	21.7
Command-R	Incremental	37.1	11.0	19.8	33.3	15.7	22.6	17.2
Command-R	Retrieval	46.5	19.9	25.1	42.7	29.0	38.6	31.8
Jamba-1.5-Mini	Full-Context	47.6	24.3	28.2	44.4	29.9	47.8	35.1
Jamba-1.5-Mini	Hierarchical	46.7	20.3	25.6	43.5	28.9	33.5	29.6
Jamba-1.5-Mini	Incremental	46.2	20.5	24.4	42.9	29.0	32.5	28.9
Jamba-1.5-Mini	Retrieval	48.5	24.7	28.0	45.2	31.9	46.2	36.3

Table 6: Results on SummHay (oracle).

Transformer	Method	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-Lsum	Recall	A3CU Precision	F1
Llama-3.1-8B	Full-Context	36.5	8.4	18.3	33.2	18.1	15.4	15.6
Llama-3.1-8B	Hierarchical	35.2	7.2	17.5	32.0	15.5	12.8	13.1
Llama-3.1-8B	Incremental	34.4	6.6	16.4	31.1	11.8	10.5	10.0
Llama-3.1-8B	Retrieval	37.7	8.7	19.0	34.2	20.0	16.2	16.9
Llama-3.1-70B	Full-Context	36.6	8.7	18.4	33.4	18.6	15.8	16.1
Llama-3.1-70B	Hierarchical	34.5	7.5	17.4	31.4	17.6	14.2	14.7
Llama-3.1-70B	Incremental	35.2	7.2	16.5	31.9	13.0	11.6	11.1
Llama-3.1-70B	Retrieval	35.7	8.0	18.6	32.2	17.6	16.0	15.7
Command-R	Full-Context	31.9	6.1	17.5	28.6	11.3	11.4	10.3
Command-R	Hierarchical	31.5	5.8	16.7	28.7	10.8	9.5	9.3
Command-R	Incremental	34.6	6.7	16.3	31.3	11.7	9.9	9.7
Command-R	Retrieval	33.2	6.4	17.2	29.9	13.3	12.0	11.8
Jamba-1.5-Mini	Full-Context	33.6	6.8	17.7	30.1	13.1	14.2	12.3
Jamba-1.5-Mini	Hierarchical	33.5	6.0	16.1	30.4	13.4	9.2	10.2
Jamba-1.5-Mini	Incremental	35.5	6.7	16.2	32.1	13.7	9.8	10.4
Jamba-1.5-Mini	Retrieval	33.0	6.1	16.8	29.5	12.5	11.8	11.0

Table 7: Results on Background.

Transformer	Method	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-Lsum	Recall	A3CU Precision	F1
Llama-3.1-8B	Full-Context	37.5	14.2	26.4	29.6	39.1	29.2	30.7
Llama-3.1-8B	Hierarchical	33.9	11.3	23.8	26.1	33.8	25.3	26.2
Llama-3.1-8B	Incremental	32.7	10.5	22.8	25.6	31.7	22.9	24.0
Llama-3.1-8B	Retrieval	36.8	13.7	26.1	29.0	37.9	28.4	29.7
Llama-3.1-70B	Full-Context	37.5	14.1	26.7	30.0	38.6	30.7	31.1
Llama-3.1-70B	Hierarchical	34.3	11.4	23.8	26.6	35.6	25.7	27.1
Llama-3.1-70B	Incremental	32.5	10.4	22.6	25.5	33.0	22.7	24.2
Llama-3.1-70B	Retrieval	37.5	14.2	26.6	30.0	38.3	29.8	30.5
Command-R	Full-Context	36.6	13.7	26.1	29.9	34.1	30.2	28.9
Command-R	Hierarchical	34.1	11.1	23.9	26.4	28.6	28.4	25.6
Command-R	Incremental	34.3	11.7	24.2	27.4	29.2	27.0	25.1
Command-R	Retrieval	36.7	13.7	26.0	29.7	33.0	29.8	28.5
Jamba-1.5-Mini	Full-Context	36.8	13.8	25.8	29.8	36.3	28.6	29.3
Jamba-1.5-Mini	Hierarchical	35.8	12.8	25.1	28.8	36.6	27.9	28.7
Jamba-1.5-Mini	Incremental	34.3	11.7	23.6	27.7	33.4	24.2	25.4
Jamba-1.5-Mini	Retrieval	36.7	13.7	25.6	29.4	36.6	28.3	29.1

Table 8: Results on WCEP.

	Full Context	Retrieval	Hierarchical		Incremental	
			Best	Final	Best	Final
SummHay (Reference: 185)						
Llama-3.1-8B	162	195	172	106	171	141
Llama-3.1-70B	106	148	161	113	150	93
Command-R	135	134	165	151	161	115
Jamba-1.5-Mini	110	120	163	211	177	145
Background (Reference: 174)						
Llama-3.1-8B	228	232	214	222	212	206
Llama-3.1-70B	232	219	208	210	210	205
Command-R	190	215	226	227	236	232
Jamba-1.5-Mini	162	183	213	237	230	233
WCEP (Reference: 35)						
Llama-3.1-8B	44	44	43	41	43	43
Llama-3.1-70B	42	42	43	42	44	43
Command-R	42	41	42	39	42	41
Jamba-1.5-Mini	45	45	45	44	45	44

Table 9: Summary length statistics, using NLTK word tokenizer.

Transformer	Method	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-Lsum	Recall	A3CU Precision	F1
Gemini-1.5-Flash	Full-Context	32.3	15.1	19.7	29.8	19.2	40.6	24.6
Gemini-1.5-Flash	Hierarchical	12.5	4.5	7.2	11.2	8.0	17.2	10.2
Gemini-1.5-Flash	Incremental	37.2	15.5	21.7	34.2	19.6	34.8	23.8
Gemini-1.5-Flash	Retrieval	37.5	18.7	23.3	34.8	22.4	47.4	28.3
Gemini-1.5-Pro	Full-Context	41.8	18.3	23.9	38.8	26.2	36.8	29.2
Gemini-1.5-Pro	Hierarchical	10.9	3.1	6.5	9.7	6.9	17.0	9.2
Gemini-1.5-Pro	Incremental	22.7	6.4	13.4	20.4	10.3	21.8	12.9
Gemini-1.5-Pro	Retrieval	42.5	19.8	24.0	39.3	27.4	41.0	31.6

Table 10: Results on SummHay using Gemini 1.5 Flash and Pro.

Transformer	Method	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-Lsum	Recall	A3CU Precision	F1
SummHay								
Llama-3-8B	Hierarchical	22.0	8.3	13.0	20.3	10.8	23.2	13.6
Llama-3-8B	Incremental	32.6	15.0	20.0	30.0	18.3	36.2	23.2
Llama-3-70B	Hierarchical	17.6	5.0	11.0	16.0	7.4	14.3	9.2
Llama-3-70B	Incremental	34.6	13.8	19.8	31.5	16.7	30.5	20.3
SummHay (oracle)								
Llama-3-8B	Hierarchical	34.0	16.3	19.4	31.4	21.0	35.5	24.6
Llama-3-8B	Incremental	39.2	19.7	23.5	36.3	25.2	45.5	29.9
Llama-3-70B	Hierarchical	30.0	13.3	17.0	27.8	17.0	29.0	19.9
Llama-3-70B	Incremental	39.9	19.0	23.5	36.7	24.1	42.7	29.3

Table 11: Results on SummHay using the short context Llama-3 models.

Transformer	Method	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-Lsum	Recall	A3CU Precision	F1
Llama-3.1-8B	Hierarchical-8K	27.3	10.1	15.3	25.1	14.0	22.9	15.6
Llama-3.1-8B	Hierarchical-16K	30.8	12.6	17.6	28.4	16.7	27.9	18.9
Llama-3.1-8B	Hierarchical-32K	28.9	11.4	16.4	26.8	15.8	26.0	17.5
Jamba-1.5-Mini	Hierarchical-8K	38.2	11.8	19.5	35.2	14.5	18.4	15.2
Jamba-1.5-Mini	Hierarchical-16K	37.7	12.0	20.4	34.5	14.7	19.9	16.0
Jamba-1.5-Mini	Hierarchical-32K	37.0	12.3	19.7	33.6	14.8	21.6	16.3

Table 12: Results on SummHay using different chunk sizes for the hierarchical method.

Transformer	Method	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-Lsum	Recall	A3CU Precision	F1
Llama-3.1-8B	Retrieval-E5	50.1	25.1	28.6	47.3	33.9	35.1	33.2
Llama-3.1-70B	Retrieval-E5	49.8	25.7	28.7	46.8	32.2	41.1	34.6
Command-R	Retrieval-E5	44.8	19.3	24.5	41.5	27.2	36.7	29.5
Jamba-1.5-Mini	Retrieval-E5	44.1	20.8	25.5	40.7	26.9	42.0	31.5

Table 13: Results on SummHay using the E5 RoPE retriever instead of SFR-Embedding-2.

Transformer	Method	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-Lsum	Recall	A3CU Precision	F1
Jamba-1.5-Mini-Grounded	Full-Context	45.4	22.0	26.7	42.6	26.0	43.5	31.3
Jamba-1.5-Mini-Grounded	Hierarchical	40.8	14.3	20.9	37.7	18.6	20.2	18.2
Jamba-1.5-Mini-Grounded	Incremental	41.8	17.6	22.7	38.5	23.9	32.6	25.9
Jamba-1.5-Mini-Grounded	Retrieval	44.4	21.8	26.3	41.2	26.5	43.2	31.1

Table 14: Results on SummHay using grounded generation chat template with Jamba-1.5-Mini.

Gold: China announces its intent to increase tariffs from 10% to 25% on $60 billion worth of imports from the United States effective June 1, 2019.
Prediction (best): US markets plummeted after China announced retaliatory tariffs on $60 billion of US imports, including batteries, spinach, and coffee, in response to the Trump administration's hike on Chinese imports from 10% to 25%.
Prediction (final): The US and China engage in an escalating trade war, imposing tariffs on each other's goods, causing market losses, economic uncertainty, and affecting global economic growth, trade, and markets, with potential impacts on US consumers and businesses.

Gold: Venezuela's Ambassador to Italy, Isaías Rodríguez, resigns, citing the government's lack of finances amid international sanctions on the Nicolás Maduro government. Rodríguez says, due to the sanctions, he has been unable to pay his staff or the rent for his office in Rome.
Prediction (best): The Venezuelan ambassador to Italy, Isaías Rodríguez, has resigned due to the country's financial difficulties caused by US sanctions. He cites inability to pay employees and rent, and will dedicate himself to being a grandfather, stating he "cannot turn water into wine"
Prediction (final): The document discusses Venezuela's humanitarian crisis, disputed re-election, and diplomatic efforts to resolve the crisis, including US sanctions, opposition negotiations, and international support for a democratic transition and potential regime change.

Gold: One colleague mentions that their department uses specific productivity metrics like the number of tasks completed per week to measure the impact of remote work on employee productivity. A colleague argues that remote work results in fewer distractions compared to an open office environment, citing a 2019 study that found remote workers are 35% less likely to be distracted by co-workers. Another colleague points out that remote employees often work more overtime, referencing a 2022 company internal report showing a 20% increase in overtime hours since the shift to remote work. A colleague highlights that remote work allows for extended deep work periods, mentioning that they've personally experienced 3-hour uninterrupted work blocks at home, which were rarely possible in the office. One colleague mentions that remote work can align better with individual employees' peak productivity times, citing a 2020 survey where 60% of employees reported higher productivity outside the traditional 9-to-5 work hours.
Prediction (best): Yes, productivity level change is discussed in the conversation. \n\n Colleague 3 mentions a 2019 study that found remote workers are 35% less likely to be distracted by co-workers compared to those in an open office, suggesting that remote work can be a productivity booster. Colleague 1 also agrees that the solitude of remote work can lead to deep focus and increased productivity. \n\n Additionally, colleague 2 compares the office environment to a marching band parading through the office, emphasizing how the constant interruptions can break one's flow and hinder productivity. This highlights the potential benefits of remote work in terms of increased productivity due to fewer distractions. \n\n Overall, the conversation suggests that remote work can have a positive impact on productivity, with fewer distractions and interruptions allowing employees to focus more effectively.
Prediction (final): Yes, productivity level change is discussed.

Gold: Suspected Baloch nationalist gunmen storm a coal mine in Islamabad, Pakistan, killing at least five people.
Prediction (best): At least five people were killed in a gun and bomb attack on a coal mine in southwestern Pakistan by ethnic Baloch separatists, with two miners, two security personnel, and a driver among the dead, amid escalating violence in the region.
Prediction (final): The document reports various news articles from 2019, including attacks on hotels in Gwadar, Pakistan, by the Balochistan Liberation Army, as well as human trafficking and security concerns during the Muslim fasting month of Ramadan.

Table 15: Examples illustrating loss of salient information in Hierarchical method.

Gold: One student shares that they do 10 minutes of deep breathing exercises each night before going to bed to help reduce stress. A student mentions that taking a 30-minute walk each afternoon helps clear their mind and manage stress levels during exam preparation. One of the students suggests taking a 5-minute break after every 25 minutes of studying, and mentions the Pomodoro technique as a helpful method. A student recommends using a specific meditation app called 'Calm' that they use for 15 minutes each morning to manage stress. Another student shares their habit of writing down 3 positive affirmations about their capabilities each day, which they find significantly helps reduce their stress.

Prediction (best): Yes, the students indirectly shared exam stress tips through their conversation. They discussed various techniques and strategies that helped them manage stress and stay focused during exam preparation, including:\n\n* Maintaining a consistent sleep schedule and taking breaks\n* Using the Pomodoro Technique to avoid burnout\n* Incorporating digital detox into their daily routine\n* Practicing deep breathing exercises\n* Taking short walks to clear their minds\n* Trying relaxation techniques such as meditation\n* Using spaced repetition with the Anki app\n* Creating mind maps for visual learners\n* Using a meditation app like 'Calm'\n\n They also shared specific experiences and recommendations, such as Alejandro's 30-minute afternoon walk, which helped him manage stress and clear his mind. They also discussed the potential benefits of physical activity, meditation, and relaxation techniques in reducing anxiety and improving focus. The conversation was supportive and encouraging, with the students offering to help and motivate each other throughout the exam preparation period.

Prediction (final): No, they did not directly share exam stress tips. Their conversations centered on sharing personal experiences and strategies related to studying, time management, prioritization, and productivity, which indirectly contributed to managing exam stress. They discussed various approaches to studying, adapting their study plan, and organizing information, but did not explicitly mention stress management techniques like deep breathing exercises or the Pomodoro technique. Their focus was on developing a study plan, sharing notes, and reviewing material together to prepare for the exam, rather than directly addressing stress management.

Gold: Prince Harry and Meghan, the Duke and Duchess of Sussex, name their son Archie Harrison Mountbatten-Windsor.

Prediction (best): Prince Harry and Meghan, Duchess of Sussex, have named their baby boy Archie Harrison Mountbatten-Windsor. The palace announced the name after the couple posed with the two-day-old for the media, meeting Queen Elizabeth II for the first time.

Prediction (final): Prince Harry and Meghan Markle introduced their son Archie to the world at Windsor Castle, presenting a more casual and intimate image, contrasting with traditional royal protocol. They shared parenting duties and responsibilities, showcasing a new generation of feminist royals.

Gold: The United States lifts tariffs on steel and aluminum from Canada and Mexico.

Prediction (best): The US will lift tariffs on steel and aluminum from Canada and Mexico, resolving a yearlong deadlock and providing economic relief to Detroit automakers, who paid billions in increased duties, and addressing a key obstacle to ratifying the new North American trade pact.

Prediction (final): The number of asylum seekers in Canada tripled between 2015 and 2018, with a peak of 8,801 claims in August 2017. Most claims came from Haitians, Nigerians, and US citizens, with many fleeing the US due

Table 16: Examples illustrating loss of salient information in Incremental method.