Social Media Summarization at Scale: How Far Small and Open LLMs Match-Up Against Commercial LLMs?

Anonymous ACL submission

Abstract

Summarizing Social media content is complex due to its unique language specifics, and the scale and speed of content generation. Recent developments in Large Language Models (LLMs) enable text summarization with unprecedented accuracy; however, their high computational cost and input length constraints remain major obstacles for real-world applications at scale. Several approaches based on Small Language Models (SLMs) and Open source alternatives exist that present affordable alternatives for managing computational complexity in practice. In this paper, we explore popular SLMs and open LLMs for long-context summarization applied to Social media, and evaluate their performance using two datasets comprising of long social media discussions collected online. The key findings show that fine-tuning smaller models and optimizing input selection can achieve high-quality summarization at significantly lower computational costs.

1 Introduction

Summarizing Social media content is complex due to the informal language, noise, and the lack of coherence across multiple posts; it requires technical solutions that can handle both the scale and speed of content generation (Thakur, 2023). The recent breakthroughs in transformer architectures and subsequent rise of LLMs have significantly boosted the accuracy of text summarization over the prior State of the Art (Pu et al., 2023). However, the exceptional results come at the cost of higher computational complexity and therefore practical deployment is limited to specific scenarios and incurs significant costs (Kaddour et al., 2023). Also, most of the published LLM's limit their input size to allow execution on modern hardware with its current memory limitations. The models with highest allowable input sizes require cutting edge hardware clustered in big and expensive servers; making them available often only to selected users.

In the following article, we focus on the task of Social media discussion summarization, and specifically edge cases of very lengthy discussions where summarization is most needed. In this setting, our objective is to evaluate how close, in terms of accuracy, smaller models can get to the top performing and largest LLMs. We aim to achieve summaries such as those produced by models with hundreds of billions of parameters but at far lesser compute expense. We set simple criteria to allow only models that are executable on relatively affordable hardware setup: two A100 GPUs (40GB VRAM each). This results in capability to run inference on most modern 7B models with their maximum possible context size. We also compare those models to smaller but more efficient pre-trained transformer architectures (Tay et al., 2022) designed for long-text input by lowered computational complexity (e.g. Longformer (Beltagy et al., 2020)). Additionally, we evaluate the most frequently used strategies for handling long-text with large models such as text trimming and text chunking.

To facilitate our experiments we construct two new datasets comprising of long social media discussions compiled from public opinions on COVID-19 in Singapore during the pandemic, and corresponding reference summaries generated by a state of the art LLM as ground truths. In this setting, our results show that quality of commercial LLM summarization can be achieved by smaller and therefore significantly more cost efficient models, albeit requiring fine-tuning with additional labeled data.

Related Work: Text summarization has a long standing history (Yadav et al., 2022; Widyassari et al., 2022), with a number of techniques and approaches proposed under social media summarization (Papagiannopoulou and Angeli, 2023), and long-context summarization (Koh et al., 2022). In

comparison, ours is a benchmarking study, and we build upon the aforementioned studies as a foundation to evaluate popular summarization approaches and assess their performance within the context of our specific application setting (Tay et al., 2022). This is similar in vein to (Zhang et al., 2024; Al Nazi et al., 2025; Yu et al., 2023); however, ours is novel in that we focus exclusively on long-context social media summarization.

2 Approach and Methodology

For our purposes, Social Media summarization is the task of generating a concise and informative summary, given a collection of social media content (including posts, comments, messages, etc.) on a topic of interest or from a discussion thread.

The primary objective of our research is to assess the extent to which the summarization performance of the largest LLMs can be achieved using significantly smaller and more cost-efficient models. For this assessment, we use reference summaries generated by GPT-4o with a 128k context window, which, at the time of writing, is widely regarded as one of the top-performing models (Liu et al., 2024, 2023).

2.1 Summarization Approaches Considered

We explore two main categories of models: 1) pretrained on large datasets not necessarily related to ours; and 2) fine-tuned on a dataset collected by us and using the same theme as test dataset. In both of those categories: we compare several long-text summarization pipelines based on Gemma-7B: a 7 billion parameter LLM with 8k token context size available openly for commercial use.

More specifically, we consider text chunking approaches that split input into multiple portions, each summarized separately; then resulting summaries are concatenated together for another round of summarization (and repeated chunking if still exceeding model input size). We compare three most popular text chunking strategies: 1) *sequential chunking* (Jaiswal et al., 2021) that splits input text into even portions equal to maximal model context size (8k tokens in our case); 2) *sequential chunking with overlap* (Ivgi et al., 2023) that enhances previous approach by overlapping subsequent chunks in order to keep some contextual relationship; 3) *clustering* (Bhaskar et al., 2023) that treats every comment separately and groups them together for summarization based on textual similarity (we use K-means applied to comment embeddings generated by *mpnet-v2*; and with cluster count in between 10 to 500, best one selected using silhouette score). For reference, we compere those chunking strategies with simpler solution that takes text trimmed to max model input and discards the reminder.

Additionally, we also evaluate two (parameter-wise) smaller state of the art models dedicated to long-text summarization: Longformer (Beltagy et al., 2020) and Primera (Xiao et al., 2021). For fine-tuning experiments, we train those two models with a single epoch using earlier mentioned train datasets (setting aside 10% for validation). We compare those models to LLM fine-tuned using LoRA (Hu et al., 2022) on all linear layers over the course of 15 epochs.

2.2 Datasets

We constructed two datasets to evaluate summarization performance for long-text social media discussions, using community comments as model input and reference summaries generated by OpenAIs GPT-4o as labels. The datasets were sourced from two platforms Facebook and Reddit focusing on COVID-19 discussions. Facebook data was collected from *Channel News Asia* posts, while Reddit data was extracted from the */r/singapore* forum, covering discussions between July 2020 and December 2022. The Facebook dataset includes 23,547 posts with 2,468,315 comments, and the Reddit dataset contains 4,143 posts with 276,389 comments. We filtered these to create test sets: *fb1k* (top 1,000 longest Facebook discussions, 16k tokens) and *r*d300 (top 300 longest Reddit discussions, 10k tokens). Additionally, we curated fine-tuning datasets: *fb500* (500 subsequent Facebook discussions by length) and *rd150* (150 Reddit discussions using the same ordering). Full dataset construction details and statistics are provided in Appendix A.

3 Experiments and Results

We present below our experiments on benchmarking summarization performance using the above datasets. All experiments were conducted with low-temperature settings to ensure consistency across multiple runs and produce comparable results.

3.1 Impact of Model Size and Fine-Tuning

We first study the performance of the models with and without fine-tuning, according to the widely

Architecture	R1	R2	RL	BS	Chrf	Meteor
Pre-trained Models						
Gemma-7b /No Chunking	0.302	0.065	0.164	0.851	56.642	0.209
Gemma-7b /Chunking (Seq)	0.310	0.068	0.167	0.849	54.103	0.202
Gemma-7b /Chunking (Seq+Overlap)	0.313	0.068	0.169	0.849	54.114	0.201
Gemma-7b /Chunking (Cluster)	0.291	0.056	0.156	0.844	55.033	0.205
Longformer (16k)	0.177	0.016	0.091	0.782	47.466	0.165
Fine-Tuned Models						
Gemma-7b /LoRA (1k)	0.344	0.083	0.169	0.859	**63.691**	0.303
Gemma-7b /LoRA/ Chunking (Seq) (1k)	0.339	0.082	0.168	0.861	63.155	**0.323**
Longformer (16k)	**0.361**	**0.103**	**0.188**	**0.873**	43.395	0.182
Primera (4k)	0.345	0.099	0.186	0.855	46.521	0.199

Table 1: Model performance on Facebook (fb1k) dataset. In brackets, next to model name, max size of input accepted by model (all Gemma-7b models have 8k input size). Fine-Tuned models use fb500 dataset with 10% validation set. *R1=Rouge-1;R2=Rouge-2;RL=Rouge-L;BS=BertScore.

used Rouge metrics (Lin, 2004). For fair comparison, we choose one candidate to represent Open LLMs (Gemma-7b), and Small LMs (Longformer). We study other open LLMs in a separate experiment.

Table 1 shows comparison of all described models and pipelines, each in their best performing configuration (full results, with all context sizes per each pipeline/model can be seen in Appendix B).

Match-up under Zero-shot Setting: We observe that, without any fine-tuning, Open LLMs are the best and perform roughly similar regardless of with or without chunking. Looking at subtle differences, the best solution is chunking with overlap, just slightly edging over its simpler version without overlap and outperforming non-chunking and cluster chunking versions by larger margin. The untuned small LM solutions clearly perform the worst. Apart of model size, this could be potentially related to the fact that used LLMs have knowledge cut-off post covid; while smaller models were pretrained on unrelated datasets.

Match-up after Fine-tuning: After fine-tuning, the situation changes dramatically in a rather unexpected way: the significantly smaller model (Longformer with 162m parameters) beats all more complex solutions (even newer Primera model with same or smaller input size). Furthermore, it should be noted that Longformer uses only initial 16K tokens of input and still overwhelmingly exceeds in performance over chunking solutions that analyse the complete input. This becomes even more radical with fine-tuned Gemma-7b that also beats any untuned model but only using initial 1k tokens of text as input.

Match-up using Metrics besides Rouge: Same experiments can be analysed with more complex metrics that address deficiencies of Rouge and put more focus on text semantic similarity (Supriyono et al., 2024; Roy et al., 2021): BertScore (Zhang et al., 2020); Chrf (Popović, 2015); and Meteor (Lavie and Agarwal, 2007). Those show fine-tuned LLM performs better than smaller models. This observation and previous are reinforced by similar results coming from experiments on our second dataset (Reddit) as can be seen in Table 4 in Appendix B. For that reason we decided to further explore the relationship between input size and model performance.

3.2 Impact of Input Size

We performed two series of experiments: 1) firstly we looked at relationship between input text length (untrimmed) and model performance using correlation; 2) secondly, we experimented with different model input limitations but constant input text length (e.g. model with 4k context size but analysing full input via chunking).

Does Input text length impact model performance? In the first experiment we discovered there was no relationship between performance and input length, ie. posts with very long discussions were handled on average with equal performance as post with little comments; regardless of used metrics. The detailed results can be seen in Table 5 in Appendix B.

Does Context Size impact model performance? In the second experiment, for Longformer and

Gemma-7B we forcefully constrained model input size from as little as 1k token up to the pre-trained model maximal context size allowed by it's architecture (different values depending on the model; or no constrains in case of chunking). Those limitations were assumed both during training and testing. We noticed that while bigger context gives slightly better performance in most cases, the differences are rather small for Chrf metric (see Fig. 1) and similar but with some outliers in case of simpler Rouge metrics (see Fig. 2 in Appendix B).

Does Chunking Overlap Size matter? Aside of the input size observation, second interesting insight from Table 1, was that among untuned models, the best performing one was using chunking with overlap. To see if this advantage could be pushed even further we checked if revealing larger potential contextual dependencies between chunks to the model (ie. bigger chunk overlap) would improve the overall performance. We did this by testing with chunk overlap value from 5 to 25% (see Table 6 in the appendix). Our experiments show that setting of this parameter does not have a major impact on performance, with minimal overlap being only sightly better than others.

3.3 Impact of Model Architectures:

Finally, in our last experiment we checked if selection of LLM architecture and pre-training regime would have a significant impact on our results. Therefore, in addition to earlier mentioned Gemma-7b, we repeated above experiments with Llama 3.1, and two flavours of Mistral (7b and 12b). The 12b was included in our experiments as it was still possible to execute it on earlier mentioned hardware setup. The final results reveal small differences between 7b models and slight advantage for the 12b model (see Table 2).

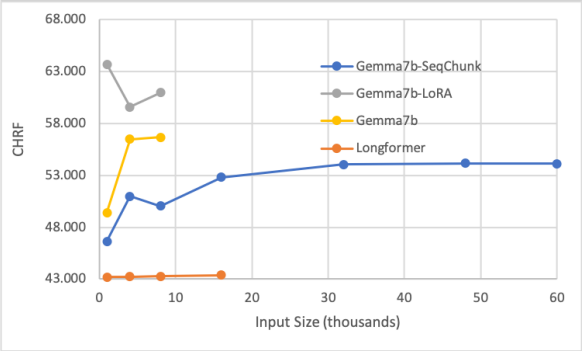

Figure 1: Input size vs. accuracy for CHRF metric

Architecture	R1	R2	RL
Gemma (7b)	0.310	0.068	0.167
Llama 3.1 (7b)	0.288	0.078	0.154
Mistral 0.3 (7b)	0.303	0.080	0.162
Mistral Nemo (12b)	0.329	0.078	0.162

Table 2: Relationship between text overlap and accuracy for chunking architectures (all based on Gemma-7b with Sequential Chunking algorithm).

4 Discussion and Conclusions

This paper studied the extent to which the long-context summarization performance of the largest LLMs can be achieved using significantly smaller and more cost-efficient models, as applied to social media. The key findings are: i) Open LLMs (e.g., Gemma-7b) perform best in zero-shot settings, with minimal impact from chunking strategies, while Small LMs (e.g., Longformer) perform poorly without fine-tuning, likely due to pre-training on unrelated datasets. ii) Fine-tuning significantly boosts performance, with Longformer (162M parameters) outperforming larger, more complex models. Fine-tuned models achieve high accuracy even with limited input size (e.g., Gemma-7b using only 1k tokens surpasses untuned models processing full input). iii) Metrics like BertScore, Chrf, and Meteor confirm that fine-tuned LLMs outperform smaller models, reinforcing earlier observations from the Reddit dataset. iv) No strong correlation found between input text length and summarization quality, indicating long discussions do not necessarily improve performance. Also, larger context windows yield minor gains, but improvements are marginal across different metrics. v) Overlapping chunking yields slight performance benefits in zero-shot settings; however, increasing chunk overlap does not significantly enhance performance beyond minimal overlap. vi) Differences between various 7B models (Gemma, Llama 3.1, Mistral) are minor, with a slight advantage for the Mistral-12B model.

These findings highlight that fine-tuning smaller models and optimizing input selection can achieve high-quality summarization at significantly lower computational costs. The key insight is that the gist of information expressed by the community is often present in the initial parts of the discussion and the rest of the content contains information which is often not considered even by very large models that can digest full input size.

Limitations

Our datasets come from only two social media platforms, this could be expanded to a wider range (e.g. Twitter, Youtube) to cover different types of discussions and communities. Furthermore, exploring a broader range of topics, including use cases beyond COVID-19 policy opinions could help to validate our findings for a broader set of applications. In terms of methodology, we report only on quantitive evaluation using popular summarisation metrics. The study could be supplemented with qualitative evaluation by human annotators to confirm our results; due to lack of resources this activity is outside of the scope of our article.

References

Zabir Al Nazi, Md Rajib Hossain, and Faisal Al Mamun. 2025. Evaluation of open and closed-source llms for low-resource language with zero-shot, few-shot, and chain-of-thought prompting. *Natural Language Processing Journal*, 10:100124.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv:2004.05150*.

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, Toronto, Canada. Association for Computational Linguistics.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Maor Ivgi, Uri Shaham, and Jonathan Berant. 2023. Efficient long-text understanding with short-text models. *Transactions of the Association for Computational Linguistics*, 11:284–299.

Aman Jaiswal, Juan Ramirez-Orta, and Evangelos Milios. 2021. Chunksumm: Extending bert for long document summarization. In *Dalhousie Computer Science In-House (DCSI) Conference (DCSI)*.

Jean Kaddour, Joshua Harris, Maximilian Mozes, Herbie Bradley, Roberta Raileanu, and Robert McHardy. 2023. Challenges and applications of large language models. *ArXiv*, abs/2307.10169.

Huan Yee Koh, Jiaxin Ju, Ming Liu, and Shirui Pan. 2022. An empirical survey on long document summarization: Datasets, models, and metrics. *ACM Comput. Surv.*, 55(8).

Alon Lavie and Abhaya Agarwal. 2007. Meteor: an automatic metric for mt evaluation with high levels of correlation with human judgments. In *Proceedings of the Second Workshop on Statistical Machine Translation*, StatMT '07, page 228231, USA. Association for Computational Linguistics.

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

Mingxin Liu, Tsuyoshi Okuhara, Zhehao Dai, Wenbo Huang, Hiroko Okada, Emi Furukawa, and Takahiro Kiuchi. 2024. Performance of advanced large language models (gpt-4o, gpt-4, gemini 1.5 pro, claude 3 opus) on japanese medical licensing examination: A comparative study. *medRxiv*.

Yiheng Liu, Tianle Han, Siyuan Ma, Jiayue Zhang, Yuanyuan Yang, Jiaming Tian, Hao He, Antong Li, Mengshen He, Zhengliang Liu, Zihao Wu, Lin Zhao, Dajiang Zhu, Xiang Li, Ning Qiang, Dingang Shen, Tianming Liu, and Bao Ge. 2023. Summary of chatgpt-related research and perspective towards the future of large language models. *Meta-Radiology*, 1(2):100017.

Afrodite Papagiannopoulou and C. Angeli. 2023. Social media text summarisation techniques and approaches: A literature review. In *Proceedings of the 2022 5th International Conference on Algorithms, Computing and Artificial Intelligence*, ACAI '22, New York, NY, USA. Association for Computing Machinery.

Maja Popović. 2015. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal. Association for Computational Linguistics.

Xiao Pu, Mingqi Gao, and Xiaojun Wan. 2023. Summarization is (almost) dead. *ArXiv*, abs/2309.09558.

Devjeet Roy, Sarah Fakhoury, and Venera Arnaoudova. 2021. Reassessing automatic evaluation metrics for code summarization tasks. In *Proceedings of the 29th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, pages 1105–1116.

Supriyono, Aji Wibawa, Suyono, and Fachrul Kurniawan. 2024. A survey of text summarization: Techniques, evaluation and challenges. *Natural Language Processing Journal*, 7:100070.

Yi Tay, Mostafa Dehghani, Dara Bahri, and Donald Metzler. 2022. Efficient transformers: A survey. *ACM Comput. Surv.*, 55(6).

Nirmalya Thakur. 2023. Social media mining and analysis: A brief review of recent challenges. *Information*, 14(9).

Adhika Pramita Widyassari, Supriadi Rustad, Guruh Fajar Shidik, Edi Noersasongko, Abdul Syukur, Affandy Affandy, and De Rosal Ignatius Moses Setiadi. 2022. Review of automatic text summarization techniques methods. *Journal of King Saud*

University - Computer and Information Sciences, 34(4):1029–1046.

Wen Xiao, Iz Beltagy, Giuseppe Carenini, and Arman Cohan. 2021. Primera: Pyramid-based masked sentence pre-training for multi-document summarization. In *Annual Meeting of the Association for Computational Linguistics*.

Divakar Yadav, Jalpa Desai, and Arun Kumar Yadav. 2022. Automatic text summarization methods: A comprehensive review. *ArXiv*, abs/1807.09834.

Hao Yu, Zachary Yang, Kellin Pelrine, Jean Francois Godbout, and Reihaneh Rabbany. 2023. Open, closed, or small language models for text classification? *arXiv preprint arXiv:2308.10092*.

Gongbo Zhang, Qiao Jin, Yiliang Zhou, Song Wang, Betina Idnay, Yiming Luo, Elizabeth Park, Jordan G Nestor, Matthew E Spotnitz, Ali Soroush, et al. 2024. Closing the gap between open source and commercial large language models for medical evidence summarization. *npj Digital Medicine*, 7(1):239.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with BERT. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

A Dataset Construction and Stats

This section contains more details about the datasets referred in Section 2.

We constructed two datasets to measure summarisation performance for long-text social media discussions. The datasets contain as model input: community comments related to same post; while as labels: reference summaries prepared by the OpenAI GPT-4o model. The data sources for those datasets are two popular social media platforms: 1) Facebook; and 2) Reddit; while the topic for both is COVID-19. The Facebook dataset was collected by inspecting the account of a Singaporean media outlet - *Channel News Asia* [1] and filtering posts by COVID-19 related keywords. For Reddit, similar operation was done but for a popular discussion forum *"/r/singapore"* [2].

The data was collected from discussion threads that had activity between July 2020 and December 2022 (ie. approximately 2.5 years). For Facebook that amounted to 23,547 posts with 2,468,315 comments, and for Reddit 4,143 posts with 276,389 comments. We further filtered this data to discussions that had at least 16k tokens of concatenated comment content for Facebook (called fb1k dataset - equivalent to top 1000 longest discussions) and 10k for Reddit (called rd300 - equivalent to top 300 longest discussions). Those two datasets are test sets for all our experiments. Additionally, we also created two more datasets that are used in our fine-tuning experiments: for Facebook - fb500 (includes 500 subsequent posts ordered by discussion length); and for Reddit - rd150 that has 150 subsequent posts using same ordering methodology. The dataset statistics are provided in Table 3.

	fb1k	**fb500**	**rd300**	**rd150**
		TOKENS		
Total	26m	7.3m	5.8m	1.3m
Per post				
- MAX	140k	58.7k	65.1k	14.4k
- MIN	14.6k	11.3k	9.6k	7.1k
- AVG	21.1k	14.5k	19.4k	8.7k
- STD	11.9k	3.8k	9.3k	0.9k
		COMMENTS		
Total	669.3k	191.8k	104.2k	22.7k
Per post				
- MAX	5.3k	1.1k	1.5k	0.4k
- MIN	131	146	64	30
- AVG	669	382	347	151
- STD	437	92	271	60

Table 3: Datasets used during experiments and their characteristics.

B Detailed Results

This section contains supplementary tables referred in Section 4.

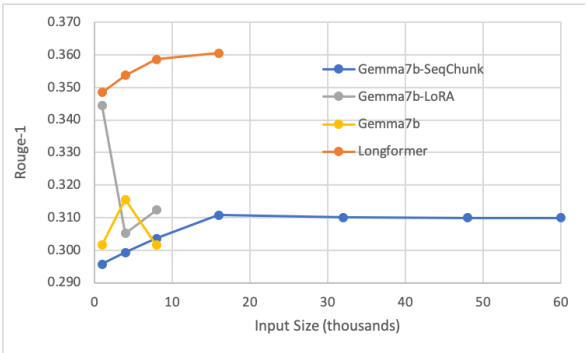

Figure 2: Input size vs. accuracy for Rouge-1 metric

[1] https://www.facebook.com/ChannelNewsAsia
[2] https://www.reddit.com/r/singapore/

Architecture	R1	R2	RL	BS	Chrf	Meteor
Pre-trained Models						
Gemma-7b /No Chunking	0.303	0.059	0.155	0.845	53.803	0.208
Gemma-7b /Chunking (Seq)	0.301	0.060	0.158	0.844	48.721	0.185
Gemma-7b /Chunking (Seq+Overlap)	0.297	0.060	0.156	0.845	47.167	0.182
Gemma-7b /Chunking (Cluster)	0.280	0.048	0.144	0.840	50.236	0.188
Longformer (16k)	0.203	0.021	0.098	0.789	49.228	0.172
Fine-Tuned Models						
Gemma-7b /LoRA (1k)	0.317	0.064	0.152	0.852	**63.153**	**0.296**
Longformer (16k)	0.285	0.054	0.142	0.850	40.178	0.148
Primera (4k)	**0.338**	**0.086**	**0.163**	**0.842**	58.129	0.242

Table 4: Model performance on Reddit (rd300) dataset. In brackets, next to model name, max size of input accepted by model (all Gemma-7b models have 8k input size). Fine-Tuned models use rd150 dataset with 10% validation set. *R1=Rouge-1;R2=Rouge-2;RL=Rouge-L;BS=BertScore.

Architecture	R1	R2	RL	BS	Chrf	Meteor
Pre-trained Models						
Gemma-7b /No Chunking	-0.032	0.011	0.003	-0.037	0.041	0.031
Gemma-7b /Chunking (Seq)	-0.047	-0.032	-0.004	-0.021	0.090	0.034
Gemma-7b /Chunking (Seq+Overlap)	-0.022	-0.016	0.006	0.010	0.073	0.048
Gemma-7b /Chunking (Cluster)	-0.036	-0.040	0.013	-0.041	-0.007	-0.022
Longformer (16k)	-0.094	-0.007	-0.054	-0.100	-0.114	-0.115
Fine-Tuned Models						
Gemma-7b /LoRA (1k)	-0.071	-0.020	0.008	-0.063	-0.070	-0.034
Longformer (16k)	-0.045	-0.032	0.022	-0.059	0.035	-0.006
Primera (4k)	-0.014	0.017	0.059	0.014	0.038	0.020

Table 5: Correlation between sample size (token count) and model accuracy for different model architectures.

Overlap value (%)	R1
5	0.313
10	0.311
15	0.310
20	0.312
25	0.311

Table 6: Relationship between text overlap and accuracy for chunking architectures (all based on Gemma-7b with Sequential Chunking algorithm).

Architecture (input limit)	R1
Gemma-7b /Chunk-Seq (4k)	0.299
Gemma-7b /Chunk-Seq (8k)	0.304
Gemma-7b /Chunk-Seq (16k)	0.311
Gemma-7b /Chunk-Seq (32k)	0.310
Gemma-7b /Chunk-Seq (48k)	0.310
Gemma-7b /Chunk-Seq (non)	0.310
Longformer (1k)	0.348
Longformer (4k)	0.354
Longformer (8k)	0.359
Longformer (16k)	0.361
Gemma-7b /LoRA (1k)	0.344
Gemma-7b /LoRA (4k)	0.305
Gemma-7b /LoRA (8k)	0.312
Gemma-7b /No Chunk (1k)	0.302
Gemma-7b /No Chunk (4k)	0.316
Gemma-7b /No Chunk (8k)	0.302

Table 7: Relationship between input size limit and accuracy for different architectures.