Topic Modeling for Small Data using Generative LLMs

Cascha van Wanrooij[1][0009−0002−5394−4489], Omendra Kumar Manhar[1][0009−0007−7834−5702], and Jie Yang[2][0000−0002−0350−0313]

[1] Kickstart AI, Molengraaffsingel 8 2629 JD Delft, Netherlands
[2] TU Delft, Mekelweg 5, 2628 CD Delft, Netherlands

Abstract. In this study, we introduce TopicGen, a topic modeling approach that leverages the extensive prior knowledge embedded in generative Large Language Models (LLMs) for small datasets ($N < 1000$). Unlike traditional methods like Latent Dirichlet Allocation (LDA) which often requires substantial data to discern thematic structures, our approach exploits the contextual understanding ability of LLMs for small data topic modeling while addressing their intrinsic limitations. Our approach first generates a preliminary set of topics using LLMs, followed by a consolidation phase that refines these topics into a coherent set to reduce redundancy and enhance thematic clarity. It then assigns each document to its best-fitting topic through classification. This three-phase approach allows the LLM to focus on one specific task at a time, reducing the potential for hallucinations in topic assignment and improving overall accuracy. We demonstrate that our approach not only overcomes the limitations of sparse data but also enriches topic modeling with the generative LLM's broad knowledge, showing improved thematic extraction and relevance abilities over traditional topic modeling methods and competitive performance to other neural methods.

Keywords: Topic Modeling · Large Language Models · Generative AI

1 Introduction

Topic modeling is a Natural Language Processing (NLP) technique used to automatically identify themes or topics within a body of text, enabling a better understanding of large text datasets [20]. This process is particularly valuable for understanding and organizing vast amounts of textual information. However, traditional topic modeling methods often struggle with small datasets, limiting their effectiveness in scenarios where data is scarce. Motivated by the practical challenge of creating a website that automatically aggregates daily machine learning news, we set out to develop an approach to handle topic modeling on small datasets. Given the limited number of newsworthy events in this field on a daily basis, existing topic modeling approaches proved inadequate for our needs.

Existing approaches to topic modeling include statistical methods like Latent Dirichlet Allocation (LDA) and Non-negative Matrix Factorization (NMF), which have been widely used but often struggle with small datasets[3]. This is because these methods rely on learning word co-occurrence patterns from the data itself. More recent neural topic models have shown improvements on some metrics [7]. For instance, BERTopic, a modular framework that clusters document embeddings created by an embedding LLM to discover topics [8], has gained popularity. These methods leverage pre-trained embedding models, which already contain information about word relationships in text, potentially improving performance on smaller datasets. However, these methods rely on embeddings and dimension reduction techniques that are optimized for general semantic similarity, not specifically for topic modeling. This similarity doesn't necessarily correspond to topical relationships. Similarly, dimension reduction optimizes for maximum variance preservation rather than topic distinction. In contrast, our approach uses generative LLMs that are explicitly prompted (e.g., conditioned) to focus on topic identification/reduction, potentially allowing for improved performance.

Large Language Models (LLMs) have recently demonstrated remarkable capabilities in various NLP

[3] We define these as datasets with less than a thousand samples.

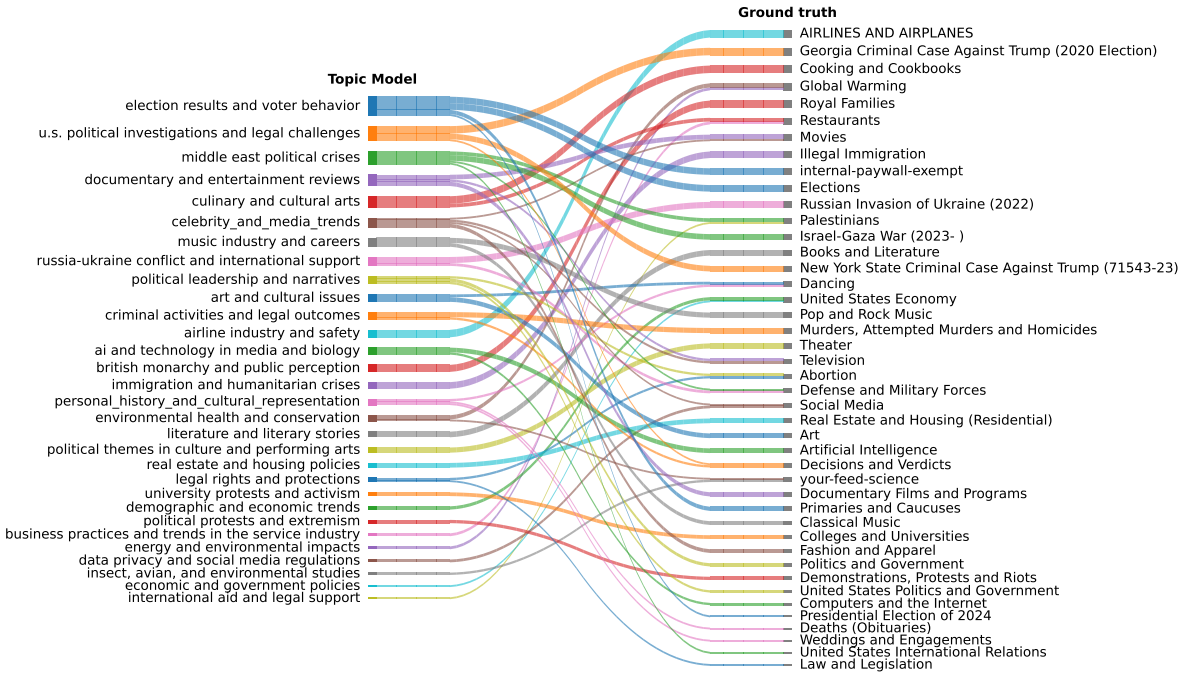


Fig. 1: Comparison of topics identified by TopicGen with the ground truth on the New York Times dataset ($N = 800$). We visualize the 55 most common (topic label, ground truth label) pairs.

tasks, often outperforming specialized models. These generalist models, benefiting from large pre-training datasets, require less fine-tuning data to generalize effectively. For example, embedding models like BERT have surpassed traditional methods in tasks like sentiment analysis [6], while generative LLMs, such as the GPT family, have further accelerated this trend [4]. Generative LLMs' ability to understand context and generate human-like text suggests potential for improving topic modeling, especially in data-sparse scenarios. However, applying generative LLMs to topic modeling presents two major challenges:

1. **Hallucinations:** LLMs can generate plausible but incorrect information [21]. When tasked with clustering documents into topics, an LLM might invent topics not prevalent in the text or assign non-existent topics to documents. This compromises the reliability of the topic modeling process.
2. **Context utilization:** LLMs struggle to effectively use information from the middle of long prompts, a phenomenon known as the 'Lost in the Middle' problem [10]. This limitation hinders the LLM's ability to generate a comprehensive set of topics that fully represents all themes present in the given documents.

To address these challenges, we introduce TopicGen[4], a generative LLM-based topic modeling framework that breaks the process of topic modeling down into three phases:

[4] Code and data available at `https://github.com/ICascha/topic_modeling_paper/tree/main`.

1. **Candidate topic creation:** We provide the LLM with chunks of documents, prompting it to generate a set of candidate topics for each chunk. After the model has done this for all chunks, we merge the topic sets together to form a final candidate topic set.
2. **Topic reduction:** We then prompt the LLM to distill the large set of candidate topics into a smaller set of core topics. We explore multiple strategies to tackle this.
3. **Topic classification:** Finally, we provide the LLM with one document at a time, as well as the final list of topics and prompt it to select the topic that is best-fitting for that document. In the end, we obtain a classification of topics over all documents.

By providing the model with chunks of documents, we mitigate the 'Lost in the middle' problem. Furthermore, breaking down the topic modeling process into concrete steps reduces the possibility of hallucinations. For instance, even if an LLM generates a hallucinated topic in the initial step, it is unlikely to classify documents into that topic during the final step if more appropriate topics are available. Similar approaches have been explored recently [14], [16], [19]. Our work, TopicGen, contributes the following design ideas and findings to the literature:

- We perform zero-shot prompting, providing the model with no example topics. We believe this is important to test as common topic modeling methods like LDA and BERTopic by default require no prior topics making for a more direct comparison.
- We evaluate LLMs' capability to perform topic modeling outside of their knowledge cutoff by constructing datasets based on recently published documents, such as New York Times abstracts. This is important as LLMs have the capability of memorizing significant chunks of their internet-scale training dataset (e.g., [2]). This introduces the possibility of data leakage of labels, especially in common datasets such as 20Newsgroups.
- We assign topics in a probability-based way using the generative LLM's logits, and show that topic probability significantly aligns with empirical probability for topic modeling.
- We provide a strong baseline by replacing BERTopic's default embedding model `all-MiniLM-L6-v2` with a model, `text-embedding-3-large`, which scores significantly higher on the ubiquitous Massive Text Embedding Benchmark (MTEB) [11].

We show that under these evaluation criteria, TopicGen (and other frameworks like it) can offer strong performance benefits over traditional topic modeling methods, and offer a compelling alternative to current embedding-based methods with a trade-off in performance versus cost. On average, over the datasets we have evaluated, TopicGen comes out on top when comparing using the V-measure against a human-curated golden set, with a significant and sizable gap in news datasets. In the end, through qualitative evaluation, we hypothesize that because of their extensive pre-training generative LLMs can facilitate the creation of more human-aligned topics in data-sparse scenarios as highlighted in Figure 1.

2 Methodology

2.1 TopicGen

TopicGen consists of three phases: Candidate topic creation, topic reduction, and topic classification. Each phase functions as follows:

1. **Candidate topic creation**: The goal is to create a list of topics $T = [t_1, t_2, ..., t_n]$, such that every document in the corpus can be characterized by a probability distribution over T. Each article should be best represented by one topic in this set. We divide our N documents into M equally-sized chunks. For each chunk, we use a generative LLM to distill a set of topics, a process we call CANDIDATE_CREATION. We then combine the topics from CANDIDATE_CREATION into a preliminary set.
2. **Topic reduction**: We present a generative LLM with a set of preliminary topics and prompt it to distill a list of core topics.

3. **Topic assignment**: After finalizing the set of topics T, we classify each document based on the highest likelihood principle. We present a generative LLM with an article and an indexed list of topics T, prompting it to estimate the most fitting topic index i. The topic $T[i]$ is then selected as the classification for that article. Alternatively, we can derive a probability distribution over T by analyzing the logits produced by the model during the estimation of topic index i. We refer to this function as TOPIC_ASSIGNMENT.

The process is formalized in Algorithm 1 and the prompts we used are shown in Appendix B. To produce consistent output, we prompt the generative LLM to respond in a JSON format. We use a temperature of 0 for each generative LLM tested.

Initially, we explored an iterative topic reduction strategy where we prompted the generative LLM to select the two most similar topics to merge into a new topic until a set number of topics was reached. However, this strategy did not perform better than a simple strategy merging topics in a single step (See Table 1). Consequently, we focus on the single-step strategy due to its lower computational costs. We also listed the prompt of this strategy in Appendix B.

Algorithm 1 TopicGen

Require: Dataset D of size N and generative LLM L, and a target number of topics NT
Divide D into chunks of size M: $\{C_1, C_2, ..., C_{\lceil M/N \rceil}\}$ truncating each article if necessary
topic_list $\leftarrow \emptyset$
for $i = 1$ to $\lceil M/N \rceil$ **do**
    topic_list $\leftarrow$ topic_list + CANDIDATE_CREATION(C_i, L)
end for
topic_list $\leftarrow$ TOPIC_REDUCTION$(topic_list, L, NT)$
for $i = 1$ to N **do**
    TOPIC_ASSIGNMENT$(D_i, topic_list, L)$
end for

By dividing our corpus into chunks, we prevent the model from processing all articles simultaneously while still allowing it to estimate the diversity within the corpus. In each experiment, we match the number of topics to the ground truth label set size and set the chunk size to $\lceil N/8 \rceil$. Adjusting the chunk size affects the generalization of topics: larger chunk sizes typically yield more general topics since the pool of candidate topics decreases.

In handling topic reduction in one step, the LLM must arrive at a set of core topics without iterative refinement. While this approach might be suboptimal, we consider it a reasonable compromise given a generative LLM's limited capacity for long-term planning [17]. This strategy balances computational efficiency with the model's ability to identify a coherent and complete set of topics.

2.2 Compute costs

The total number of calls to a generative LLM with TopicGen is equal to:

$$\lceil N/M \rceil \text{ (Candidate topic creation) } + 1 \text{ (Topic reduction) } + N \text{ (Topic assignment)}$$

which is in the order of N. However, the generative LLM is presented with each document twice, making the method potentially expensive for long documents, especially considering the quadratic complexity of transformer models [18]. In such cases, strategies like document truncation may be applied to mitigate computational costs.

TopicGen is significantly more computationally intensive than classical methods such as NMF [1], LDA [3],

or even BERTopic, which only requires a single pass of an embedding LLM over each document. Therefore, we recommend using this method primarily on small datasets, where the extensive language and knowledge embedded in generative LLMs can be fully leveraged. As the corpus size increases, the corpus itself can provide more insights into the relationships between documents. Consequently, we hypothesize that TopicGen will show better performance on small datasets compared to traditional methods, but this advantage may diminish as the corpus size grows.

2.3 Evaluation

To experimentally verify whether TopicGen outperforms traditional methods on small datasets, we evaluate each topic modeling method against a ground truth topic classification using the V-measure, which is the harmonic mean of homogeneity and completeness of the clusters created by the topic modeling method relative to the ground truth. The V-measure is defined as:

$$V = 2 \cdot \frac{homogeneity \cdot completeness}{homogeneity + completeness}$$

Homogeneity measures the extent to which each cluster contains only members of a single class, while completeness assesses whether all members of a given class are assigned to the same cluster. Intuitively, a perfect clustering would have both high homogeneity (each cluster contains only one topic) and high completeness (all documents of one topic are in the same cluster). Our choice of the V-measure is straightforward: it provides a comprehensive and easily interpretable metric. Furthermore, we focus solely on the clustering aspect of topic modeling, rather than the quality of the labels. This decision is based on the consideration that evaluating label quality using the common coherence metric might bias the results in favor of neural methods [9]. Similarly, using an LLM like GPT-4o to evaluate label quality could also favor neural methods. By comparing results to human-created ground truth datasets, we aim to offer an unbiased, purely data-driven perspective on the human alignment of the method.

For our evaluations, we select the following methods:

- **TopicGen**: We vary several setup options:
  - We use GPT-4o and GPT-4o-mini to assess how the choice of model affects the results and to test the robustness of our method with different model sizes.
  - We implement two different topic reduction approaches: A default setup, following the topic reduction process outlined in Algorithm 1. And an iterative merging strategy, where the model successively combines the two most similar identified topics until reaching the desired number of topics.
- **Non-negative Matrix Factorization (NMF)**: This technique is known to perform better than LDA with shorter texts [5]. We use a vocabulary size of 1000 and employ sklearn's feature to remove English stopwords before tokenizing, leaving all other hyperparameters at default values.
- **BERTopic**: This neural topic modeling method generates topics by clustering LLM embeddings. We use OpenAI's `text-embedding-3-large` for the embedding phase of BERTopic. To address BERTopic's tendency to generate relatively few topics in its default configuration, we use the settings `min_cluster_size=2` and set `nr_topics` to match the ground truth.
- **LDA**: We use Gensim's library implementation for LDA. We leave all tuning options on default values (`auto`), using the NLTK library to perform lemmatization and outlier removal with a vocabulary size of 1000.

In this study, we didn't perform hyperparameter optimization. Due to cost constraints, it was not possible to optimally tune TopicGen on matters such as the chunk size or the prompts used; therefore, we used reasonable hyperparameter choices for all methods to yield a fair comparison. In addition to the modeling methods, we conduct experiments on four datasets. First, we utilize the well-known 20Newsgroups dataset, which is commonly used for testing topic modeling methods. This dataset comprises 20 categories covering 18,846 newsgroup documents. We also create three additional datasets that fall outside of GPT-4o's current knowledge cutoff of October 2023; all texts contained within these datasets were published in 2024. These datasets are:

- **New York Times (NYT):** We use the New York Times archive API to obtain summaries of articles published between January 2024 and May 2024, along with their associated most relevant keyword as ground-truth topic labels. This dataset contains 7,805 documents in total, as we take a look at the most common 50 keywords.
- **arXiv (ARXIV):** Using the arXiv API, we collect the 100 most popular abstracts from January 2024 to August 2024 (when available) for each subcategory of Computer Science. We use these arXiv subcategories as the topic labels for this dataset.
- **PubMed (PUBMED):** For this biomedical literature dataset, we use the PubMed API to gather the 100 most popular abstracts from January 2024 to August 2024 (when available) for each MeSH (Medical Subject Headings) subheading. There are 76 subheadings, which broadly specify the subject covered. These subheadings serve as the topic labels for this dataset.

For a given dataset size N, we randomly sample an equal number of articles within each category to ensure that each topic is represented. We then shuffle the resulting dataset to ensure the order of articles is random. While we recognize that a topic model displaying a high degree of similarity to human-curated ground truth does not directly imply better topics and vice versa, we assume that on average, human-curated topics will provide a strong signal of how the model should ideally cluster articles.

We refrain from using common topic modeling evaluation metrics, such as the ubiquitous C_v metric, because these metrics do not compare the generated topics to ground truth, but rather evaluate the keywords chosen to represent the topics. Since the technique of topic representation can be swapped out for all evaluated methods, we focus solely on evaluating how documents are clustered into topics, which is captured by the V-measure. For the dataset size N, we chose to test sizes of 400 and 800. These sizes represent small datasets where we hypothesize that TopicGen will excel, while also keeping our compute costs relatively affordable. For each combination of method, dataset, and data size, we randomly sample 5 dataset subsets for all methods [5] and compute the V-measure for evaluation, allowing us to assess the consistency and reliability of our results across multiple trials.

3 Results and Discussion

3.1 Quantitative results

We present the results for all methods as averages over all datasets along with 95%-confidence intervals, in Figure 2. In Table 1 and 2 we present the results for $N = 800$ and $N = 400$ respectively.

TopicGen, on average, outperforms or matches all other tested approaches; at $N = 800$, there is a significant ($\alpha = 0.05$) gap for TopicGen compared to BERTopic, for the setup using GPT-4o. Looking at results for individual datasets we observe a relatively large performance increase in V-measure for both the Newsgroups and NYT datasets versus other methods, with our method producing more complete and homogeneous topics relative to ground truth. The average performance gap relative to BERTopic vanishes for the PUBMED and ARXIV datasets. Contrary to expectations, the performance gap widens as dataset size increases, rather than diminishing. Surprisingly, we observe only a modest performance increase in moving from GPT-4o-mini to GPT-4o. Additionally, we observe no performance increase for the more complex iterative merging strategy.

We attribute TopicGen's improved performance primarily to the extensive knowledge embedded in generative language models. Unlike traditional approaches such as LDA and NMF, which rely solely on corpus-specific information, generative LLMs leverage pre-existing knowledge about world events, enabling pattern recognition with fewer articles. Furthermore, these models are implicitly aligned through training on human-generated data and explicitly through techniques like Reinforcement Learning from

[5] Except TopicGen using the initially explored iterative merging strategy, for which we only perform one run per dataset given the computational cost.

dataset	metric name model	V-measure	completeness	homogeneity
NYT	BERTopic	0.54 (0.015)	0.59 (0.014)	0.49 (0.017)
	LDA	0.37 (0.005)	0.37 (0.005)	0.37 (0.005)
	NMF	0.43 (0.01)	0.44 (0.01)	0.42 (0.01)
	TopicGen (4o-mini)	0.58 (0.022)	0.61 (0.022)	0.54 (0.024)
	TopicGen (GPT-4o)	**0.62** (0.012)	**0.65** (0.01)	**0.59** (0.015)
	TopicGen* (GPT-4o)	0.57	0.60	0.55
20NEWSGROUPS	BERTopic	0.48 (0.039)	0.57 (0.03)	0.42 (0.044)
	LDA	0.12 (0.01)	0.12 (0.01)	0.12 (0.01)
	NMF	0.29 (0.023)	0.29 (0.023)	0.28 (0.023)
	TopicGen (4o-mini)	0.57 (0.009)	0.60 (0.008)	0.54 (0.015)
	TopicGen (GPT-4o)	**0.58** (0.017)	**0.61** (0.017)	**0.55** (0.017)
	TopicGen* (GPT-4o)	0.48	0.51	0.44
ARXIV	BERTopic	**0.51** (0.015)	**0.56** (0.016)	0.46 (0.015)
	LDA	0.28 (0.003)	0.29 (0.003)	0.28 (0.003)
	NMF	0.41 (0.016)	0.42 (0.017)	0.4 (0.015)
	TopicGen (4o-mini)	0.50 (0.032)	0.51 (0.024)	**0.48**(0.039)
	TopicGen (GPT-4o)	0.48 (0.039)	0.51 (0.031)	0.46 (0.045)
	TopicGen* (GPT-4o)	0.48	0.51	0.46
PUBMED	BERTopic	0.48 (0.011)	**0.56** (0.008)	0.42 (0.013)
	LDA	0.42 (0.005)	0.44 (0.005)	0.39 (0.005)
	NMF	0.45 (0.01)	0.49 (0.009)	0.42 (0.01)
	TopicGen (4o-mini)	0.49 (0.02)	0.54 (0.016)	0.45 (0.023)
	TopicGen (GPT-4o)	0.50 (0.038)	0.54 (0.021)	0.46 (0.052)
	TopicGen* (GPT-4o)	**0.51**	0.55	**0.47**

Table 1: Table with results for $N = 800$. Standard deviations are provided between brackets.
* = iterative merging strategy applied

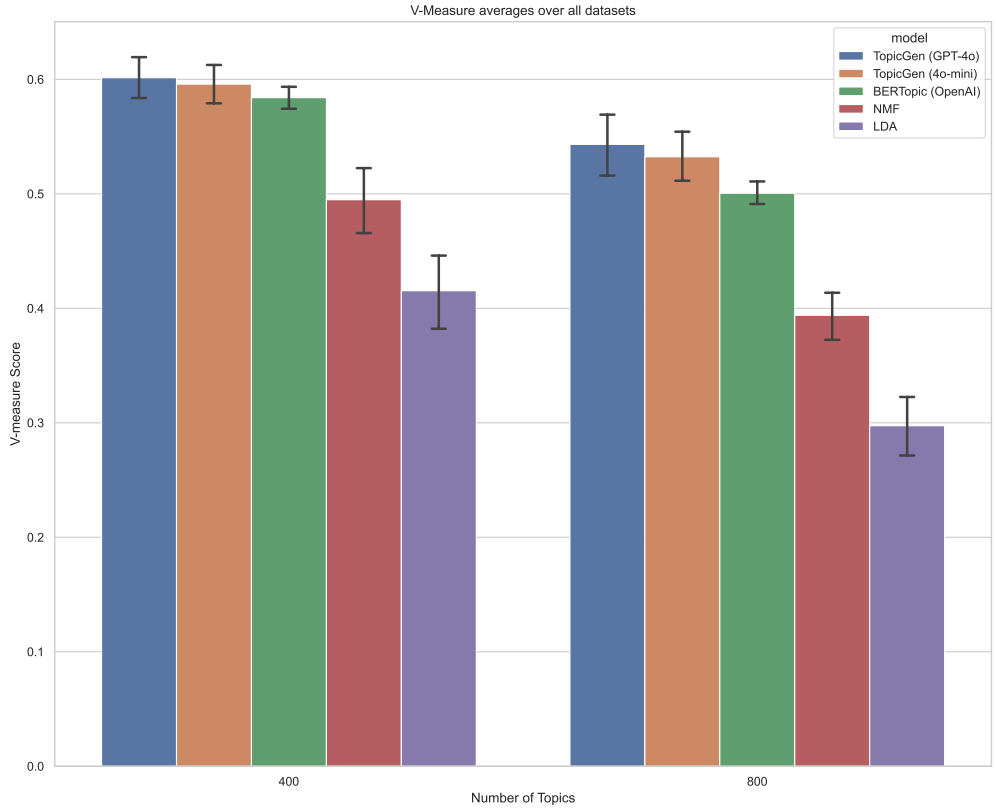


Fig. 2: Average V-measure computed over all datasets with 95% confidence intervals.

Human Feedback [13] (RLHF), resulting in strong alignment with human-created topics.

With a strong embedding model, we observe a relatively modest gap between TopicGen and BERTopic. This is understandable since BERTopic's embedding model introduces a significant prior knowledge, similar to that of generative LLMs. The remaining performance gap could then be explained by various hypotheses, most predominantly the idea that despite `text-embedding-3-large` delivering near state-of-the-art performance in the Massive Text Embedding Benchmark (MTEB), significantly more resources may be invested in generative models such as GPT-4o, resulting in a more comprehensive prior knowledge[6].

We show that the performance gap of neural methods such as TopicGen and BERTopic cannot be explained solely by dataset leakage, as both methods outperform traditional methods on datasets outside of their respective knowledge cutoffs on the NYT dataset. The absence of a large performance gap for the PUBMED and ARXIV datasets, observed for both TopicGen and BERTopic compared to traditional methods may be attributed to the increased complexity of scientific abstracts compared to news

[6] While we cannot provide direct evidence for this, as of writing, the current best-performing embedding model on the MTEB benchmark is `bge-en-icl` with 7 billion trainable parameters, significantly smaller than state-of-the-art generative models such as the largest LLama 3.1 with 405 billion parameters.

article summaries, as well as the lower degree of variance between documents in topics within specific scientific fields. Current state-of-the-art LLMs may still lack the necessary performance to distinguish between these more nuanced texts, potentially falling back on more superficial characteristics similar to LDA/NMF approaches on more complex datasets.

We further investigated our method's capability to probabilistically assign topics. To evaluate GPT-4o's probabilistic assignments against empirical correctness, we used the ground-truth labels from ARXIV as input for the TOPIC_ASSIGNMENT function. This approach allowed us to construct a calibration chart, presented in Figure 3. In this chart, we compare GPT-4o's given confidence level with its empirical confidence: the ratio of correct answers to the total number of predictions at each confidence level.

The results reveal a clear correlation between the model's assigned probability for a topic and its empirical accuracy, with a Pearson's r coefficient (referred to as σ) of 0.64. However, this relationship deviates from a one-to-one correspondence ($\sigma = 1$). Notably, the model exhibits overconfidence, with this overconfidence gap widening significantly when the model's assigned probability falls below 0.95.

This observation aligns with findings from Figure 8 in the GPT-4 technical report [12], suggesting that the RLHF process may induce this overconfidence. This raises the possibility that an unaligned model might potentially perform better in topic assignment tasks.

Despite this overconfidence, a consistent relationship persists between the model's probability predictions and empirical accuracy. This indicates that the output probabilities could still prove valuable in practical applications, such as filtering out documents for which the model expresses low confidence in its topic assignments.

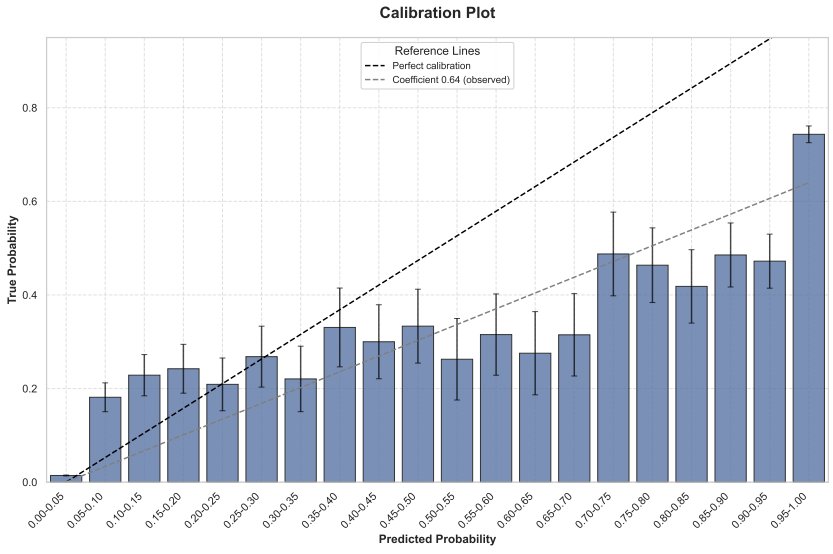


Fig. 3: Calibration plot of GPT-4o on the ARXIV dataset.

3.2 Qualitative evaluation

In examining the qualitative results, we compare TopicGen's clustering to the ground truth clustering on the New York Times dataset, as depicted in Figure 1. TopicGen successfully identifies topics with a granularity similar to that of human clustering. Certain topics, such as those related to airlines or fashion, align almost directly with the ground truth.

For more abstract topics, such as those pertaining to US elections and Middle Eastern conflicts, TopicGen tends to combine various ground truth topics into broader themes. Despite this, the consolidated topics still reflect underlying themes. For instance, articles related to the Israel-Gaza conflict are frequently merged into a single topic, irrespective of the article's specific perspective, be it the conflict itself or its broader implications (on e.g., US relations).

Figures 5 and 6 illustrate the performance of BERTopic and NMF, the two alternative best-performing methods. NMF exhibits errors that are readily apparent to human observers. For instance, it combines articles about movies and politics into a single topic, groups articles about war and climate change together, and clusters articles related to various New York City events or places without any other unifying themes (e.g., articles about restaurants, dancing, and documentaries). These errors could stem from the similar words or expressions used across these diverse articles, thereby highlighting a weakness inherent in methods like NMF or LDA that rely primarily on word co-occurrence patterns.

BERTopic, leveraging its context-aware embedding model, avoids such obvious mistakes. However, it tends to generate numerous outliers, especially for smaller datasets. These outliers are particularly prevalent in articles assigned to ground truth topics with high diversity, such as book and movie reviews, as well as articles about politics and the economy. TopicGen demonstrates better performance with these diverse topics. We hypothesize that this is due to the generative LLM's ability to identify the central theme of an article (e.g., recognizing that both horror and drama movie reviews fall under the broader scope of movie reviews). This task is more challenging for encoding models like `text-embedding-3-large`, which typically average or max-pool output states [15], resulting in a vector that represents all content of the article more equally. Consequently, two political articles discussing vastly different issues might end up with entirely different vector representations in BERTopic, while TopicGen can more easily recognize their shared political nature.

For the ARXIV dataset, whose comparison to ground truth we highlight in In Figure 4, we observe that TopicGen again achieves a level of topic granularity similar to the ground truth. However, some topics exhibit less justifiable generalization compared to the NYT dataset. For instance, the largest generated topic, 'formal methods and logic', encompasses articles related to logic, automata, discrete mathematics, and programming languages. While these topics share thematic similarities, they are distinct fields. For articles that are difficult to assign to a single topic, BERTopic tends to label them as outliers. We note that BERTopic combines topics on sound, computer vision, information retrieval, and machine learning, likely due to the similar machine learning methods employed in recent studies across these fields. NMF demonstrates a lower tendency for overgeneralization and performs well on topics with highly specific terms, such as robotics and cryptography. However, it struggles when handling articles with similar terms, for example, combining articles on numerical analysis and sound processing. Overall, we observe that the ARXIV and PUBMED datasets present greater ambiguity, which may explain the smaller performance gap between all methods on these datasets, as evident in Table 1.

In general, we observe that TopicGen creates topics that have a similar granularity to human-created ground truth and contain fewer obvious errors or outliers than alternative methods. The tendency for overgeneralizing suggests there might still be room for improvement in the topic reduction step of our framework.

4 Conclusion

We have introduced TopicGen, a framework that leverages generative LLMs for topic modeling. Our results demonstrate that, on average, TopicGen outperforms both traditional topic modeling methods and BERTopic, a state-of-the-art approach, when evaluated against human-labeled ground truth on small datasets. Notably, this performance increase is achieved without employing in-context learning or fine-tuning techniques.

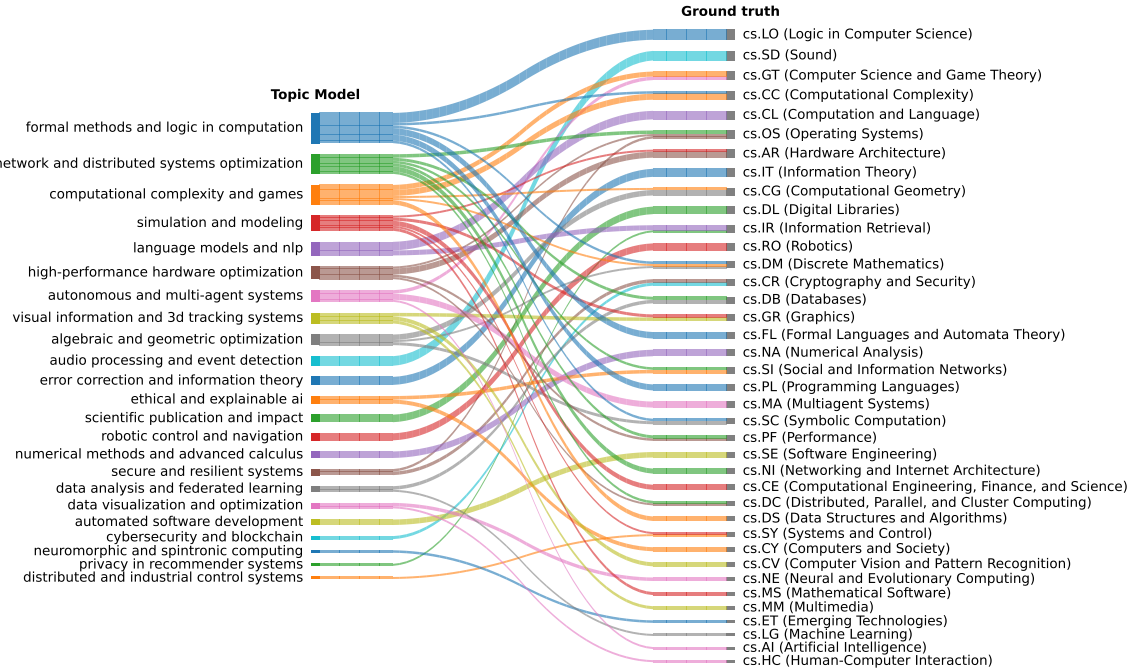


Fig. 4: Comparison of topics identified by TopicGen with the ground truth on the Arxiv dataset ($N = 800$). We visualize the 55 most common (topic label, ground truth label) pairs.

Our findings indicate that the improved performance of TopicGen is not solely attributable to the memorization of training data. Rather, we posit that the performance of our framework, and similar approaches, stems from the extensive prior knowledge embedded within generative LLMs. This observation aligns with a broader trend in machine learning where more general models increasingly surpass specialized models across various tasks.

Qualitative evaluation reveals that TopicGen makes fewer obvious errors in topic assignment compared to alternative methods and successfully generates topics with comparable granularity to human-labeled ones. However, we note that TopicGen tends to overgeneralize when dealing with closely related topics. This suggests potential areas for future research, particularly in exploring more optimal merging strategies. Interestingly, our experiments indicate that an iterative strategy does not significantly improve performance, warranting further investigation.

We propose that datasets with a degree of thematic overlap between topics serve as particularly interesting evaluation cases, as they present non-trivial challenges for topic modeling methods. While our quantitative metrics provide valuable insights, human evaluation may offer a more nuanced understanding of the performance gaps between models on these complex datasets. Such evaluations could potentially reveal subtle differences in model performance and provide insights into areas for improvement.

Finally, we demonstrate that the model's logits can be utilized to assign topics probabilistically. In addition to this, TopicGen's ability to perform effectively without extensive tuning in a zero-shot manner makes it particularly suitable for scenarios requiring topic modeling over small, diverse document sets, such as automated news topic extraction. In these cases, the limited scale of the datasets allows one to leverage the generative LLM's extensive prior knowledge while minimizing concerns about computational expenses.

Acknowledgments. This study was funded by Kickstart AI.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Arora, S., Ge, R., Halpern, Y., Mimno, D., Moitra, A., Sontag, D., Wu, Y., Zhu, M.: A practical algorithm for topic modeling with provable guarantees. In: Dasgupta, S., McAllester, D. (eds.) Proceedings of the 30th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 28, pp. 280–288. PMLR, Atlanta, Georgia, USA (17–19 Jun 2013), https://proceedings.mlr.press/v28/arora13.html
2. Balloccu, S., Schmidtová, P., Lango, M., Dušek, O.: Leak, cheat, repeat: Data contamination and evaluation malpractices in closed-source llms (2024), https://arxiv.org/abs/2402.03927
3. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. Journal of machine Learning research **3**(Jan), 993–1022 (2003)
4. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., Amodei, D.: Language models are few-shot learners. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H. (eds.) Advances in Neural Information Processing Systems. vol. 33, pp. 1877–1901. Curran Associates, Inc. (2020), https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf
5. Chen, Y., Zhang, H., Liu, R., Ye, Z., Lin, J.: Experimental explorations on short text topic mining between lda and nmf based schemes. Knowledge-Based Systems **163**, 1–13 (2019)
6. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding. In: Burstein, J., Doran, C., Solorio, T. (eds.) 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). pp. 4171–4186. Association for Computational Linguistics, Minneapolis, Minnesota (Jun 2019). https://doi.org/10.18653/v1/N19-1423, https://aclanthology.org/N19-1423
7. Doan, T.N., Hoang, T.A.: Benchmarking neural topic models: An empirical study. In: Zong, C., Xia, F., Li, W., Navigli, R. (eds.) Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021. pp. 4363–4368. Association for Computational Linguistics, Online (Aug 2021). https://doi.org/10.18653/v1/2021.findings-acl.382, https://aclanthology.org/2021.findings-acl.382
8. Grootendorst, M.: Bertopic: Neural topic modeling with a class-based tf-idf procedure (2022)
9. Hoyle, A., Goel, P., Hian-Cheong, A., Peskov, D., Boyd-Graber, J., Resnik, P.: Is automated topic model evaluation broken? the incoherence of coherence. Advances in neural information processing systems **34**, 2018–2033 (2021)
10. Liu, N.F., Lin, K., Hewitt, J., Paranjape, A., Bevilacqua, M., Petroni, F., Liang, P.: Lost in the middle: How language models use long contexts. Transactions of the Association for Computational Linguistics **12**, 157–173 (2024). https://doi.org/10.1162/tacl_a_00638, https://aclanthology.org/2024.tacl-1.9
11. Muennighoff, N., Tazi, N., Magne, L., Reimers, N.: Mteb: Massive text embedding benchmark (2023), https://arxiv.org/abs/2210.07316
12. OpenAI: Gpt-4 technical report (2024)
13. Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C.L., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., Schulman, J., Hilton, J., Kelton, F., Miller, L., Simens, M., Askell, A., Welinder, P., Christiano, P., Leike, J., Lowe, R.: Training language models to follow instructions with human feedback (2022), https://arxiv.org/abs/2203.02155
14. Pham, C.M., Hoyle, A., Sun, S., Iyyer, M.: Topicgpt: A prompt-based topic modeling framework (2023)

15. Reimers, N., Gurevych, I.: Sentence-bert: Sentence embeddings using siamese bert-networks (2019), https://arxiv.org/abs/1908.10084

16. Reuter, A., Thielmann, A., Weisser, C., Fischer, S., Säfken, B.: Gptopic: Dynamic and interactive topic representations (2024), https://arxiv.org/abs/2403.03628

17. Valmeekam, K., Marquez, M., Olmo, A., Sreedharan, S., Kambhampati, S.: Planbench: An extensible benchmark for evaluating large language models on planning and reasoning about change (2023)

18. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need (2023), https://arxiv.org/abs/1706.03762

19. Wang, H., Prakash, N., Hoang, N.K., Hee, M.S., Naseem, U., Lee, R.K.W.: Prompting large language models for topic modeling (2023), https://arxiv.org/abs/2312.09693

20. Wu, X., Nguyen, T., Luu, A.T.: A survey on neural topic models: methods, applications, and challenges. Artificial Intelligence Review **57**(2), 18 (Jan 2024). https://doi.org/10.1007/s10462-023-10661-7, https://doi.org/10.1007/s10462-023-10661-7

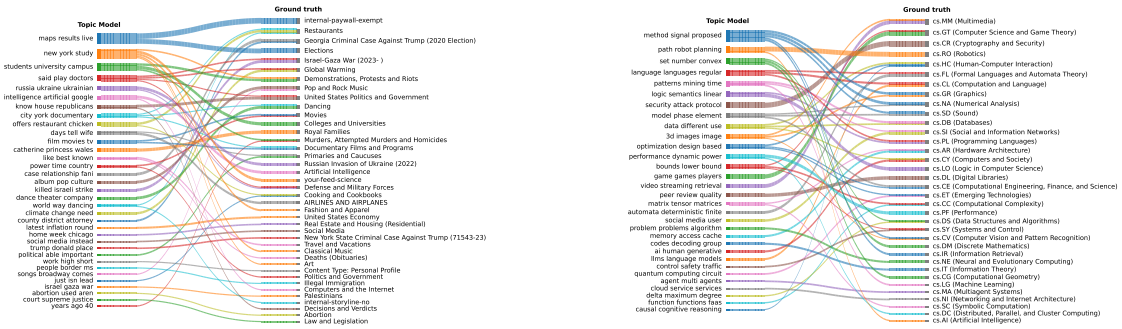
21. Xu, Z., Jain, S., Kankanhalli, M.: Hallucination is inevitable: An innate limitation of large language models (2024), https://arxiv.org/abs/2401.11817

A Additional Figures and Tables

(a) New York Times dataset ($N = 800$)

(b) Arxiv dataset ($N = 800$)

Fig. 5: Comparison of topics identified by BERTopic with the ground truth. We visualize the 55 most common (topic label, ground truth label) pairs for each dataset.



(a) New York Times dataset ($N = 800$) (b) Arxiv dataset ($N = 800$)

Fig. 6: Comparison of topics identified by NMF with the ground truth. We visualize the 55 most common (topic label, ground truth label) pairs for each dataset.

dataset	metric name model	V-measure	completeness	homogeneity
NYT	BERTopic	0.61 (0.02)	0.65 (0.018)	0.57 (0.022)
	LDA	0.5 (0.007)	0.51 (0.007)	0.48 (0.007)
	NMF	0.54 (0.009)	0.55 (0.008)	0.53 (0.01)
	TopicGen (4o-mini)	0.64 (0.015)	0.68 (0.015)	0.61 (0.016)
	TopicGen (GPT-4o)	0.65 (0.048)	0.68 (0.046)	0.61 (0.051)
	TopicGen (GPT-4o)*	**0.69**	**0.71**	**0.67**
20NEWSGROUPS	BERTopic	0.56 (0.037)	0.62 (0.034)	0.51 (0.039)
	LDA	0.20 (0.01)	0.20 (0.011)	0.19 (0.01)
	NMF	0.35 (0.03)	0.35 (0.029)	0.34 (0.03)
	TopicGen (4o-mini)	0.58 (0.029)	0.63 (0.027)	0.55 (0.033)
	TopicGen (GPT-4o)	**0.61** (0.019)	**0.64** (0.03)	**0.58** (0.01)
	TopicGen (GPT-4o)*	0.54	0.57	0.51
ARXIV	BERTopic	**0.57** (0.013)	**0.62**(0.013)	0.52 (0.015)
	LDA	0.41 (0.008)	0.42 (0.008)	0.40 (0.009)
	NMF	0.51 (0.016)	0.52 (0.017)	0.50 (0.016)
	TopicGen (4o-mini)	0.56 (0.026)	0.59 (0.024)	**0.53** (0.031)
	TopicGen (GPT-4o)	0.56 (0.017)	0.59 (0.013)	**0.53** (0.024)
	TopicGen (GPT-4o)*	**0.57**	**0.62**	0.52
PUBMED	BERTopic	**0.60** (0.022)	**0.67** (0.016)	**0.55** (0.027)
	LDA	0.56 (0.004)	0.60 (0.004)	0.52 (0.005)
	NMF	0.58 (0.012)	0.63 (0.01)	0.54 (0.013)
	TopicGen (4o-mini)	**0.60** (0.013)	0.65 (0.007)	**0.55** (0.017)
	TopicGen (GPT-4o)	0.59 (0.016)	0.66 (0.012)	0.54 (0.02)
	TopicGen (GPT-4o)*	0.59	0.65	**0.55**

Table 2: Table with results for $N = 400$. Standard deviations are provided between brackets.
* = iterative merging strategy applied

B Prompts Used

CANDIDATE_CREATION

Your task will be to distill a list of topics from the following documents:
DOCUMENT: {document 1}
DOCUMENT: {document 2}
...
DOCUMENT: {document N}
Your response should be a JSON in the following format: {'topics': ['topic1', 'topic2', 'topic3']}
Topics should not be too specific, but also not too general. For example, 'food' is too general, but 'lemon cake' is too specific. A topic does not need to be present in multiple documents.

TOPIC_REDUCTION

Your task will be too distill a list of core topics from the following topics:
0: {topic 1}
1: {topic 2}
...
NCT-1: {topic NCT} (where NCT = Number of topics)
Your response should be a JSON in the following format: "topics": ["topic1", "topic2", "topic3"]
Remove duplicate topics and merge topics that are too general. Merge topics together that are too specific. For example, 'food' might be too general, but 'lemon cake' might be too specific. In the end, try to arrive at a list of about NT topics.

TOPIC_REDUCTION_ITERATIVE
This is the prompt for the iterative strategy we initially attempted

Your task will be to merge a pair of topics out of the following topics:
0: {topic 1}
1: {topic 2}
...
NCT-1: {topic NCT} (where NCT = Number of topics)
Your response should be a JSON in the following format:
{'topic_pair': [idx1, idx2], 'new_topic': 'new_topic'}
The index should be the index of the topic in the list of topics. The new topic should be a combination of the two topics. Keep the name of the topic simple, try to generalize. So if you merge topic 'A' and 'B' together, do not name the topic something like 'A and B'. Rather, find the common more general denominator. In selecting the pair to merge, please merge the most similar, and most granular topics first.

TOPIC_ASSIGNMENT

Your task will be to classify the following document into one of the following topics:
DOCUMENT: {document}

0: {topic 1}
1: {topic 2}
...
NT-1: {topic NT}
Your response should be a JSON in the following format: {'topic': idx}
The index should be the index of the topic in the list of topics.