Topic Modeling for Short Texts with Large Language Models

Tomoki Doi [1] **Masaru Isonuma** [1,2] **Hitomi Yanaka** [1]
[1] The University of Tokyo [2] The University of Edinburgh
{doi-tomoki701, hyanaka}@is.s.u-tokyo.ac.jp m.isonuma@ed.ac.uk

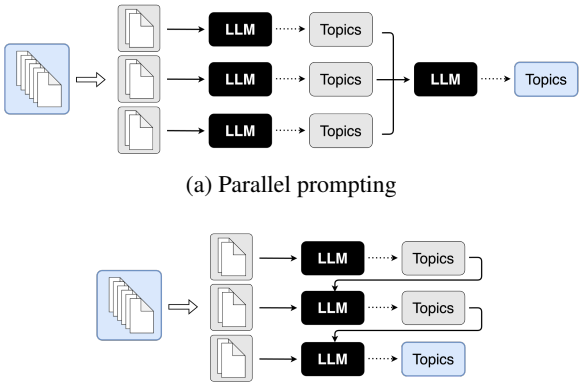
Abstract

As conventional topic models rely on word co-occurrence to infer latent topics, topic modeling for short texts has been a long-standing challenge. Large Language Models (LLMs) can potentially overcome this challenge by contextually learning the meanings of words via pretraining. In this paper, we study two approaches to using LLMs for topic modeling: parallel prompting and sequential prompting. Input length limitations prevent LLMs from processing many texts at once. However, an arbitrary number of texts can be handled by LLMs by splitting the texts into smaller subsets and processing them in parallel or sequentially. Our experimental results demonstrate that our methods can identify more coherent topics than existing ones while maintaining the diversity of the induced topics. Furthermore, we found that the inferred topics cover the input texts to some extent, while hallucinated topics are hardly generated.

1 Introduction

Topic modeling is the classical task of discovering latent topics that best describe a set of documents (Blei et al., 2003; Churchill and Singh, 2022). Recently, while neural topic models have worked successfully on various kinds of long documents (Miao et al., 2017; Srivastava and Sutton, 2017; Dieng et al., 2020), they have not been able to handle short texts, such as social media posts and news headlines (Li et al., 2016; Wu et al., 2022).

Large Language Models (LLMs), such as InstructGPT (Ouyang et al., 2022) and GPT-4 (OpenAI, 2023), have shown impressive results on various tasks by providing task instructions in a zero-shot manner (Wang et al., 2023; Kocoń et al., 2023). Since conventional topic models infer the topics of words by relying on word co-occurrence, they perform worse on short texts. In contrast, as LLMs contextually learn the meanings of words by pre-training on massive text corpora, they could accurately infer the latent topics.



(a) Parallel prompting

(b) Sequential prompting

Figure 1: Topic modeling with LLMs by splitting a document set into subsets and prompting (a) in parallel or (b) sequentially.

We propose two approaches to using LLMs for topic modeling: parallel prompting and sequential prompting (Figure 1). Due to the input length limitations of LLMs, an input document set must be split into smaller subsets, which are processed individually. Parallel prompting concurrently infers the topics of each subset and merges them to represent the topics of the whole document set. Sequential prompting processes each subset successively, updating the topics in every iteration. We assess our approaches across texts from various domains using multiple evaluation metrics.

The contributions of this study are as follows:

1. We propose parallel and sequential prompting methods for topic modeling using LLMs. Our methods can handle a large number of texts that cannot be processed in a single run due to the input length limitations of LLMs.
2. We validate the performance of our methods by comparing them with existing models and show that ours can identify more coherent topics than existing models while maintaining

Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 4: Student Research Workshop), pages 21–33

the diversity of the induced topics.

3. We assess the document coverage and factuality of the inferred topics, due to concerns that LLMs may focus on only parts of documents or generate hallucinated topics. Evaluation results indicate that those concerns are negligible.

2 Background

Topic modeling is the task of identifying latent topics as a set of topic words representing each topic from a collection of documents (Blei et al., 2003). Topic modeling has conventionally been tackled with probabilistic models such as latent Dirichlet allocation (LDA, Blei et al., 2003). In recent years, however, neural models have come into widespread use due to their high performance (Srivastava and Sutton, 2017; Dieng et al., 2020; Grootendorst, 2022).

It is known that topic modeling for short texts is difficult for current topic models due to data sparsity (Li et al., 2016; Wu et al., 2022). TSCTM (Wu et al., 2022) is a current state-of-the-art neural topic model for short texts. This model addresses data sparsity by learning representations of documents using VQ-VAE (van den Oord et al., 2017), contrastive learning, and incorporation of data augmentation into the learning.

BERTopic (Grootendorst, 2022) uses a pretrained encoder, Sentence-BERT (Reimers and Gurevych, 2019), to obtain clusters of documents and assigns topic words to each cluster by using a class-based TF-IDF procedure. Another related study is Stammbach et al. (2023), in which LLMs are utilized to automatically evaluate topic quality. However, our study is the first to explore how well LLMs perform topic modeling.

3 Topic Modeling with LLMs

We introduce two approaches to performing topic modeling with LLMs: **parallel prompting** and **sequential prompting**. For these approaches, we apply common preprocessing, which involves randomly splitting a document set into subsets with the same size, smaller than the context length of the LLMs.

Parallel Prompting In the parallel prompting, LLMs identify topics for each subset in parallel by prompting the subset and the instruction of topic modeling. The topics of each subset are then merged by LLMs. We use two kinds of prompts as shown in Table 1: (i) a $\textbf{Par}_{\textbf{TM}}$ prompt for parallel topic modeling for each subset, and (ii) a $\textbf{Par}_{\textbf{Mrg}}$ prompt for merging the topics from the results

ID	Prompt
$\textbf{Par}_{\textbf{TM}}$	Write the results of simulating topic modeling for the following documents: `[DOCS]`.
$\textbf{Par}_{\textbf{Mrg}}$	Write the results of merging the following topic modeling results:`[TOPICS]`,`[TOPICS]`, ...
$\textbf{Seq}_{\textbf{TM}}$	Write the results of simulating topic modeling for the following documents: `[DOCS]`, Make the most use of the following topics: `[TOPICS]`.

Table 1: Prompts for our methods. `[DOCS]` and `[TOPICS]` are replaced by a subset of documents and by previously identified topics, respectively.

Dataset	# of Documents	Text Length	Vocabulary Size
Tweet	2000	5.47	706
GoogleNewsT	11000	5.25	2376
StackOverFlow	19000	4.71	2544

Table 2: Dataset statistics. Each value is the average for five runs.

Sequential Prompting In the sequential prompting, LLMs identify topics for each subset sequentially, considering the topics previously identified for the previous subset. We use the $\textbf{Par}_{\textbf{TM}}$ for the first subset, then use a $\textbf{Seq}_{\textbf{TM}}$ prompt in Table 1 for the other subsets. This prompt contains topics identified in the prior subset and instructions for referring to them.

4 Experiments

We investigate how well our methods perform topic modeling for short texts.

4.1 Dataset

We employ three tokenized datasets provided by Zhang et al. (2021): GoogleNewsT (Rakib et al., 2020), Tweet (Yin and Wang, 2016), and StackOverFlow.[1] Following Wu et al. (2022), the datasets are preprocessed as follows: (i) characters are converted to lower case; (ii) words with two or fewer letters are removed; (iii) words appearing fewer than five times are filtered out. We then split each preprocessed dataset into subsets for

[1] https://www.kaggle.com/competitions/predict-closed-questions-on-stack-overflow/data?select=train.zip

Model	Tweet				GoogleNewsT				StackOverFlow			
	$K = 5$		$K = 15$		$K = 5$		$K = 15$		$K = 5$		$K = 15$	
	Cv	*TU*	*Cv*	*TU*	*Cv*	*TU*	*Cv*	*TU*	*Cv*	*TU*	*Cv*	*TU*
LDA	0.394	0.800	0.401	0.568	0.426	0.984	0.406	0.963	0.320	0.928	0.425	0.883
LDA_{Aug}	0.445	0.968	0.436	0.856	0.411	0.984	0.381	0.981	0.360	0.920	0.508	0.952
TSCTM	0.393	**1.000**	0.467	0.997	0.333	**1.000**	0.374	**1.000**	0.244	**1.000**	0.313	**1.000**
$\text{TSCTM}_{\text{Aug}}$	0.355	**1.000**	0.433	**1.000**	0.243	**1.000**	0.346	**1.000**	0.218	**1.000**	0.276	**1.000**
BERTopic	0.514	**1.000**	0.537	**1.000**	0.439	**1.000**	0.437	**1.000**	0.459	**1.000**	0.485	0.971
$\text{BERTopic}_{\text{Aug}}$	0.535	**1.000**	0.526	**1.000**	0.412	**1.000**	0.417	**1.000**	0.460	**1.000**	0.489	0.955
GPT-3.5$_{\text{Par}}$	0.476	0.992	0.532	0.900	0.571	0.960	0.535	0.913	0.312	0.864	0.496	0.913
GPT-3.5$_{\text{Seq}}$	0.552	0.960	0.515	0.920	0.562	0.984	0.489	0.948	0.441	0.896	0.517	0.775
GPT-4$_{\text{Par}}$	0.562	**1.000**	**0.576**	0.971	**0.618**	0.976	0.532	0.925	**0.466**	0.904	**0.571**	0.864
GPT-4$_{\text{Seq}}$	**0.577**	0.992	0.551	0.976	0.556	0.944	**0.561**	0.963	0.318	0.744	0.532	0.853

Table 3: Topic coherence (*Cv*) and diversity (*TU*) results under 5 and 15 topics ($K = 5$ and $K = 15$). LLM_{Seq} and LLM_{Par} correspond to the parallel and sequential topic modeling methods with LLMs, respectively. $\text{MODEL}_{\text{Aug}}$ corresponds the performance of the model with data augmentation. The maximum *TU* is 1.000 when topic words are totally distinct from each other. The best scores are shown in **bold**.

Model	Tweet				GoogleNewsT				StackOverFlow			
	$K = 5$		$K = 15$		$K = 5$		$K = 15$		$K = 5$		$K = 15$	
	DC	*Fa*	*DC*	*Fa*	*DC*	*Fa*	*DC*	*Fa*	*DC*	*Fa*	*DC*	*Fa*
LDA	**0.337**	**1.000**	0.561	**1.000**	0.488	**1.000**	0.664	**1.000**	**0.684**	**1.000**	**0.842**	**1.000**
LDA_{Aug}	0.307	**1.000**	**0.579**	0.997	**0.531**	**1.000**	**0.763**	**1.000**	0.659	**1.000**	0.838	**1.000**
TSCTM	0.176	**1.000**	0.388	**1.000**	0.405	**1.000**	0.740	**1.000**	0.141	**1.000**	0.480	**1.000**
$\text{TSCTM}_{\text{Aug}}$	0.187	**1.000**	0.331	0.987	0.309	**1.000**	0.608	0.979	0.419	0.888	0.441	0.888
BERTopic	0.293	**1.000**	0.471	**1.000**	0.433	**1.000**	0.748	**1.000**	0.656	**1.000**	0.796	**1.000**
$\text{BERTopic}_{\text{Aug}}$	0.303	**1.000**	0.468	**1.000**	0.422	**1.000**	0.749	**1.000**	0.637	**1.000**	0.795	**1.000**
GPT-3.5$_{\text{Par}}$	0.213	**1.000**	0.384	0.994	0.321	0.968	0.585	0.952	0.636	**1.000**	0.694	**1.000**
GPT-3.5$_{\text{Seq}}$	0.197	0.984	0.335	0.967	0.334	0.975	0.583	0.954	0.479	**1.000**	0.689	0.994
GPT-4$_{\text{Par}}$	0.241	**1.000**	0.402	**1.000**	0.392	**1.000**	0.661	0.995	0.578	**1.000**	0.754	**1.000**
GPT-4$_{\text{Seq}}$	0.224	0.983	0.403	0.994	0.373	**1.000**	0.660	0.951	0.554	0.931	0.626	0.883

Table 4: Document coverage (*DC*) and factuality (*Fa*) results under 5 and 15 topics ($K = 5$ and $K = 15$). Since baseline models without data augmentation discover topics based only on documents, the factuality values are 1.000.

our methods, setting the size at 1000[2] and truncating the remaining example. Table 2 shows the final statistics of the datasets we use. Note that baseline models take the union of subsets as input, and each subset contains different examples for each run.

4.2 Model

We evaluate our approaches with GPT-3.5 (gpt-3.5-turbo-0125) and GPT-4 (gpt-4-0125-preview) provided by the OpenAI API.[3] For baseline models, we employ the three models mentioned in Section 2: LDA[4], TSCTM[4], and BERTopic.[5] Additionally, we report the results of each baseline model with data augmentation. Regarding data augmentation techniques and the hyperparameters of TSCTM, we follow the original settings that were used in prior research (Wu et al., 2022).[6]

4.3 Evaluation

We evaluate the models under the condition that the number of topics is 5 or 15, and the number of topic words for each topic is 5. For evaluation metrics, we employ two widely used metrics for topic quality and two new metrics to assess possible issues of LLMs, i.e., the possibility of outputting topics reflecting only a very limited documents or hallucinated topics not included in documents. We run each model five times and report the average scores.

Topic Coherence and Diversity Following Wu et al. (2022), we calculate the coherence value[7] (*Cv*, Röder et al., 2015) with Wikipedia for topic coherence, and the topic uniqueness (*TU*, Nan et al., 2019) to assess the diversity in the inferred topics.

Document Coverage We are concerned that LLMs infer topics that reflect only a very limited

[2] In preliminary experiments, we checked the performance of our methods with subset sizes of 250, 500, and 1000. See Appendix A.3.

[3] In preliminary experiments, we also tried Llama 2 (Touvron et al., 2023), but we found that it was not sufficiently controllable for its output to be used in our approach. See Appendix A.2.

[4] https://github.com/BobXWu/TopMost

[5] https://maartengr.github.io/BERTopic

[6] The details can be found in Appendix B.1.

[7] https://github.com/dice-group/Palmetto

Model	Tweet		GoogleNewsT		StackOverFlow	
	$K = 15$		$K = 15$		$K = 15$	
	Cv	*DC*	*Cv*	*DC*	*Cv*	*DC*
GPT-3.5	0.532	0.366	0.517	0.569	0.464	0.634
GPT-4	0.580	0.395	0.523	0.665	0.519	0.747

Table 5: Average coherence (*Cv*) and document coverage (*DC*) of topics discovered by LLMs in parallel prompting without the merging process under 15 topics ($K = 15$). For each subset, we take the average of the values in five runs.

documents. Thus, we propose the metric *document coverage*, which measures the extent to which discovered topics cover documents. Document coverage is defined as follows:

$$DC = \frac{\#\left(d_{ref} \text{ that contains at least one } w_{topic}\right)}{\#\left(d_{ref}\right)}$$

where d_{ref} is a document within the reference document collection, and w_{topic} is the topic word constituting the outputted topics. A higher *DC* means that discovered topics cover more reference documents. In this experiment, we use the preprocessed datasets without augmentation as references.

Factuality Another potential issue is hallucination, where topics discovered by LLMs may not be included in given documents. Therefore we introduce *factuality*, which measures the degree to which topic words are composed from the vocabulary in the reference documents. Factuality is defined as follows:

$$Fa = \frac{\#\left(w_{topic} \text{ present in at least one } d_{ref}\right)}{\#\left(w_{topic}\right)}$$

A higher *Fa* indicates that more topic words are composed from the vocabulary in the reference documents. Note that the factuality could be less than one in existing topic modeling with data augmentation due to word substitution using out-of-vocabulary words of the documents.

5 Results and Discussion

Topic Quality Table 3 shows that the topics discovered by our methods are relatively high-quality both in terms of coherence (*Cv*) and diversity (*TU*).[8] For coherence in particular, GPT-4 achieved the state-of-the-art performance in all settings, with up to 40 % improvement. For instance, the scores on GoogleNewsT have risen by 41% (from 0.439 to 0.618) and 28% (from 0.437 to 0.561), respectively, for each setting of the number of topics.

Document Coverage Table 4 reports that LLMs showed relatively lower scores for document coverage (*DC*) than the best baseline models. This means that the topics discovered by LLMs often cover fewer documents than those discovered by the baseline models. However, note that there is a trade-off between topic coherence (*Cv*) and document coverage. For example, LDA_{Aug} achieved the highest coverage on GoogleNewsT but showed the lowest coherence, with the exception of TSCTM and $\text{TSCTM}_{\text{Aug}}$.

Factuality As shown in Table 4, LLMs showed lower scores for factuality (*Fa*) than the baseline models, particularly those without augmentation. This indicates that some topic words output by LLMs are not included in the documents. However, their factuality loss was less than 5% in almost all settings. Furthermore, we analyzed these non-existent words and found that most were not problematic enough to mislead topic interpretation; these include synonyms, derivatives, and related words of the ones in the documents.[9] This suggests that LLMs do not generate hallucinated topics that would cause misinterpretation of the content.

Parallel and Sequential Prompting Table 3 and Table 4 show that the parallel prompting approach can identify topics with better coherence and document coverage than the sequential prompting one. To analyze the superior performance of the parallel approach, we calculated *Cv* and *DC* of topics before merging. Table 5 shows that *Cv* and *DC* scores before merging were worse than those of the parallel approach, demonstrating that the merging process can improve both their coherence and document coverage. On the other hand, we analyzed the transition of topics during the sequential approach and then observe that it tended to update the previously identified topic very little due to strict adher-

[8] Examples of topics are given in Appendix C.1.

[9] Examples of non-existent words and analysis details are provided in Appendix C.2.

Model	#	Topics
BERTopic	**#1**	**kanye black thanksgiving west xbox**
	#2	china independence zone scotland air
	#3	hiv aarushi watkins ian woman
	#4	jellyfish robot seahorse flying methane
	#5	alzheimer brain infant risk gene
GPT-4_{Par}	**#1**	**kanye west kim kardashian parody**
	#2	**thanksgiving black friday shopping deal**
	#3	**xbox microsoft game console sale**
	#4	nokia lumia microsoft smartphone tablet
	#5	syria peace talk geneva conference
GPT-4_{Seq}	**#1**	**kanye west kim kardashian parody**
	#2	**black friday shopping thanksgiving deal**
	#3	**xbox game console playstation microsoft**
	#4	comet ison sun spectacular encounter
	#5	scottish independence salmond white paper

Table 6: Examples of topics discovered from GoogleNewsT when the number of topics and topic words is five, respectively. We have reordered the topics for illustrative purposes. **Bold topics** are mentioned in Section 5.

ence to our instructions, leading to lower document coverage compared with the parallel approach.[10]

Qualitative Analysis We conducted a qualitative analysis of the representative results that achieved the median topic coherence (*Cv*) across five trials using the GoogleNewsT dataset under five topics and five topic words. Table 6 demonstrates that BERTopic, the best baseline model for *Cv*, has the potential to identify topics encompassing multiple themes, while our methods using LLMs discover highly consistent and distinct topics. For instance, topic **#1** identified by BERTopic could be considered to contain three distinct themes (**Kanye West**, **Thanksgiving**, and **Xbox**), while GPT-4_{Par} and GPT-4_{Seq} effectively separated these into topics **#1**, **#2**, and **#3**, respectively.

6 Conclusion

In this study, we proposed two approaches to using LLMs for topic modeling: parallel prompting and sequential prompting. We implemented our methods on GPT-3.5 and GPT-4 and evaluated their performance on three datasets together with three existing topic models. In the evaluation, in addition to the well-known metrics for topic quality, we introduced two new metrics, document coverage and factuality, to assess the potential issues with LLMs reflecting only some documents or outputting hallucinated topics. The results showed that LLMs could find higher-quality topics than existing methods, and the impact of these issues was not remarkable in practice. Future work will include improving our methods to enable topic assignment to each document.

[10]Examples and further analysis are provided in Appendix C.3.

Acknowledgements

We thank the three anonymous reviewers for their helpful comments and suggestions, which improved this paper. This work was supported by JSPS KAKENHI Grant Number JP24H00809, Japan.

References

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022.

Rob Churchill and Lisa Singh. 2022. The evolution of topic modeling. *ACM Comput. Surv.*, 54(10s).

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

Adji B. Dieng, Francisco J. R. Ruiz, and David M. Blei. 2020. Topic Modeling in Embedding Spaces. *Transactions of the Association for Computational Linguistics*, 8:439–453.

Maarten Grootendorst. 2022. Bertopic: Neural topic modeling with a class-based tf-idf procedure. *Computing Research Repository*, arXiv:2203.05794. Version1.

Sosuke Kobayashi. 2018. Contextual augmentation: Data augmentation by words with paradigmatic relations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 452–457, New Orleans, Louisiana. Association for Computational Linguistics.

Jan Kocoń, Igor Cichecki, Oliwier Kaszyca, Mateusz Kochanek, Dominika Szydło, Joanna Baran, Julita Bielaniewicz, Marcin Gruza, Arkadiusz Janz, Kamil Kanclerz, Anna Kocoń, Bartłomiej Koptyra, Wiktoria Mieleszczenko-Kowszewicz, Piotr Miłkowski, Marcin Oleksy, Maciej Piasecki, Łukasz Radliński, Konrad Wojtasik, Stanisław Woźniak, and Przemysław Kazienko. 2023. Chatgpt: Jack of all trades, master of none. *Information Fusion*, 99:101861.

Chenliang Li, Haoran Wang, Zhiqian Zhang, Aixin Sun, and Zongyang Ma. 2016. Topic modeling for short texts with auxiliary word embeddings. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, page 165–174, New York, NY, USA. Association for Computing Machinery.

Yishu Miao, Edward Grefenstette, and Phil Blunsom. 2017. Discovering discrete latent topics with neural variational inference. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, page 2410–2419.

Feng Nan, Ran Ding, Ramesh Nallapati, and Bing Xiang. 2019. Topic modeling with Wasserstein autoencoders. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6345–6381, Florence, Italy. Association for Computational Linguistics.

OpenAI. 2023. Gpt-4 technical report. *Computing Research Repository*, arXiv:2303.08774. Version 3.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744.

Md Rashadul Hasan Rakib, Norbert Zeh, Magdalena Jankowska, and Evangelos Milios. 2020. Enhancement of short text clustering by iterative classification. In *Natural Language Processing and Information Systems*, pages 105–117, Cham. Springer International Publishing.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Michael Röder, Andreas Both, and Alexander Hinneburg. 2015. Exploring the space of topic coherence measures. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*, page 399–408, New York, NY, USA. Association for Computing Machinery.

Akash Srivastava and Charles Sutton. 2017. Autoencoding variational inference for topic models. In *International Conference on Learning Representations*.

Dominik Stammbach, Vilém Zouhar, Alexander Hoyle, Mrinmaya Sachan, and Elliott Ash. 2023. Revisiting automated topic model evaluation with large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9348–9357, Singapore. Association for Computational Linguistics.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *Computing Research Repository*, arXiv:2307.09288. Version 2.

Aaron van den Oord, Oriol Vinyals, and koray kavukcuoglu. 2017. Neural discrete representation learning. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Longyue Wang, Chenyang Lyu, Tianbo Ji, Zhirui Zhang, Dian Yu, Shuming Shi, and Zhaopeng Tu. 2023. Document-level machine translation with large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 16646–16661, Singapore. Association for Computational Linguistics.

Xiaobao Wu, Anh Tuan Luu, and Xinshuai Dong. 2022. Mitigating data sparsity for short text topic modeling by topic-semantic contrastive learning. In *Proceedings of the 2022 Conference on Empirical Methods*

in Natural Language Processing, pages 2748–2760, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Jianhua Yin and Jianyong Wang. 2016. A model-based approach for text clustering with outlier detection. In *2016 IEEE 32nd International Conference on Data Engineering (ICDE)*, pages 625–636.

Dejiao Zhang, Feng Nan, Xiaokai Wei, Shang-Wen Li, Henghui Zhu, Kathleen McKeown, Ramesh Nallapati, Andrew O. Arnold, and Bing Xiang. 2021. Supporting clustering with contrastive learning. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5419–5430, Online. Association for Computational Linguistics.

A Preliminary Experiments

In preliminary experiments, we tested different prompts and subset sizes to determine which maximize the performance of our methods.

A.1 Prompts

We first considered the **Par$_{\text{TM}}$** prompt and then proceeded to the **Par$_{\text{Mrg}}$** and the **Seq$_{\text{TM}}$** prompts.

Par$_{\text{TM}}$ We checked three kinds of prompts, which are shown in Table 7. Finally, we tentatively selected a **Direct** prompt as a **Par$_{\text{TM}}$** prompt, which achieved the highest performance. We also considered the effects from inserting the following phrases, which were expected to improve scores for topic coherence, diversity, and document coverage, respectively.

Cv "NOTE: Make top words for each topic likely to occur together in the documents"

TU "NOTE: Make the top words unique across topics."

DC "NOTE: Maximize the number of documents that contain at least one of the top words."

However, we found that none of these can positively influence LLMs' performance in our methods. Therefore, we selected a **Direct** prompt without phrase insertion as the **Par$_{\text{TM}}$** prompt.

Par$_{\text{Mrg}}$ Regarding the Par$_{\text{TM}}$ prompt, we created a **Base Par$_{\text{Mrg}}$** prompt, which has a similar structure to the Par$_{\text{TM}}$ (Table 8). We then considered the insertion of the following phrases:

Goal "We aim to identify topics for the entire document set by merging the topic modeling results for each subset."

Detail "NOTE: Outputs should reflect the topics before merging as much as possible. Output should contain topics that often appear before merging and not have ones that don't appear much before merging."

Experimental results showed our methods performed the best when we inserted both the *Goal* phrase and the *Detail* phrase into the **Base Par$_{\text{TM}}$**.

Consequently, we employed a **Base Par$_{\text{TM}}$** prompt with both phrases as the **Par$_{\text{TM}}$** prompt for the parallel approach.

Seq$_{\text{TM}}$ Similar to the prompt for parallel, we first created a simple **Base Seq$_{\text{TM}}$** prompt for the sequential approach in Table 8, after which we validated the effect from inserting the following phrases.

Goal "We aim to identify topics for the entire document set by sequentially updating tentative topics identified from each subset, considering topics identified just before from another subset."

Detail "NOTE: Outputs should be the same as the previous topics as much as possible. You can change them minimally only when the given documents don't include them much, and a new topic needs to be added to describe the documents."

We also found that the insertion of both of the above phrases was most effective at improving the performance of the sequential method. Thus, we utilized a **Base Seq$_{\text{TM}}$** prompt that incorporates both phrases as the **Seq$_{\text{TM}}$** prompt for the sequential approach.

A.2 Llama 2

In preliminary experiments, we also tried using Llama-2-7b-chat[11] and Llama-2-13b-chat[11] as LLMs for our methods and found that it is difficult for Llama 2 (Touvron et al., 2023) to perform topic modeling regardless of the prompts and the subset size we use. Table 9 shows the outputs of Llama 2 when given the **Par$_{\text{TM}}$** prompt with a subset size of 100 on GoogleNewsT. Llama 2 could not make adequate output for the number of topics and topic words in line with our instructions, while GPT-3.5 and GPT-4 could do so consistently under identical settings.

A.3 Subset Size

We used 250, 500, and 1000 as options for the subset size. It would be difficult for the subset size to exceed 1000 due to the context length of GPT-3.5 (gpt-3.5-turbo-0125), which we planned to use for the main experiments.

We ran the parallel and the sequential methods with GPT-3.5 on GoogleNewsT for each subset size. Table 10 presents the average scores of each method for five runs. There was a tendency for

[11] https://huggingface.co/collections/meta-llama/llama-2-family-661da1f90a9d678b6f55773b

topic coherence to improve as the subset size increased, but we could not discern any tendency for the other metrics. We ultimately selected 1000 as the subset size because the performance of each model was relatively high in all metrics under that setting.

Note, however, that using our proposed methods with the subset size of 250 or 500 could enable discovery of competitive or higher-quality topics compared with the existing models shown in Table 3 and Table 4. This suggests our methods could perform well regardless of the context length of LLMs applied them.

B Experimental Details

B.1 Implementation Details

We run TSCTM for 200 epochs. In the case without data augmentation, we run it with temperature as 0.5 and weight contrast as 1.0. In the case with data augmentation, we run it with temperature as 0.07, weight contrast as 3.0, and same quant as 0.001. For data augmentation, we apply WordNet[12] and Contextual Augmenter[3] (Kobayashi, 2018) with 30% word replacement, and filtered low-frequency words as in the preprocessing. Each Augmenter randomly replaces words in an input text with synonyms defined by WordNet and with words predicted by BERT (Devlin et al., 2019)[13], respectively. We utilized the original configurations of gpt-3.5-turbo-0125, gpt-4-0125-preview, and BERTopic without modification.

B.2 Examples of Prompts

Table 11 shows examples of prompts used in the experiment.

C Result Details

C.1 Examples of Topics

Following Wu et al. (2022), we randomly select some examples of topics identified by LDA, BERTopic, and our proposed methods with GPT-4.

C.2 Examples of Topic Words Not Included in the Documents

Table 13 shows examples of words not included in the documents outputted in topic modeling on GoogleNewsT. The bold portion of the GPT-3.5 outputs are the names of entities (e.g., **broncos**, **gree**, and **watson**) or words that do not exist in the real world (e.g., **dorffiefskee**). Such words are considered harmful because they may induce misinterpretation of topics. However, only a small number of such words were found, and most of them were synonyms, derivatives, or related words in the documents.

C.3 Examples of the Processing

Table C.3 shows specific the concrete examples of topics identified for each subset and the final output to demonstrate the processing in our methods. In the parallel approach, we find that LLM reasonably merges topics from each subset. For instance, bold topics in each subset are merged into one topic in the final output, using words from both subsets. On the other hand, in the sequential approach the final output is the same as the topics for the first subset except for the one pair of bold words. This indicates that LLMs with the the sequential approach could too strictly retain topics from the previous subset, and thus they cannot output topics that sufficiently reflect the entire set.

D Limitations

We do not thoroughly consider whether pretraining and instruction-tuning datasets of GPT-3.5 and GPT-4 might contain the datasets used in this study. Since topic modeling is an unsupervised task and we change the order of the samples randomly, we do not consider them able to utilize their knowledge about these datasets in our experiment.

[12] https://github.com/makcedward/nlpaug

[13] https://huggingface.co/bert-base-uncased

ID	Candidates for the Base Prompt Template
Direct	**Write the results of simulating topic modeling for the following documents**, each starting with "#." Assume you will finally identify `[NUM_TOPICS]` topics and use 5 top words for each topic. NOTE: Outputs must always be in the format "Topic k: word word word word word" and nothing else. """ `[DOCS]` """
Indirect	Discover latent `[NUM_TOPICS]` topics in the following documents, each starting with "#." For each topic, write 5 words extracted from input texts to show its meanings. NOTE: Outputs must always be in the format "Topic k: word word word word word" and nothing else. """ `[DOCS]` """
Direct$_{\text{reverse}}$	""" `[DOCS]` """ **Write the results of simulating topic modeling for the above documents**, each starting with "#." Assume you will finally identify `[NUM_TOPICS]` topics and use 5 top words for each topic. NOTE: Outputs must always be in the format "Topic k: word word word word word" and nothing else.

Table 7: Candidate prompts for Par_{TM}. `[DOCS]` and `[NUM_TOPICS]` are replaced by a subset of documents and by the number of topics.

ID	Base Prompt Template
Base Par$_{\text{TM}}$	Write the results of merging the following topic modeling results for each subset of the document set. Each result starts with "- n" and its topics start with "#" """ - 1 `[TOPICS]` - 2 `[TOPICS]` - 3 ... """

Base Seq$_{\text{TM}}$	Write the results of simulating topic modeling for the following documents, each starting with "#."
	Make the most use of the following topics previously identified from another set of documents, each starting with "Topic k:":
	"""
	`[TOPICS]`
	"""
	Assume you will finally identify `[NUM_TOPICS]` topics and use 5 top words for each topic.
	NOTE: Outputs must always be in the format "Topic k: word word word word word" and nothing else.
	"""
	`[DOCS]`
	"""

Table 8: Base prompts for the parallel and sequential methods. `[DOCS]`, `[TOPICS]`, and `[NUM_TOPICS]` are replaced by a subset of documents, previously identified topics, and the number of topics, respectively.

Model size	Examples of Llama 2 Output
7B	Topic 1: Top words: relief, challenge, face Topic 2: Top words: welker, concussion, test Topic 3: Top words: live, stream, champion, league Topic 4: Top words: bargain, black, friday, shopping Topic 5: Top words: scotland, independence, white, paper Note: Each topic is represented by 5 top words, which are the most frequently occurring words in the given documents.
13B	Topic 1: Disasters and Relief Efforts Topic 2: Sports and Injuries Topic 3: Technology and Gadgets Topic 4: Politics and Leadership Topic 5: Entertainment and Celebrities

Table 9: Examples of Llama 2 outputs when we provide Par_{TM} on GoogleNewsT under the conditions that the number of topics and topic words is five and the subset size is 100.

Subset Size	*Cv*	*TU*	*DC*	*Fa*
250	0.531	0.936	**0.241**	**1.000**
500	**0.572**	0.896	**0.241**	**1.000**
1000	0.571	**0.960**	0.213	**1.000**

(a) Parallel

Subset Size	*Cv*	*TU*	*DC*	*Fa*
250	0.524	0.976	**0.198**	**0.992**
500	0.529	**0.992**	0.193	0.976
1000	**0.562**	0.984	0.197	0.984

(b) Sequential

Table 10: Results of the parallel and sequential methods under five topics on GoogleNewsT for subset sizes of 250, 500, and 1000. The best scores are shown in **bold**.

ID	Prompt Example
Par_{TM}	**Write the results of simulating topic modeling for the following documents**, each starting with "#." Assume you will identify 5 topics and use 5 top words for each topic. NOTE: Outputs must always be in the format "Topic k: word word word word word" and nothing else. """ # philippine typhoon relief effort face challenge # wes welker concussion test bronco # basel chelsea live stream champion league watch ... # discus black friday shopping secret """
Par_{Mrg}	We aim to identify topics for the entire document set by merging the topic modeling results for each subset. **Write the results of merging the following topic modeling results for each subset of the document set.** Each result starts with "- n" and its topics start with "#" """ - 1 # comet ison thanksgiving sun solar # kanye west bound parody video # nokia lumia release mobile device # black friday shopping thanksgiving sale # alec baldwin msnbc cancellation defends ... - 11 # nokia lumia sale december phone # kanye west kim kardashian taylor # black friday deal best sales # irs rule political activity tax # bronco patriot win game rivalry """ Assume you will finally identify 5 topics and use 5 top words for each topic. NOTE: Outputs should reflect the topics before merging as much as possible. Output should contain topics that often appear before merging and not have ones that don't appear much before merging. NOTE: Outputs must always be in the format "Topic k: word word word word word" and nothing else.
Seq_{TM}	We aim to identify topics for the entire document set by sequentially updating tentative topics identified from each subset, considering topics identified just before from another subset. Write the results of simulating topic modeling for the following documents, each starting with "#." **Make the most use of the following topics previously identified from another set of documents, each starting with "Topic k:":** """ Topic 1: kanye west kim kardashian bound Topic 2: xbox black friday cyber monday Topic 3: hewlett packard nokia lumia company Topic 4: dancing star finale winner season Topic 5: syria peace talk china air """ Assume you will finally identify 5 topics and use 5 top words for each topic. NOTE: Outputs should be the same as the previous topics as much as possible. You can change them minimally only when the given documents don't include them much, and a new topic needs to be added to describe the documents. NOTE: Outputs must always be in the format "Topic k: word word word word word" and nothing else. """ # spacex falcon launch attempt # taylor swift princess gown winter white # redbox instant window phone appears nokia exclusive ... # google backed company selling dna analysis kit ordered sale """

Table 11: Examples of prompts used as Par_{TM}, Par_{Mrg}, and Seq_{TM} for topic modeling on GoogleNewsT under five topics.

Model	Examples of Topics
LDA	xbox microsoft game patriot bronco nokia lumia oldboy launch google kobe bryant chelsea lakers basel
TSCTM	macy parade hanukkah thanksgiving travel china zone african japan johansson bronco patriot packer welker illinois
BERTopic	china zone air nsa porn methane ant emission fire burning thanksgiving friday black comet parade
GPT-4$_{\text{Seq}}$	wes welker nfl concussion game nokia lumia window phone december nfl season game player concussion
GPT-4$_{\text{Par}}$	san andreas mobile game release nokia lumia tablet smartphone launch thanksgivukkah hanukkah holiday feast rare

Table 12: Examples of topics discovered from GoogleNewsT under 15 topics.

Model	Examples of Topic Words Not Included in the Documents
TACTM$_{\text{Aug}}$	twelvemonth sink railway blowout
GPT-3.5$_{\text{Seq}}$	**dorffiefskee broncos** patriots health advancement ocean guilty france legal attorney
GPT-3.5$_{\text{Par}}$	**gree watson** advertisement boat funding attorney declared refugees crash digital

Table 13: Examples of topic words not included in the documents when topic modeling on GoogleNewsT.

Subset 1
fishing fish bass fly report
superbowl commercial bowl super best
king speech oscar nomination award
facebook privacy setting user change
acai berry weight loss diet plan

Subset 2
fishing fish fly book saltwater
superbowl commercial doritos pepsi volkswagen
king speech oscar nomination award best
acai berry weight loss diet plan
christina aguilera national anthem super bowl

Final Output
fishing fish fly bass saltwater
superbowl commercial bowl pepsi doritos
king speech oscar nomination award
acai berry weight loss diet health
facebook privacy setting user change

(a) Parallel

Subset 1
fishing commercial superbowl fly bass
facebook privacy setting user **setting**
king speech oscar nomination award
berry acai weight diet loss
christina aguilera national anthem super

Final Output
fishing fly superbowl commercial bass
facebook privacy setting user **security**
king speech oscar nomination award
acai berry weight diet loss
christina aguilera national anthem super

(b) Sequential

Table 14: Topics identified for each subset and the final output by each method using GPT-4 on Tweet under five topics. **Bold words** are mentioned in Appendix C.3.