IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 10 November 2025, accepted 25 November 2025. Date of publication 00 xxxx 0000, date of current version 00 xxxx 0000.

Digital Object Identifier 10.1109/ACCESS.2025.3638956

A Comparative Analysis of Topic Reduction Techniques for BERTopic

WANNES JANSSENS, MATTHIAS BOGAERT, AND DIRK VAN DEN POEL, (Senior Member, IEEE)
Research Group Data Analytics, Faculty of Economics and Business Administration, Ghent University, 9000 Ghent, Belgium
FlandersMake@UGent–corelab CVAMO, 9000 Ghent, Belgium

Corresponding author: Matthias Bogaert (matthias.bogaert@ugent.be)

This work was supported by Flanders Make under Grant 180C0123.

ABSTRACT BERTopic is a state-of-the-art topic modeling framework that generates topics by clustering contextualized document embeddings. However, its default clustering algorithm, HDBSCAN, often generates an excessive number of topics, which hinders meaningful comparisons or applications in downstream tasks. While different topic reduction methods exist, literature lacks a comparison of these methods and their impact on the quality of the reduced topics. This study offers an in-depth exploration of 1) topic reduction methods for BERTopic, including existing approaches for direct or indirect reduction, and 2) novel techniques that leverage large language models (LLMs), either by using the LLM-generated topic labels to create topic embeddings or by directly prompting the model for identifying overlapping topics. A comparative study of these topic reduction methods is performed, evaluating their performance in terms of coherence and diversity across multiple social media and web datasets. Our findings indicate that indirect topic reduction (e.g., agglomerative clustering) yields more diverse topics, though sometimes at the cost of coherence, compared to direct topic reduction (increasing the minimum cluster size). LLM prompting outperforms agglomerative clustering in both coherence and diversity, though at a higher computational cost. Therefore, we recommend selecting a reduction method based on dataset characteristics, computational resources, and the desired balance between diversity and coherence.

INDEX TERMS BERTopic, HDBSCAN, large language models, natural language processing, topic modeling, topic reduction.

I. INTRODUCTION

BERTopic [1] is widely adopted by practitioners and researchers, due to its user-friendly modular approach and its proven ability to generate high-quality topics. Prior work has shown that it performs favorably compared to traditional topic modeling techniques [1], [2], [3], [4]. It is particularly popular for analyzing user-generated content (UGC) and short texts, as these often benefit from BERTopic's strengths in handling noisy, and sparse content [2]. The default structure involves four modules: (i) document embedding using Sentence-BERT (SBERT) [5], (ii) dimensionality reduction via UMAP (Uniform Manifold Approximation and Projection for Dimension Reduction) [6], (iii) clustering with HDBSCAN (Hierarchical Density-Based Spatial Clustering of Applications with Noise) [7], and (iv) a representation module combining CountVectorizer and class-based term frequency-inverse document frequency (c-TF-IDF) for the tokenizer and weighting steps, respectively. Each module involves distinct underlying algorithms and is governed by its own set of parameters, making BERTopic a complex framework with a large parameter space for tuning.

The HDBSCAN module is arguably the most critical component of the BERTopic framework. The module creates density-based hierarchical clusters from the reduced document embeddings, also identifying outliers (documents that are considered noise and not assigned to any cluster). HDBSCAN does not require specifying the number of clusters (topics) upfront, which often results in coherent but an excessive number of topics, especially when dealing with

The associate editor coordinating the review of this manuscript and approving it for publication was Giacomo Fiumara.

short-text and UGC. Consequently, topic reduction should be applied to obtain a manageable number of dimensions for downstream tasks (e.g. classification) or to enable fair comparisons with other topic modeling approaches, where an equal number of topics is essential.

A first gap in the literature is the lack of clarity and systematic comparison of BERTopic's built-in topic reduction methods. BERTopic includes built-in functionalities for topic reduction in a direct way (i.e., performed during the fitting process of the BERTopic model) and an indirect way (i.e., applied after the model has been fitted). These methods can, in turn, be applied on different types of topic embeddings. In addition, studies that employ BERTopic are often unclear about which reduction methods they use. Some rely on the default settings [1], [3], [4], others do not specify their approach at all [8], and others perform manual topic reduction [9], indicating that users have limited knowledge of existing reduction methods and how different techniques affect the quality of the reduced topics, leading to suboptimal use of the framework.

A second gap is the limited exploration of large language models (LLMs) for topic reduction within BERTopic. LLMs have recently gained popularity in topic modeling, either by enhancing the interpretation of complex topic representations [10], [11] or by enabling end-to-end topic modeling [12], [13], [14]. These end-to-end approaches often include a topic merging step, where overlapping topics are identified directly through LLM prompting. However, despite their growing relevance, the use of LLMs for topic reduction in BERTopic has received little attention.

To address these gaps, we first review various topic reduction techniques for BERTopic, including both existing methods (i.e., tuning the parameters of HDBSCAN and applying agglomerative clustering) and new approaches involving the use of LLMs (i.e., improving the topic embeddings and LLM prompting to join overlapping topics). Finally, we evaluate these techniques across three datasets (i.e., Yelp reviews, Trump and #Covid19 tweets), analyzing their impact on topic coherence and diversity.

Our key *contributions* are as follows:

- We present the first comprehensive comparative analysis of topic reduction methods for BERTopic, systematically evaluating various techniques based on coherence and diversity across multiple social media and web datasets and HDBSCAN parameter configurations.
- We categorize both established and newly proposed topic reduction techniques, provide insights into their mechanisms and effectiveness, and give practical guidelines for deciding which topic reduction methods to use.
- We analyze the different steps of the HDBSCAN algorithm and the impact of the *minimum cluster size* and *cluster selection method* parameters on the resulting topics, supported by empirical results illustrating effects on outlier creation, topic coherence and diversity.
- We propose using SBERT embeddings of LLM-generated topic labels as a representation for indirect topic reduction via agglomerative clustering and evaluate its effectiveness within BERTopic.
- We propose an indirect topic reduction technique for BERTopic by using LLM prompting to identify overlapping topics. To evaluate its effectiveness, we conduct an empirical analysis using the OpenAI GPT-4o-mini [15] model and Meta's open-source Llama-3-8B-instruct [16] model.

The aim of this study is twofold: (1) to provide BERTopic users with deeper insight into how the framework generates a specific set of topics, which reduction methods exist, how they work, and how they impact the quality of the reduced topics; and (2) to explore the potential of LLMs to support and enhance the topic reduction process, ultimately aiming to produce higher-quality topics. To support transparency and reproducibility, we have made all code and data publicly available at https://github.com/Wannes15/BERTopic_reduction

Section II covers related literature on topic modeling, the performance of BERTopic, topic reduction techniques, and the application of LLMs. Section III outlines both existing and newly proposed topic reduction techniques evaluated in this study. Section IV details the experimental setup, including dataset selection, BERTopic configuration, and evaluation metrics. Section V discusses obtained results. Section VI summarizes the findings and offers recommendations on selecting the most suitable reduction method. Finally, section VII covers limitations and further research.

II. RELATED WORK

A. APPLICATIONS OF TOPIC MODELING

Topic modeling is an unsupervised machine learning technique used to find underlying topics in a collection of documents (or corpus). It reveals the underlying semantic structure of unstructured texts by identifying patterns of word use and connecting documents that exhibit similar patterns [17], [18]. Traditionally, the primary focus of topic modeling was on the efficient analysis of large volumes of longer text, such as emails, scientific abstracts, and newspaper archives [17], [19]. However, the rise of customer engagement and experience platforms, such as social media, consumer review sites and peer-to-peer applications, has tremendously increased the amount of unstructured data and especially UGC. This has driven a growing need for advanced text analytical techniques [20], including topic modeling, to identify hidden patterns and relationships hidden within these messages [2].

A common application is the analysis of tweets where topic modeling can be used to identify broad topics of conversation or emerging themes [9], [21], and to conduct sentiment analysis on these topics [22]. Additionally, it enables the tracking of trending subjects and their evolution over time, providing valuable insights into how public discourse shifts in response to events or social dynamics (e.g Covid 19) [4], [23]. Other studies employ topic modeling to extract

features or dimensions from textual data, which can then be utilized in various downstream processes. For instance, topics extracted from Airbnb property descriptions have been used to predict demand [24], while topics from tweets have been integrated into online analytical processing [25]. Similarly, topics identified from company web pages have been leveraged to classify the innovativeness of firms [26].

In the following, the most common topic modeling methods are discussed in studies involving social media and web data as this is the primary focus of our research. For a comprehensive overview of different topic modeling approaches we refer the reader to [27]. Topic modeling approaches can be categorized into non-contextual and contextual methods. Non-contextual methods are based on a Bag-of-Words and include techniques such as Latent Dirichlet Allocation (LDA) and Non-Negative Matrix Factorization (NMF) [28]. In the case of LDA, a probabilistic model, each document in a collection is represented as a finite mixture of underlying topics, and each topic is represented as a distribution over an underlying set of word probabilities [29]. NMF, a non-probabilistic algorithm, uses matrix factorization to model each document as a mixture of hidden topics [2]. Despite showing a decent performance, traditional Bag-of-Words approaches do not take into account the order of words, thereby failing to capture semantic relationships among words and sentences [30]. Additionally, they frequently require extensive preprocessing steps such as stop-word removal and lemmatization to achieve satisfactory performance [31]. Consequently, the emergence of neural embedding techniques has led to criticism of these traditional methods in the literatures [1], [2], and [31].

Contextual methods employ embedding techniques to take into account the context of a word or sentence to create vector representations in such a way that similar texts/sentences are closer in vector space. This way, text embeddings can capture semantic relationships between words and documents [1]. Examples of topic models that use an embedding approach are Top2Vec [31] and BERTopic [1]. Top2Vec leverages Doc2Vec's [30] word- and document representations to jointly learn embedded topic, document, and word vectors. Similar documents are then clustered together and topic representations are created by finding words close to the cluster's centroid [31]. BERTopic leverages SBERT [5] embeddings and uses a class-based version of TF-IDF [32] to extract topic representations, overcoming the centroid-based assumption of Top2Vec [1].

B. BERTopic PERFORMANCE

BERTopic has been frequently adopted both in industry and academia. It has been used not only for uncovering trends and topics in data but also in numerous comparative studies. These studies have evaluated BERTopic against traditional topic modeling methods such as LDA and NMF, as well as more contemporary approaches such as Top2Vec. In the following, we examine the key strengths and limitations of BERTopic.

Firstly, Grootendorst [1] demonstrated BERTopic's competitive performance in generating coherent topics, comparing it with LDA, NMF, and CTM across benchmark datasets. Other studies [3], [4], [8] also confirmed BERTopic's superior topic coherence across various applications and configurations. However, topic models should not be evaluated solely on coherence [27]. Egger and Yu [2] conducted a qualitative comparison of BERTopic, LDA, NMF, and Top2Vec, emphasizing BERTopic's well-separated topics, its modular flexibility, and its use of HDBSCAN for hierarchical topic reduction. They also noted LDA's unsuitability for social media data. Additionally, BERTopic has shown advantages in clustering performance. Gan et al. [33] found it outperformed LDA and Top2Vec, improving clustering by at least 34.2% on Chinese and English datasets. Similarly, Turan et al. [34] found BERTopic produced more meaningful results on an Amazon review dataset compared to LDA and NMF. These findings further support BERTopic's effectiveness over traditional topic modeling methods especially on social media and web data, which motivates our decision to focus on the BERTopic framework for the comparative analysis presented in this paper.

Although BERTopic offers notable advantages, researchers have also pointed out some drawbacks of BERTopic. Egger and Yu [2] noted the inability of BERTopic to generate true topic distributions since each document is assigned to a single topic. Other studies also argued that outlier generation can, in some cases, be excessive for BERTopic and Top2Vec, which can make analysis difficult [2], [8]. Despite these concerns, Grootendorst [1] argued that outliers can actually improve topic quality by preventing the assignment of irrelevant documents. This view is supported by De Groot et al. [8], who found that BERTopic, when used with HDBSCAN, achieves higher topic coherence and diversity. Furthermore, BERTopic includes built-in methods for both outlier reduction and topic distribution approximation, although these features fall outside the scope of this study. Regarding the issue of single-topic assignment, this assumption is often reasonable for social media and web texts, where individual posts typically focus on a single topic.

Another notable property of BERTopic is that it does not require the number of topics to be specified in advance. This can be viewed as both an advantage and a disadvantage. On the one hand, BERTopic's clustering module can automatically identify a coherent set of topics, on the other hand, this can sometimes result in an undesirable number of topics. To set the number of topics to a fixed number and increase the overall interpretability of the topics, topic reduction is often applied. However, there is no clear consensus in the literature on which reduction method is most appropriate.

For instance, Grootendorst [1] and Abuzayed and Al-Khalifa [3] trained models across a range of topic numbers to compare results but did not explicitly describe their reduction

method for BERTopic. From context, they likely used the *nr_topics* parameter, which merges topics based on c-TF-IDF embeddings. Similarly, Ogunleye et al. [4] explicitly stated that they used the *nr_topics* parameter to reduce the number of topics in their BERTopic model with the HDBSCAN module from over 100 to 10.

In contrast, Egger and Yu [2] explicitly described merging topics based on the cosine distance matrix of the topic embeddings for their qualitative comparison. However, they did not clarify whether the embeddings used were neural embeddings or c-TF-IDF embeddings. In the case of de Groot et al. [8], the method used for topic reduction is entirely unclear. Meanwhile, Baird et al. [9] identified the optimal number of topics based on coherence and then manually reduced topics through a detailed procedure, not by merging but by simply dropping certain topics.

In summary, the specific reduction methods used in these studies are often not clearly reported. Some studies rely on the default methods, others do not specify their approach, whereas others perform manual topic reduction. This lack of clarity and methodological consistency suggests that many studies are likely unaware of the exact underlying processes and that insufficient attention is given to exploring different topic reduction methods. Therefore, it is essential to conduct a systematic comparison of the different topic reduction methods available in BERTopic, as their impact on topic coherence and diversity remains underexplored. Specifically, this study aims to (1) provide a structured overview of these reduction techniques and (2) perform a comparative analysis of their impact on coherence and diversity. This type of analysis is crucial for guiding practitioners and researchers in selecting the most appropriate topic reduction technique for their applications and is currently missing from the literature.

C. LLMs FOR TOPIC MODELING

More recently, studies have explored the potential of LLMs to enhance the topic modeling process in several ways, including topic evaluation and interpretation, as well as end-to-end topic generation.

Stammbach et al. [35] leverage LLMs to evaluate topic models by assessing the topic coherence through two key tasks: an intrusion detection task and a coherence rating task. In the intrusion detection task, the LLM is prompted to identify an ''intruder'' word from a list of top words associated with a given topic, aiming to gauge how well the top words represent the topic. In the coherence rating task, the LLM is asked to rate the coherence of the topic words on a scale of 1 to 3, providing a more direct evaluation of the quality of the topic. The study utilized both ChatGPT and the open-source FLAN-T5 model, concluding that LLMs can effectively and accurately assess topic coherence. Moreover, the research demonstrated that LLMs can assist in automatically determining the optimal number of topics for LDA and other bag-of-words-based topic modeling techniques.

Li et al. [11] used ChatGPT to improve topic representation by generating more interpretable labels for the top 10 most frequent words related to topics output by an LDA model and showed promising results. Their study reached the interesting conclusion that these generated labels are often preferred over human-generated labels. More advanced approaches could include in-context learning techniques, such as the inclusion of carefully crafted demonstrations and instructions in the prompt to create more accurate and interpretable topic representations. In related work, Koloski et al. [10] applied LLMs to generate topic labels that are tailored to specific domains, showing that prompts informed by domain expertise can yield labels that more closely reflect the context of the data. They further introduced a new evaluation metric and explored adapting BERTopic to particular domains by fine-tuning sentence transformer embeddings, though their study did not investigate tuning BERTopic parameters or applying post hoc topic reduction.

PromptTopic and TopicGPT are two recent LLM-based end-to-end approaches to topic modeling. PromptTopic [13] involves three stages: extracting high-level topics at the sentence level using carefully crafted prompts, iteratively merging similar topics to reduce redundancy, and refining topic representations by selecting the most representative words. Their study made use of ChatGPT and LLaMA-13B. TopicGPT [14] follows a two-stage process: generating topics from a document sample and assigning these topics to the full corpus. Similar to PromptTopic, it refines topics by merging similar ones using embedding similarities, but using GPT-3.5-Turbo and Mistral-7B-Instruct. Doi et al. [12] present another end-to-end topic modeling framework in which large language models are prompted either in parallel or sequentially to extract and merge topics, using GPT-3.5 and GPT-4. Similarly, Mu et al. [36] demonstrate that LLMs can be prompted to generate topics from a set of documents and refine them according to human guidelines, highlighting both the potential and limitations of LLM-based topic extraction. While these approaches demonstrate that LLMs can produce coherent topics, they come with notable drawbacks, including substantial computational overhead and long runtimes. Although effective at identifying broad, high-level topics, they often struggle to capture more detailed or nuanced themes, and their evaluations against BERTopic do not take into account the sensitivity of BERTopic to parameter settings. In contrast, our work explicitly addresses this by averaging results across multiple HDBSCAN configurations.

The above studies show the potential of LLMs to add value in the topic modeling process. None of these studies, however, have explored the potential of LLMs to assist in the topic reduction process for BERTopic, which we aim to investigate in this study. In particular, we employ LLMs in two ways: (1) to identify overlapping topics within the set of topics generated by BERTopic, and (2) to label topic representations prior to embedding them for use in the topic reduction process.

III. TOPIC REDUCTION TECHNIQUES

BERTopic generates a set of topics, which is often too large for downstream tasks or for comparison with other topic modeling techniques. To address this, topic reduction is applied to obtain a smaller number of topics. To our knowledge, no study has yet explored or compared topic reduction techniques for BERTopic. This section classifies and describes the topic reduction techniques analyzed in this study.

Topic reduction methods are classified into two categories: direct and indirect. Direct approaches are performed during the fitting process of a BERTopic model, specifically, the clustering step of the framework is altered to produce fewer clusters. Indirect approaches start with an initial set of clusters/topics and iteratively merge similar ones. Therefore, indirect approaches can also be applied to other embedding based topic modeling frameworks, such as Top2Vec.

Figure 1 summarizes the topic reduction techniques compared in this study with existing methods shown in gray and newly proposed methods highlighted in blue. The number of topics can be reduced directly by increasing the *minimum cluster size* parameter of HDBSCAN (BERTopic's clustering module). BERTopic also includes built-in functionalities for indirect topic reduction using agglomerative clustering, which can be based on either the topic's c-TF-IDF embedding or the average reduced SBERT embedding of the topic. We extend this analysis by introducing a new embedding option for agglomerative clustering, which utilizes the SBERT embeddings of the topic labels generated by an LLM or the top 10 keywords. Additionally, we apply an indirect reduction method for BERTopic that merges topics using prompts to an LLM. A detailed description of each topic reduction method and its implementation is discussed below.

A. INCREASING THE MINIMUM CLUSTER SIZE

Increasing the *minimum cluster size* parameter of the HDBSCAN module is considered a direct reduction technique since it directly influences the tree condensing process step of the HDBSCAN algorithm. As the *minimum cluster size* increases, fewer clusters are created and subsequently available for selection. Additionally, variations in this parameter influence the creation of outliers, which in turn affects topic representations. For a more thorough understanding of the HDBSCAN algorithm and its parameters we refer the reader to section IV.

It should be noted that this method does not allow for specifying the exact number of topics. As a result, achieving the desired number of clusters or topics may require trial and error when tuning the *minimum cluster size* parameter. Nevertheless, adjusting the *minimum cluster size* is a straightforward and commonly used method for topic reduction, often serving as the first approach in practical applications. For the purposes of this study, this method is used as a benchmark.

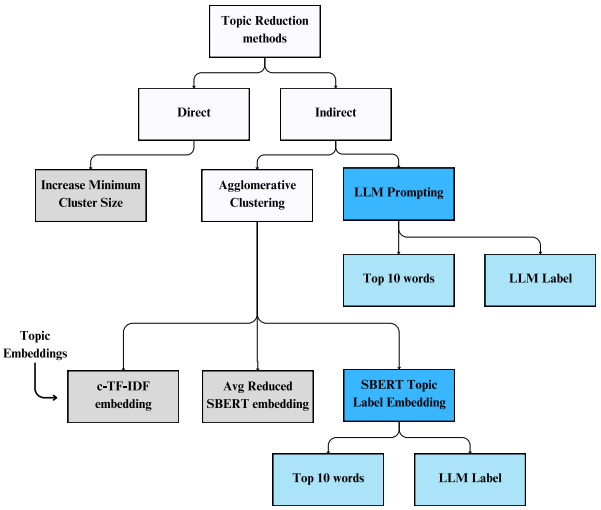

FIGURE 1. Topic reduction techniques compared in this study, with existing methods shown in gray and new methods shown in blue.

B. AGGLOMERATIVE CLUSTERING

In BERTopic, the number of topics can be reduced indirectly through a post-processing step involving agglomerative clustering. This method is considered indirect because it does not alter the topic modeling process. Instead, it is applied after the BERTopic model has already been fitted and the topics have been generated.

Agglomerative clustering starts from a set of clusters output by a BERTopic model and iteratively merges the two clusters that are closest to each other, based on the cosine average linkage distance between the topic embeddings. In other words, it calculates the average embedding of topics and merges the two nearest ones. The average embedding of these merged clusters is then used to represent the newly merged topic. This process continues until the desired number of topics is reached.

While the impact of various distance metrics and linkage methods could also be explored, we focus on comparing different types of topic embeddings since their averages are used for merging nearby clusters. Specifically, we consider the c-TF-IDF embedding, the average reduced SBERT embedding, and the newly proposed SBERT topic label embeddings, which are described in more detail below.

1) C-TF-IDF EMBEDDING

The c-TF-IDF embedding of a topic is generated by consolidating all documents assigned to that topic/cluster and then extracting the importance of all terms in the vocabulary using the following formula:

$$\text{c-TF-IDF}(t, c) = \text{TF}(t, c) \times \text{IDF}(t) \tag{1}$$

where:

$$\text{TF}(t, c) = \frac{f_{t,c}}{\sum_{t' \in c} f_{t',c}}$$

$$\text{IDF}(t) = \log\left(1 + \frac{A}{f_t}\right)$$

Here, t represents a term (word), c represents a cluster (topic), $f_{t,c}$ is the frequency of term t in cluster c, f_t is the frequency of term t across all clusters, and A is the average number of words per cluster.

The c-TF-IDF embedding is a bag-of-words approach to represent a topic. This same formula is also used to identify the most important words within a cluster (i.e., those with the highest c-TF-IDF score), which are then used to create the topic representation. As a result, topics with similar c-TF-IDF embeddings are expected to share similar important words.

2) AVERAGE REDUCED SBERT TOPIC EMBEDDING

The average reduced SBERT embedding for a topic is computed by taking the mean of the reduced SBERT embeddings of the documents assigned to that topic. This approach provides a semantic embedding that captures the overall meaning of the documents within the topic.

The BERTopic clustering algorithm, HDBSCAN, is density based, but averaging the embeddings of a cluster transforms the embedding into a centroid-based one. Fig. 2 illustrates this issue with two clusters that could have been found by the HDBSCAN module for two dimensions (note that in reality, typically 5 dimensions are used). The red cross represents the resulting average SBERT embedding of each cluster. For the left cluster, the embedding falls well within the cluster, indicating that it effectively captures the overall meaning of the topic. In the right cluster, the embedding lies outside the main cluster, which may negatively impact the topic reduction process when relying on this topic embedding.

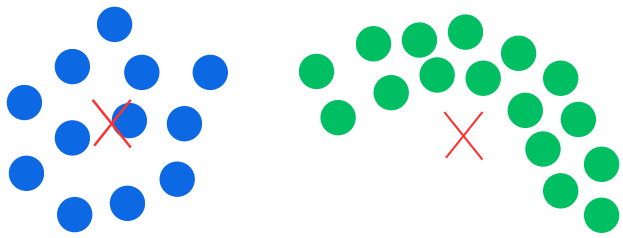

FIGURE 2. Illustration of the centroid-based assumption of the average reduced SBERT topic embedding. The red cross indicates the topic embedding for each cluster.

3) SBERT TOPIC LABEL EMBEDDING

Finally, we introduce a new type of topic embedding which starts from the topic representations. Each topic is either represented by their top 10 keywords, or by an LLM-generated label.

This topic representation (top 10 words or LLM label) is then embedded using the same SBERT model used in BERTopic's embedding module. By embedding the topic label we attempt (1) to overcome the centroid-based assumption that is present when using the average reduced SBERT topic embedding, and (2) to obtain a topic embedding that better represents the overall message of the documents assigned to this topic. A visual illustration of the SBERT topic label embedding is provided in Fig. 3.

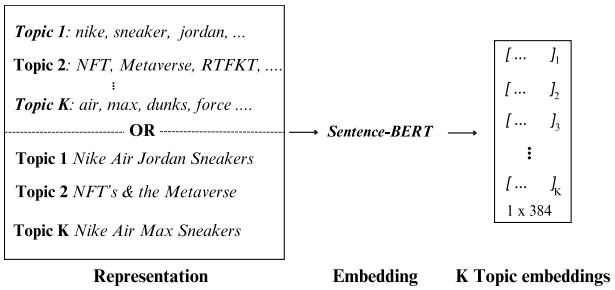

FIGURE 3. Illustration of the SBERT topic label embedding.

When using the top-10 words representation, the ten most representative words for each topic are selected based on their c-TF-IDF scores and encoded into a 384-dimensional vector using SBERT. For the LLM-label representation, a topic label is constructed by prompting an LLM (in our case, GPT-4o-mini) with the topic's keywords and four representative documents, as provided by BERTopic. An example of this prompt format is shown in table 1. The resulting label is then encoded into a vector using SBERT.

TABLE 1. Illustration of the LLM prompt for labelling topics based on their top 10 keywords and four representative documents from the Trump Tweets dataset.

Representation Prompt
I have a topic that contains the following documents: *1. 'Stock Market just hit another record high! Jobs looking very good.'* *2. 'Stock Market up big.'* *3. 'New and Historic Record. Job, jobs, jobs!'* *4. 'Record Stock Market & Jobs!'* The topic is described by the following keywords: *'recession', 'economy', 'jobs', 'market', 'unemployment', 'stock', 'economic', 'employment', 'gdp', 'business'* Based on the above information, can you provide a short label for the topic? **Answer:** 'Economic Growth and Stock Market Success'

C. LLM PROMPTING

Next to agglomerative clustering based on different types of topic embeddings, a novel indirect topic reduction method that leverages LLMs to merge overlapping topics is proposed. This approach involves representing each topic either by its top 10 keywords, extracted using c-TF-IDF, or by its LLM-generated topic label (using the prompt defined in table 1). The LLM is then prompted to identify the two most overlapping topics based on their representation. These topics are subsequently merged, and their representations are updated accordingly. This method is illustrated in Fig. 4.

In this study, we use OpenAI's GPT-4o-mini [15] and Meta's Llama-3-8B-instruct [16]. When prompted with

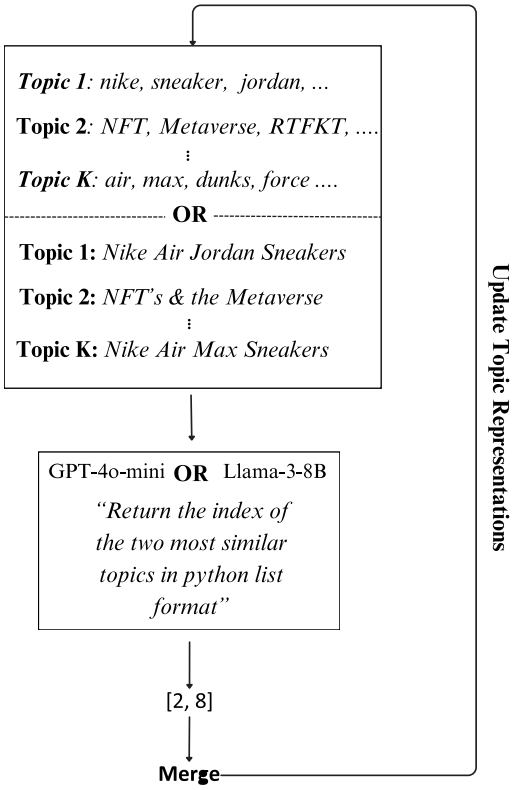

FIGURE 4. Illustration of topic reduction by prompting an LLM.

LLM-generated topic labels, Llama often declined to respond to sensitive content, such as defamation of political figures or conspiracy theories, particularly in the Trump Tweets and #Covid19 Tweets datasets. As a result, for the Llama model, we focused solely on the top 10 words when reducing topics through prompting.

Tables 2 and 3 present the full prompts used for identifying similar topics based on the topic's top 10 words and its LLM label respectively. To enhance the performance of the LLM, prompts were developed through a trial-and-error process informed by best practices in prompt engineering, incorporating a structured format with a system prompt, demonstration examples, and a clear instruction prompt to improve response robustness [37]. While an ablation study on prompt templates, temperature, and other hyperparameters could provide useful insights into variability, it is considered beyond the scope of this study.

IV. EXPERIMENTAL SETUP

In this section, we discuss our experimental setup for comparing the abovementioned topic reduction techniques. This includes a description of the datasets, the detailed settings of the BERTopic framework, including an interpretation of the HDBSCAN module for topic modeling, and the specific evaluation metrics employed for comparing the different reduction methods. Fig. 5 provides an overview of our experimental setup. All experiments, including BERTopic configurations, topic reduction procedures, and evaluation workflows, can be reproduced

TABLE 2. LLM prompt for merging based on top 10 words representation.

Prompt Section	Content
System Prompt	You are a helpful assistant tasked with identifying overlapping topics.
Instruction Prompt	Below I will provide a numbered list of topics. Each line represents the topic words of a topic. Return the index of the two most similar topics in Python list format. [TOPICS] Return only the Python list and nothing more!
Demonstration Prompt	Below I will provide a numbered list of topic words. Each line represents the topic words of a topic. Return the index of the two most similar topics in Python list format. • 0: astronaut, rocket, planet, galaxy, NASA, telescope, satellite, mission, orbit, cosmos • 1: apple, fruit, red, sweet, tree, orchard, juicy, vitamin, snack, fresh • 2: computer, laptop, technology, device, screen, keyboard, processor, portable, battery, software • 3: orange, fruit, citrus, vitamin, juicy, tropical, snack, fresh, tree, peel • 4: car, vehicle, engine, road, drive, wheels, speed, travel, gasoline, transportation • 5: book, library, read, pages, author, fiction, genre, chapter, knowledge, cover Return only the Python list and nothing more!
Demonstration Answer	`[1, 3]`

TABLE 3. Demonstration prompt for merging based on labelled topics.

Prompt Section	Content
Instruction Prompt	Below I will provide a numbered list of topics. Each line represents the topic words of a topic. Return the index of the two most similar topics in Python list format. [TOPICS] Return only the Python list and nothing more!
Demonstration Prompt	Below I will provide a numbered list of topics. Each line represents a topic. Return the index of the two most similar topics in Python list format. • 1: Space Exploration • 2: Artificial Intelligence in Industry • 3: Ocean Conservation • 4: Medieval History • 5: AI applied in Industry Return only the Python list and nothing more!
Demonstration Answer	`[2, 5]`

using the code and data provided in our public repository: https://github.com/Wannes15/BERTopic_reduction

A. DATASETS

The different topic reduction methods are benchmarked on three commonly used open-source datasets: Trump Tweets,[1] #Covid19 Tweets,[2] and Yelp reviews.[3] The Trump Tweets dataset consists of 43,352 tweets scraped from the @realDonaldTrump account of former United States President Donald Trump, spanning the period from May 2009 to June 2020. The #Covid19 Tweets dataset contains 178,683 tweets from June 2020 to August 2020, all featuring the hashtag *Covid19*. The average length of these tweets is comparable to those in the Trump dataset. Finally, the Yelp reviews dataset includes 10,000 reviews from April 2005 to May 2013, covering 4,171 companies. These reviews are significantly longer, with an average length approximately six times that of the tweets in the other datasets. By including both tweets and reviews, we combine short-text with longer-text UGC, offering a diverse range of input for analysis. The datasets are not preprocessed to ensure fair comparisons and preserve as much contextual information as possible for the SBERT embedding module within the BERTopic framework. Table 4 summarizes the main characteristics of each dataset.

TABLE 4. Dataset size and average document length.

Dataset	Size (#docs)	Avg #Words	Avg #Characters
Trump Tweets	43,352	20.76	131.53
#Covid19 Tweets	179,108	17.61	130.52
Yelp reviews	10,000	131.04	710.74

B. BERTopic FRAMEWORK SETUP

The BERTopic framework consists of four sequential modules: an embedding module, a dimensionality reduction module, a clustering module, and a representation module. While BERTopic allows flexibility in selecting alternatives for each module, the default configurations are used due to its strong performance and widespread adoption in practice. The default setup includes the pre-trained *"all-MiniLM-L6-v2"* model from SBERT for embedding generation, UMAP for dimensionality reduction, HDBSCAN for clustering, and a representation module combining CountVectorizer with c-TF-IDF, as illustrated in Fig. 6.

1) EMBEDDING

SBERT is a modification of the BERT model [38] designed for efficient sentence embeddings. Unlike traditional BERT, which requires comparing sentence pairs individually, SBERT uses a siamese or triplet network structure with a pooling layer to generate fixed-size vector representations of sentences. This enables fast and accurate semantic similarity comparisons [5].

The pre-trained *"all-MiniLM-L6-v2"* model is used to transform each input document into 384-dimensional vector representations where semantically similar documents will be 'close' in the high-dimensional vector space. While alternative embedding models are possible, the *"all-MiniLM-L6-v2"* is chosen for its good balance between speed and quality.

2) DIMENSIONALITY REDUCTION

After embedding the input documents, UMAP reduces the dimensions of each embedding to five to simplify the clustering process for HDBSCAN. UMAP is a non-linear dimensionality reduction algorithm that learns the manifold structure of high-dimensional data and creates a lower-dimensional embedding that preserves its essential topological structure. Although other dimensionality reduction techniques are available, UMAP is particularly effective at capturing both local and global structures in high-dimensional spaces and transferring them to lower-dimensional representations [39]. The candidate settings for UMAP are given in table 5. These hyperparameters remain unchanged during our experiments.

TABLE 5. UMAP candidate settings.

Hyperparameters	Value
n_neighbors	15
n_components	5
min_dist	0.0
metric	'cosine'
random_state	42

3) CLUSTERING WITH HDBSCAN

In the clustering step, we employ HDBSCAN to extract density-based hierarchical clusters from the reduced document embeddings, including the identification of outliers (documents classified as noise and not assigned to any cluster).

Rather than using the default configuration of HDBSCAN, we vary the *minimum cluster size* and *cluster selection method*, which significantly affect the clustering process. To understand exactly how these hyperparameters affect the clustering process, the pseudocode of HDBSCAN[4] is provided in Algorithm 1.

The algorithm starts from the reduced document embeddings and calculates the pairwise mutual reachability distance between all embeddings. The mutual reachability distance will "push" dense points further from less dense points. Using this distance matrix, a minimum spanning tree is then constructed via Prim's algorithm, which starts from a random node and iteratively connects the shortest edge not

[1] https://www.kaggle.com/datasets/austinreese/trump-tweets
[2] https://www.kaggle.com/datasets/gpreda/covid19-tweets
[3] https://www.kaggle.com/datasets/omkarsabnis/yelp-reviews-dataset
[4] https://hdbscan.readthedocs.io/en/latest/how_hdbscan_works.html

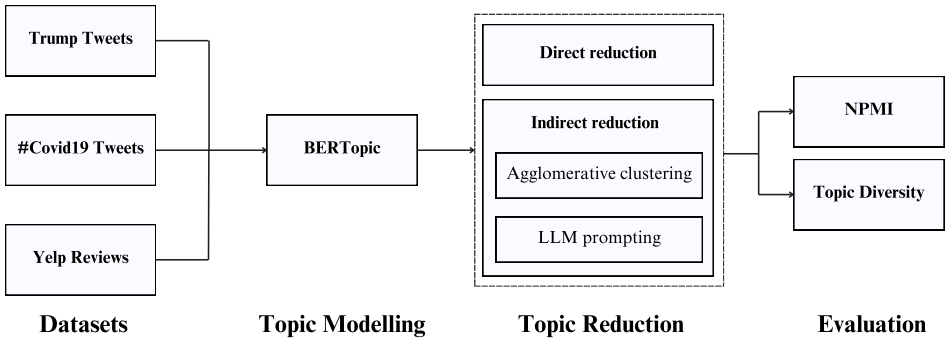

FIGURE 5. Overview of the experimental setup for comparing topic reduction techniques for BERTopic.

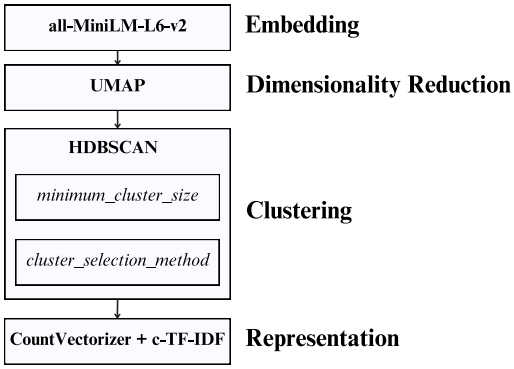

FIGURE 6. The BERTopic framework shown with the specific chosen modules.

Algorithm 1 HDBSCAN Clustering Algorithm

1: Compute the Mutual Reachability Distance (MRD) matrix between all reduced document embeddings.
2: Construct the Minimum Spanning Tree (MST) from the MRD graph.
3: Perform hierarchical clustering by sorting and iteratively removing edges from the MST.
4: Condense the tree by retaining only the splits where remaining points are larger than *minimum cluster size*.
5: Assign noise points (outliers) to label -1.
6: Select either all leaf clusters or apply stability-based selection.
7: **return** Final clusters and outliers.

yet included in the tree. Edges are subsequently removed from the tree one by one, starting with the largest edge, causing points or clusters to split off from the rest. This process creates a hierarchical tree structure. However, many edge removals may result in only a few points or singletons splitting off, resembling a decaying cluster rather than a meaningful split. To address this, the tree is condensed by retaining only clusters where the number of remaining points exceeds the *minimum cluster size* parameter. Points that fail to meet this criterion after a split are classified as outliers.

The resulting condensed tree is typically deep and contains numerous nested clusters. To select clusters from this structure, HDBSCAN offers two cluster selection methods (defined by the *cluster selection method* parameter): "excess of mass" (eom) and "leaf". The "leaf" method selects all leaf clusters, while the "eom" method moves up the tree, selecting a parent cluster when it is deemed more "stable" than its child clusters. Cluster stability, in this context, reflects how persistent a cluster is or how "slowly" it decays. It is worth noting that the "leaf" selection method typically results in more outliers, as it captures more cluster splits, which naturally include newly created outliers. For this study, we operate under the assumption that our primary goal is not to assign every document to a topic but rather to identify underlying trends. Therefore, the presence of outliers is acceptable in our context if it enhances the coherence and diversity of the topics discovered.

4) TOPIC REPRESENTATION

After the reduced document embeddings have been clustered, a topic representation is generated for each cluster using CountVectorizer for corpus tokenization and a class-based TF-IDF procedure to determine the importance of all words in a cluster, using this score, the top 10 most important words for each cluster/topic are typically used for representing the topic. The c-TF-IDF score is calculated as described in equation 1.

C. EVALUATION

For a fair and objective evaluation of the topic reduction techniques and due to the scale of our experiments, we chose to rely on automated metrics. More specifically, we compare each reduction technique based on the Normalised Pointwise Mutual Information (NPMI) coherence and topic diversity scores. This way, we achieve both a local and global evaluation of the reduced topics. In addition, we do this across multiple configurations of BERTopic, each time varying the *minimum cluster size* and the *cluster selection method* parameters.

1) COHERENCE

Several coherence metrics exist, including Cv Coherence, UMass Coherence, and NPMI Coherence [2], [3], [4]. All these metrics are quite similar, as they involve taking the set of top N words of a topic and summing a confirmation measure over all word pairs. The greatest correlation with human topic coherence ratings was observed when using the NPMI metric [40]. The formula for this metric is given in equation 2.

$$\text{NPMI}(w_i, w_j) = \frac{\ln\left(\frac{P(w_i,w_j)}{P(w_i)\cdot P(w_j)}\right)}{-\ln(P(w_i, w_j))} \tag{2}$$

The NPMI implementation [41] uses a sliding window of ten words to calculate $P(w_i, w_j)$ (probability of w_i and w_j occurring together in the corpus). In our case we chose to focus on the top 10 words for each topic. The values of NPMI range from $[-1, 1]$, with 1 indicating perfect co-occurrence, 0 representing independence, and -1 signifying that two words never occur together. This range makes NPMI a highly interpretable measure for topic coherence. The measure is then averaged across all word pairs within each topic and subsequently across all topics, resulting in an overall topic coherence score for the reduced topics.

2) DIVERSITY

The topic diversity is found by calculating the number of unique words in the top m across all topics and dividing it by the total number of topic words. The formula from the OCTIS library is used [42] and is given in equation 3 where m represents the top m (10 in our case) words from each topic. A higher topic diversity indicates better separation of different topics.

$$TD = \frac{|\text{unique words}|}{m \times |\text{topics}|} \tag{3}$$

V. RESULTS AND DISCUSSION

This section presents the results of our analysis and is structured as follows. First, we examine the direct topic reduction method by analyzing the effects of different HDBSCAN parameter settings on the resulting topics. Next, we compare this benchmark with indirect topic reduction methods via agglomerative clustering (using the different topic embeddings). Finally, we discuss the results of LLM-prompting and compare this method to both the benchmark and agglomerative clustering.

A. INCREASING THE MINIMUM CLUSTER SIZE

Before comparing different indirect topic reduction methods, we first illustrate key properties of the BERTopic framework. Specifically, we examine the impact of the *minimum cluster size* and *cluster selection method* in HDBSCAN on the number of topics, NPMI coherence, topic diversity, and the number of outliers. Understanding this impact is crucial, as other reduction techniques build on topics generated by an initial BERTopic model. The choice of initial hyperparameters can therefore significantly influence the reduction process.

Fig. 7 illustrates this impact for the Trump dataset, where we started from a *minimum cluster size* of 50, and increased this value to 400 in steps of 50 while alternating between the eom and leaf *cluster selection method*. The plot shows the number of outliers as well as the NPMI and topic diversity scores compared to the number of topics. Moving right on the graphs corresponds to decreasing *minimum cluster size*, leading to more topics. Each cross represents a BERTopic model with a specific *minimum cluster size* and *cluster selection method*.

The left graph shows that the leaf *cluster selection method* results in significantly more outliers compared to the eom *cluster selection method*, which is to be expected since taking the leaf clusters will result in more cluster splits in HDBSCAN's algorithm and more outliers. This implies that more documents are excluded from clusters which impacts the calculation of c-TF-IDF and the resulting topic words. Genuine outliers can improve topic representation by reducing noise, but excessive exclusion of documents may negatively influence coherence and diversity, as coherence is calculated across the entire corpus, including these outliers.

The middle graph shows no clear monotonic relationship between the number of topics and NPMI coherence, indicating the need for balance in setting the *minimum cluster size*. Additionally, the plot shows no consistent difference in coherence between eom and leaf selection. It should be noted that similar relationships are observed between the number of outliers and NPMI coherence for both the #Covid19 and Yelp datasets. To maintain clarity and avoid clutter, the results are not included, but available upon request.

Lastly, the relationship between topic diversity and the number of topics is also not very straightforward. While more topics intuitively might become less diverse, for small numbers of topics, this relation is less apparent and differs depending on the *cluster selection method*. For short-text datasets, such as Trump and #Covid19, we found that eom cluster selection often results in more diverse topics. This pattern aligns with intuition but is not universally true. For Yelp reviews, the relationship we saw is less consistent: eom selection yields better coherence at lower minimum cluster sizes, while leaf selection performs better at higher minimum cluster sizes. These variations underscore the dependency on the initial BERTopic configuration.

To ensure robustness against the influence of the initial parameter settings, we run multiple configurations for each dataset, apply the topic reduction techniques, and report the average coherence and diversity across runs. For the Trump dataset, this involved six different configurations: three different cluster sizes (50, 100 and 150), each evaluated with both the eom and leaf cluster selection methods. The #Covid-19 dataset used six configurations as well (with cluster sizes 100, 200 and 300). For the Yelp reviews dataset, we used eight configurations: four different minimum cluster

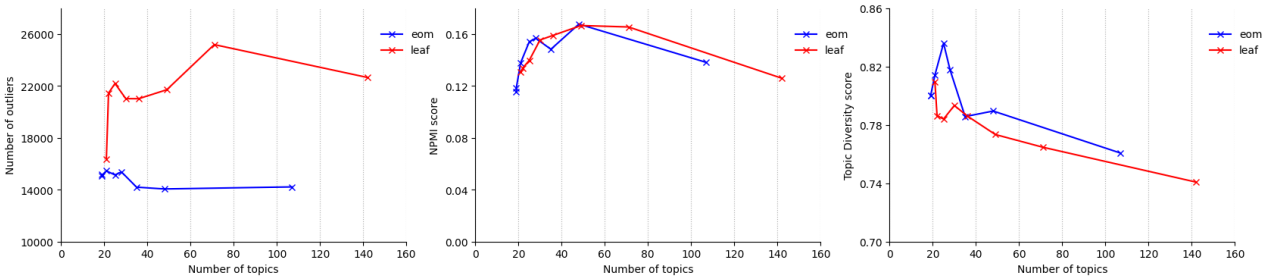

FIGURE 7. Number of outlier documents, NPMI coherence and topic diversity compared to the number of topics for the benchmark topic reduction method for the Trump Tweets dataset. Each cross indicates an increase of 50 of the *minimum cluster size* parameter in HDBSCAN. The line colour indicates the *cluster selection method* which can be 'excess of mass' (eom) or the leaf cluster selection method.

sizes (10, 20, 30 and 40), again with both cluster selection methods.

To compare increasing the *minimum cluster size* with other topic reduction techniques, two predefined topic counts are chosen for each dataset: 50 and 25 for the Trump dataset, 100 and 50 for the #Covid19 dataset, and 40 and 20 for the Yelp reviews dataset. To achieve these target quantities, we adjusted the *minimum cluster size* until the desired number of topics was approximately reached (within a range of $\pm 10\%$), also alternating between the eom and leaf cluster selection methods.

B. AGGLOMERATIVE CLUSTERING

First, we compare the benchmark of increasing the *minimum cluster size* to agglomerative clustering, which encompasses four different types of topic embeddings (Section III). Figs. 8 and 9 present the NPMI coherence and topic diversity scores, respectively, for each dataset and reduction method with the line colouring representing the reduced topic count. Within each dataset, the topic count differs and only lines of the same colour should thus be compared, as they correspond to reductions to the same number of topics. The NPMI coherence and topic diversity values for the reduced topics are averaged across various runs involving different fixed HDBSCAN parameter settings, as discussed previously. To illustrate stability across configurations, standard deviations are provided in parentheses. Some methods display a higher standard deviation than others, suggesting that their performance is more sensitive to the initial BERTopic configuration.

For the Trump dataset, the benchmark method achieves the highest coherence compared to agglomerative clustering with different embeddings. When reducing to 50 topics, coherence is 0.025 higher than with the c-TF-IDF and SBERT LLM Label embeddings. A similar but smaller gap is observed when reducing to 25 topics, with c-TF-IDF embedding performing best again, followed by the SBERT LLM label embedding. The average Reduced SBERT embedding and SBERT Top 10 Words embedding result in lower coherence, with the average reduced SBERT embedding performing the worst. A similar pattern emerges in the Yelp Reviews dataset, where the coherence gap is slightly larger, and the SBERT top 10 Words embedding performs the poorest. For the #Covid19 dataset, the c-TF-IDF, the average reduced SBERT, and the SBERT LLM Label embedding slightly outperform the benchmark in coherence when reducing to 100 topics. When further reducing to 50 topics, only the c-TF-IDF and SBERT LLM Label embeddings continue to show higher coherence, suggesting that topic reduction performance through agglomerative clustering also depends on the dataset.

When examining topic diversity, a different trend emerges. Agglomerative clustering results in significantly higher diversity than the benchmark method. The c-TF-IDF and SBERT LLM Label embeddings yield the most diverse topics, with diversity scores approximately 0.10 higher when reducing to 50 topics and 0.06 higher when reducing to 25 topics in the Trump dataset. The same holds true for the #Covid19 tweets and Yelp Reviews datasets, where the largest increases in diversity are also observed when using the c-TF-IDF or SBERT LLM label embedding. While the average reduced SBERT embedding and SBERT Top 10 words embedding also produce more diverse topics than the benchmark for the Trump tweets, they do not surpass c-TF-IDF or SBERT LLM Label embeddings. For the other datasets, the average reduced SBERT embedding struggles to achieve a higher diversity score than the benchmark. One possible explanation for the poor performance of the average reduced SBERT embedding is that it relies on a centroid-based representation, which is less suited for density-based clusters. Additionally, the worse performance of the SBERT Top 10 Words embedding compared to the SBERT LLM Label embedding can be explained by the fact that it does not capture contextual information, as the top 10 words are not structured into a sentence. The LLM-generated label, on the other hand, provides a more meaningful representation.

Overall, reducing topics through agglomerative clustering leads to more diverse topics compared to the benchmark method, while remaining competitive in terms of coherence. The increase in diversity appears to be more significant than the decrease in coherence, but the choice between coherence and diversity depends on the specific application.

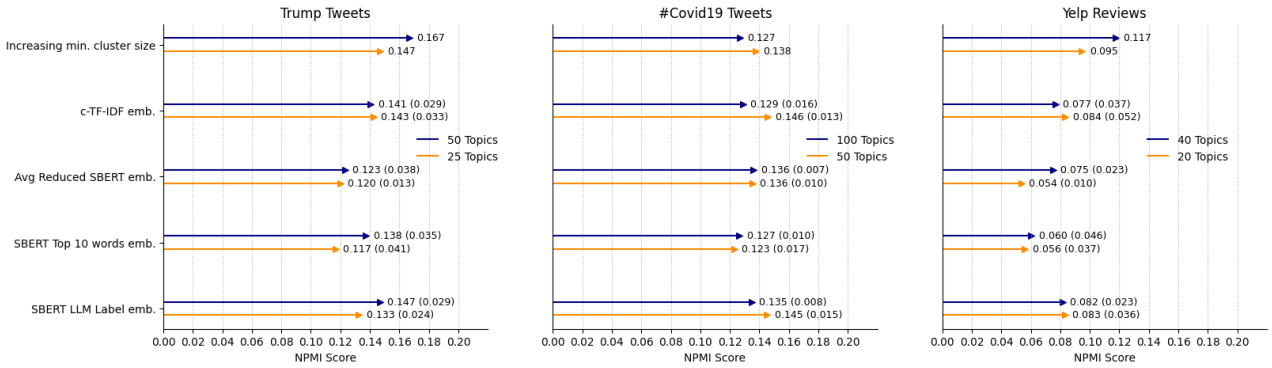

FIGURE 8. Values represent mean NPMI coherence scores (standard deviation across six HDBSCAN configurations) of the agglomerative clustering reduction method when reducing to two predefined topic amounts. Different topic embeddings are compared with the benchmark (increasing min. cluster size) across datasets. Increasing the minimum cluster size performs strongly in coherence, closely followed by agglomerative clustering with c-TF-IDF topic embeddings.

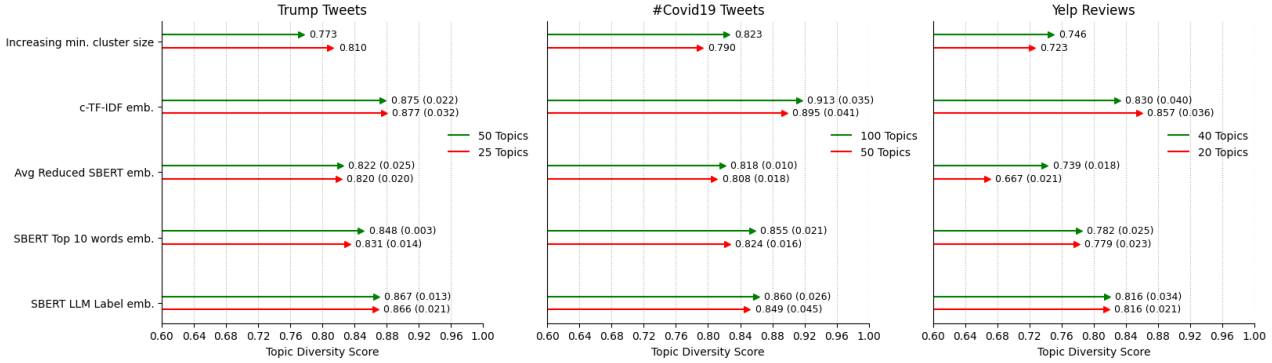

FIGURE 9. Values represent mean topic diversity scores (standard deviation across six HDBSCAN configurations) of the agglomerative clustering reduction method when reducing to two predefined topic amounts. Different topic embeddings are compared with the benchmark (increasing min. cluster size) across datasets. The c-TF-IDF embedding consistently achieves the highest diversity.

Among the embeddings, c-TF-IDF emerges as the preferred option, with the SBERT LLM Label embedding as a viable alternative, though the latter is more computationally intensive without offering substantial improvements over c-TF-IDF. Given its simplicity and strong performance, c-TF-IDF is recommended for agglomerative clustering. The average reduced SBERT embedding and SBERT Top 10 words embedding are less favourable, as they perform poorly in both coherence and diversity compared to the other topic embeddings.

C. LLM PROMPTING

Finally, we describe the results obtained using the LLM prompting method, as described in Section III. It should be noted that for GPT prompting, we use either the topic's top 10 words or the LLM-generated topic label. For Llama prompting, we only used the top 10 words, as Llama often refused to generate meaningful results when using the LLM-generated label. Figs. 10 and 11 present the coherence and diversity scores across the three datasets when reducing to two predefined topic counts. As before, only lines of the same colour should be compared within each dataset, as they correspond to the same reduced topic count.

For the Trump dataset, when reducing to 50 topics, LLM prompting struggles to surpass the benchmark in coherence, though the gap is smaller than with agglomerative clustering. However, when reducing to 25 topics with GPT prompting using either the top 10 words or Llama, coherence improves significantly. Among the LLM-based methods, GPT prompting with the top 10 words achieves the highest overall coherence for this dataset.

For the #Covid19 dataset, GPT prompting (whether using the top 10 words or the LLM-generated label) results in a higher coherence than the benchmark for both 100 and 50 topics. Llama also slightly improves coherence when reducing to 100 topics, but when reducing to 50, it no longer outperforms the benchmark. Overall, GPT prompting provides the highest coherence gains for this dataset. For the Yelp reviews dataset, none of the LLM prompting methods improve coherence compared to the benchmark. However, when comparing LLM prompting methods among themselves, GPT still slightly outperforms Llama.

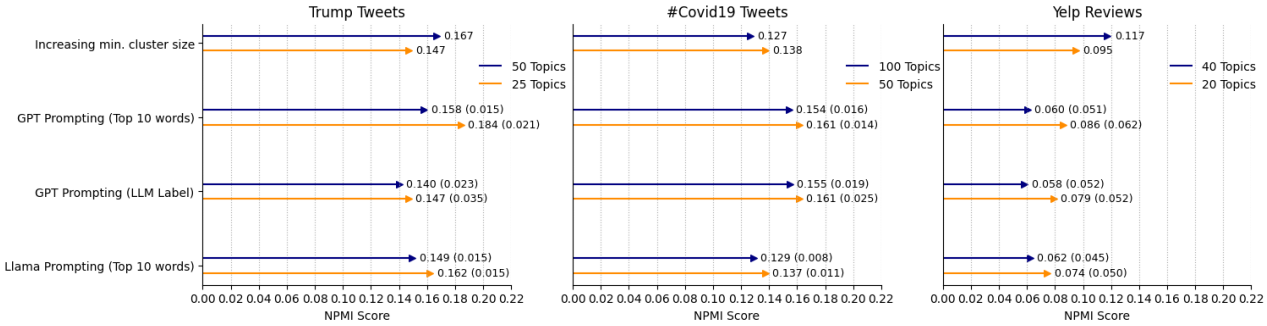

FIGURE 10. Values represent mean NPMI coherence scores (standard deviation across six HDBSCAN configurations) of the LLM prompting method when reducing to two predefined topic amounts, compared with the benchmark (increasing min. cluster size) across datasets. The LLM prompting method shows competitive performance to the benchmark, with some dataset-specific variation and more pronounced differences between the two reduction amounts.

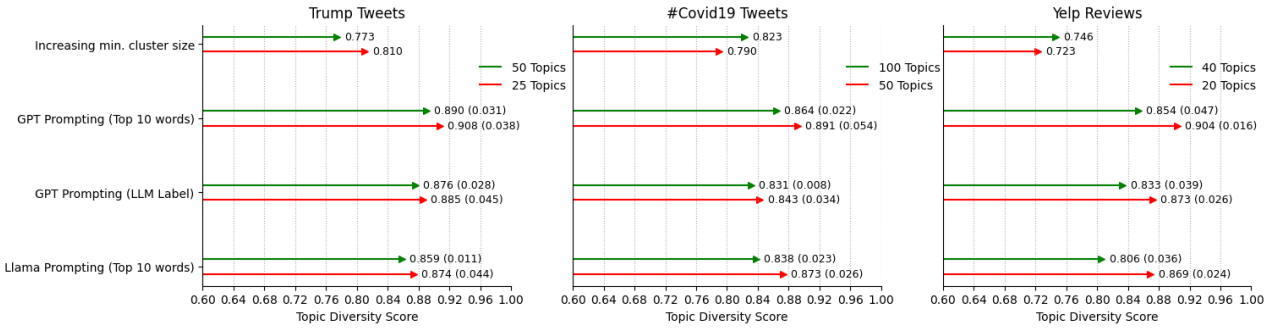

FIGURE 11. Values represent mean topic diversity scores (standard deviation across six HDBSCAN configurations) of the LLM prompting method when reducing to two predefined topic amounts, compared with the benchmark (increasing min. cluster size) across datasets. The prompting method consistently outperforms the benchmark, with GPT prompting (top-10 words representation) showing the strongest performance.

Regarding topic diversity, LLM prompting significantly enhances diversity compared to the benchmark across all datasets. GPT prompting with the topic's top 10 words consistently achieves the highest diversity. The ranking between GPT prompting with the LLM-generated label and Llama prompting with the top 10 words varies depending on the dataset and the number of topics. Additionally, for the Trump and Yelp reviews datasets, LLM prompting yields higher maximum diversity than agglomerative clustering, though for the #Covid19 dataset, it falls just short.

One possible explanation for these results is that GPT prompting with the topic's top 10 words captures more information than the LLM-generated label. It is possible that the LLM inherently performs an intermediate step of generating a label and comparing it with other topics, making the explicit LLM label generation redundant. Additionally, while Llama performs well, it is often slightly outperformed by GPT. This may be due to differences in model size and capabilities (GPT-4o-mini is generally considered more advanced than Llama-3 8B-Instruct), which could explain its stronger performance.

In summary, when reducing topics using LLM prompting, GPT-4o-mini with the topic's top 10 words is preferred and LLM prompting outperforms agglomerative clustering in both coherence and diversity. However, this approach is more computationally intensive, as it requires repeated LLM calls, which may be costly depending on the application. Additionally, c-TF-IDF must still be computed beforehand, as it is necessary for generating the top 10 words used in LLM prompting. When comparing short and long texts, Yelp reviews are generally longer than tweets, which may enhance transformer embeddings and, consequently, improve HDBSCAN's clustering quality. This could explain why the

TABLE 6. Runtime and token usage for different topic reduction methods on a toy example from the Trump Tweets dataset.

Method	Runtime	LLM Token Usage
Increasing minimum cluster size	–	–
Agglomerative Clustering		
c-TF-IDF embedding	1.6504 s	–
Avg. Reduced SBERT embedding	1.6067 s	–
SBERT Top-10 words embedding	1.7991 s	–
SBERT LLM Label embedding	32.8718 s	8,556
LLM prompting		
Top-10 words representation	1 min 6 s	28,406
LLM label representation	12 min 45 s	123,197

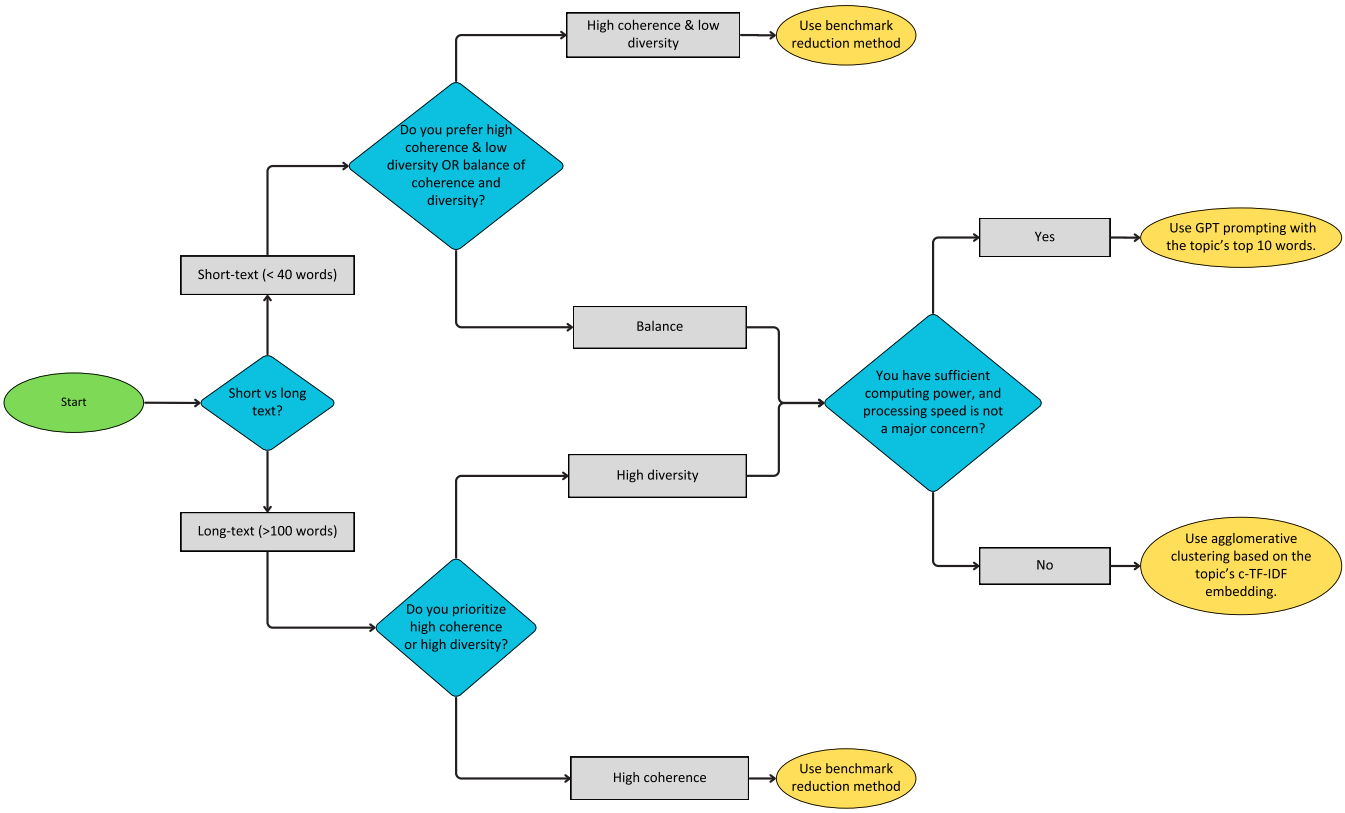

FIGURE 12. Flowchart for selecting a topic reduction method based on data characteristics, performance priorities, and resource constraints. Qualitative decision points such as "high coherence" and "high diversity" represent their relative importance within the context of specific tasks and user preferences, without fixed thresholds. This flowchart is intended as general guidance rather than a strict decision rule.

benchmark method performs relatively well for Yelp reviews, as the initial clustering step yields more coherent topics than other reduction methods. However, when topic diversity is a primary concern, alternative methods remain preferable.

D. RUNTIME AND TOKEN USAGE

This subsection compares the computational costs of the evaluated topic reduction methods. Both runtime and token usage depend strongly on the number of topics present before applying reduction. To illustrate relative differences, we report a simple example of reducing 50 topics to 25 topics from the Trump Tweets dataset, with runtimes and token usage summarized in Table 6. All LLM-based reductions were performed via the OpenAI API. Experiments were conducted on a laptop equipped with an NVIDIA RTX 4050 GPU, an Intel(R) Core(TM) Ultra 7 CPU, and 32 GB RAM.

Since increasing the minimum cluster size represents a direct reduction method, no separate runtime is reported, as it requires re-running the entire BERTopic model rather than an additional reduction step. Among the indirect methods, agglomerative clustering based on the c-TF-IDF, average reduced SBERT, or SBERT Top-10 words embeddings is very fast because these embeddings are already computed during the initial BERTopic run. Using the SBERT LLM label embedding introduces a clear increase in runtime, as each topic must first be labeled by an LLM before being embedded with SBERT. The LLM prompting methods are substantially slower and incur additional token costs. When prompting with the top 10 words, token usage is limited to identifying overlapping topics. However, prompting with LLM-generated topic labels requires an additional labeling step, resulting in significantly higher runtimes and token consumption.

VI. CONCLUSION AND RECOMMENDATIONS

This study is the first to categorize and compare various topic reduction techniques for the BERTopic framework. By revealing some of the ''black box'' aspects of BERTopic and exploring how different initial configurations and topic reduction techniques affect the quality of reduced topics, we allow users to make more informed decisions when using the framework, enhancing both the usability and effectiveness of BERTopic.

Our findings show that direct topic reduction by increasing the minimum cluster size in HDBSCAN often results in coherent topics. However, indirect reduction methods offer greater diversity and diversity can often be significantly improved with little or no loss of coherence, particularly in short-text datasets. For longer texts, such as the Yelp review, the direct method yields a high coherence score, while indirect approaches show a notable drop in coherence but substantially improve diversity. This suggests that for long

texts, there could be a trade-off between high diversity and high coherence when reducing topics. Greater topic diversity is especially valuable in scenarios where capturing a wider range of perspectives is desired.

For indirect topic reduction with agglomerative clustering, c-TF-IDF is the most effective topic embedding, offering strong performance with minimal computational cost. While the SBERT LLM label embedding is a potential alternative, it is more resource-intensive (as it requires an LLM for labelling topics) without significant improvements. The average reduced SBERT and SBERT Top 10 words embeddings are less favorable, as they in most cases perform poorly in both coherence and diversity compared to the other topic embeddings.

For indirect topic reduction via LLM prompting, using GPT-4o-mini with the top 10 topic words of each topic appears to be the most practical and effective method for short-text datasets. It achieves an excellent balance between high diversity, good coherence, and relatively low computational cost compared to prompting GPT with the LLM-generated topic labels. Prompting GPT with a topic's top 10 words avoids the need to re-label merged topics at each iteration. Llama also shows promise, achieving high diversity and decent coherence with the top 10 topic words. However, it falls slightly short of GPT and can be sensitive to certain corpus content, occasionally refusing to process specific topics.

Finally, LLM prompting tends to outperform agglomerative clustering in both coherence and diversity. However, it is more computationally expensive due to the need for multiple LLM calls and is also slower, potentially incurring costs depending on the application. Since c-TF-IDF must be computed beforehand to generate the top 10 words before prompting can be applied, this method will always be more compute-intensive than agglomerative clustering with the c-TF-IDF embedding.

To support practitioners in selecting an appropriate topic reduction method, we propose a decision flowchart (Fig. 12). It is designed to guide users based on the type of input text (short vs. long), the relative importance of coherence vs. diversity in their use case, and computational resource availability. While the flowchart uses qualitative decision points like "high coherence" and "high diversity" these terms refer to the relative importance of these metrics in the context of the specific downstream task. These are not based on fixed thresholds, as acceptable values can vary by use case and user preference.

The first decision depends on whether your dataset consists of short texts (<40 words on average) or longer texts (>100 words on average). Next, a trade-off is made between coherence and diversity. If coherence is preferred, increasing the minimum cluster size is often sufficient. If diversity is important, an indirect reduction method should be considered. The choice depends on computational resources and processing speed requirements. If sufficient computing power is available and speed is not a major concern, GPT prompting with the top 10 topic words is a strong option. Otherwise, agglomerative clustering based on the topic's c-TF-IDF embedding is a more efficient alternative. For mid-length texts, users are advised to experiment with different approaches while considering the results presented in this study. A reasonable starting point is the benchmark method, followed by agglomerative clustering using the topic's c-TF-IDF embedding.

It should be noted that these recommendations are based on the results obtained for the specific datasets in this study, and outcomes may vary depending on the application. We suggest starting with a BERTopic configuration that produces a manageable number of topics before applying a reduction technique. By experimenting with different configurations and leveraging the outlined techniques, users can achieve optimal results tailored to their datasets and objectives.

VII. LIMITATIONS AND FUTURE RESEARCH

While this study provides a comprehensive comparison of topic reduction methods for BERTopic, a number of limitations remain.

First, this study primarily examines topic reduction within the BERTopic framework. While BERTopic is widely used, other topic modeling techniques, such as Top2Vec, could benefit from similar comparative studies. Future research could explore whether the findings from this study hold across different topic modeling frameworks and investigate how reduction techniques influence results in alternative clustering or embedding configurations.

Second, the study focuses on a limited set of datasets, primarily consisting of short social media texts and customer reviews. While these are relevant for many applications, it remains unclear how well the proposed topic reduction techniques generalize to more structured domains, such as scientific literature, legal documents, or highly technical corpora. Exploring topic reduction in these contexts would be valuable for expanding the applicability of BERTopic.

Third, due to the scale of our experiments, we did not perform human or LLM-based qualitative evaluations. Instead, we relied on validated automated metrics (e.g., NPMI), following the precedent of prior studies [1], [3], [4]. Similarly, we did not apply formal statistical tests (e.g. Friedman test) due to their limited applicability in unsupervised topic modeling and the small number of datasets, which would result in low statistical power [21], [43]. We acknowledge that the absence of statistical testing and human evaluation limits the strength of the evidence and future work could explore incorporating human evaluation and statistical analyses to further validate the comparative findings.

Finally, there are some limitations related to the use of LLM prompting for topic reduction. While these methods offer promising improvements in topic diversity, they come with significant computational costs. In this study, we used GPT-4o-mini and Llama-3-8B, selecting methods that are already well-established and widely adopted. Given the rapid

development in this field and the growing number of available implementations, future work could explore the integration of more lightweight or fine-tuned LLMs which may lead to more scalable and cost-effective solutions for topic reduction. Beyond computational costs, LLMs can also be constrained by token limits, especially when processing an exceptionally large number of topics. In addition, their outputs can be inconsistent and often depend heavily on the quality of prompt engineering and the nature of the content itself. For example, performance may vary when handling sensitive material such as political tweets. Future research could include ablation studies on prompt design and temperature settings to better understand their impact on performance.

REFERENCES

[1] M. Grootendorst, "BERTopic: Neural topic modeling with a class-based TF-IDF procedure," 2022, *arXiv:2203.05794*.

[2] R. Egger and J. Yu, "A topic modeling comparison between LDA, NMF, Top2 Vec, and BERTopic to demystify Twitter posts," *Frontiers Sociology*, vol. 7, May 2022, Art. no. 886498.

[3] A. Abuzayed and H. Al-Khalifa, "BERT for Arabic topic modeling: An experimental study on BERTopic technique," *Proc. Comput. Sci.*, vol. 189, pp. 191–194, Jan. 2021.

[4] B. Ogunleye, T. Maswera, L. Hirsch, J. Gaudoin, and T. Brunsdon, "Comparison of topic modelling approaches in the banking context," *Appl. Sci.*, vol. 13, no. 2, p. 797, Jan. 2023.

[5] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using Siamese BERT-networks," 2019, *arXiv:1908.10084*.

[6] L. McInnes, J. Healy, N. Saul, and L. Großberger, "UMAP: Uniform manifold approximation and projection," *J. Open Source Softw.*, vol. 3, no. 29, p. 861, Sep. 2018, doi: 10.21105/joss.00861.

[7] C. Malzer and M. Baum, "A hybrid approach to hierarchical density-based cluster selection," in *Proc. IEEE Int. Conf. Multisensor Fusion Integr. Intell. Syst. (MFI)*, Sep. 2020, pp. 223–228.

[8] M. de Groot, M. Aliannejadi, and M. R. Haas, "Experiments on generalizability of BERTopic on multi-domain short text," 2022, *arXiv:2212.08459*.

[9] A. Baird, Y. Xia, and Y. Cheng, "Consumer perceptions of telehealth for mental health or substance abuse: A Twitter-based topic modeling analysis," *JAMIA Open*, vol. 5, no. 2, p. 028, Apr. 2022.

[10] B. Koloski, N. Lavrač, B. Cestnik, S. Pollak, B. Škrlj, and A. Kastrin, "AHAM: Adapt, help, ask, model harvesting llms for literature mining," in *Proc. Int. Symp. Intell. Data Anal.*, 2024, pp. 254–265.

[11] D. Li, B. Zhang, and Y. Zhou, "Can large language models (LLM) label topics from a topic model?" *SocArXiv*, 2023. [Online]. Available: https://doi.org/10.31235/osf.io/23x4m

[12] T. Doi, M. Isonuma, and H. Yanaka, "Topic modeling for short texts with large language models," in *Proc. 62nd Annu. Meeting Assoc. Comput. Linguistics Student Res. Workshop*, 2024, pp. 21–33.

[13] H. Wang, N. Prakash, N. K. Hoang, M. S. Hee, U. Naseem, and R. K.-W. Lee, "Prompting large language models for topic modeling," in *Proc. IEEE Int. Conf. Big Data (BigData)*, Dec. 2023, pp. 1236–1241.

[14] C. Minh Pham, A. Hoyle, S. Sun, P. Resnik, and M. Iyyer, "TopicGPT: A prompt-based topic modeling framework," 2023, *arXiv:2311.01449*.

[15] (2024). *GPT-4O Mini: Advancing Cost-Efficient Intelligence*. Accessed: Feb. 7, 2025. [Online]. Available: https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence/

[16] (2024). *Llama 3: Open Foundation and Fine-Tuned Chat Models*. Accessed: Feb. 27, 2025. [Online]. Available: https://ai.meta.com/blog/meta-llama-3/

[17] D. M. Blei and J. D. Lafferty, "Topic models," in *Text Mining*. London, U.K.: Chapman & Hall, 2009, pp. 101–124.

[18] P. Kherwa and P. Bansal, "Topic modeling: A comprehensive review," *EAI Endorsed Trans. scalable Inf. Syst.*, vol. 7, no. 24, Jul. 2018, Art. no. 159623, doi: 10.4108/eai.13-7-2018.159623.

[19] G. Xu, Y. Meng, Z. Chen, X. Qiu, C. Wang, and H. Yao, "Research on topic detection and tracking for online news texts," *IEEE Access*, vol. 7, pp. 58407–58418, 2019.

[20] R. Cheruku, K. Hussain, I. Kavati, A. M. Reddy, and K. S. Reddy, "Sentiment classification with modified RoBERTa and recurrent neural networks," *Multimedia Tools Appl.*, vol. 83, no. 10, pp. 29399–29417, Sep. 2023.

[21] Zoya, S. Latif, F. Shafait, and R. Latif, "Analyzing LDA and NMF topic models for Urdu tweets via automatic labeling," *IEEE Access*, vol. 9, pp. 127531–127547, 2021.

[22] M. Reusens, M. Reusens, M. Callens, S. vanden Broucke, and B. Baesens, "Comparison of different modeling techniques for flemish Twitter sentiment analysis," *Analytics*, vol. 1, no. 2, pp. 117–134, Oct. 2022.

[23] A. Bogdanowicz and C. Guan, "Dynamic topic modeling of Twitter data during the COVID-19 pandemic," *PLoS ONE*, vol. 17, no. 5, May 2022, Art. no. e0268669.

[24] B. Janssens, M. Bogaert, and D. Van den Poel, "Evaluating the influence of airbnb listings' descriptions on demand," *Int. J. Hospitality Manage.*, vol. 99, Oct. 2021, Art. no. 103071.

[25] D. Yu, D. Xu, D. Wang, and Z. Ni, "Hierarchical topic modeling of Twitter data for online analytical processing," *IEEE Access*, vol. 7, pp. 12373–12385, 2019.

[26] A. Crijns, V. Vanhullebusch, M. Reusens, M. Reusens, and B. Baesens, "Topic modelling applied on innovation studies of flemish companies," *J. Bus. Anal.*, vol. 6, no. 4, pp. 243–254, Oct. 2023.

[27] A. Abdelrazek, Y. Eid, E. Gawish, W. Medhat, and A. Hassan, "Topic modeling algorithms and applications: A survey," *Inf. Syst.*, vol. 112, Feb. 2023, Art. no. 102131.

[28] Y. Zhang, R. Jin, and Z.-H. Zhou, "Understanding bag-of-words model: A statistical framework," *Int. J. Mach. Learn. Cybern.*, vol. 1, nos. 1–4, pp. 43–52, Dec. 2010.

[29] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, Jan. 2003.

[30] Q. V. Le and T. Mikolov, "Distributed representations of sentences and documents," in *Proc. Int. Conf. Mach. Learn.*, vol. 4, 2014, pp. 1188–1196.

[31] D. Angelov, "Top2 Vec: Distributed representations of topics," 2020, *arXiv:2008.09470*.

[32] B. Das and S. Chakraborty, "An improved text sentiment classification model using TF-IDF and next word negation," 2018, *arXiv:1806.06407*.

[33] L. Gan, T. Yang, Y. Huang, B. Yang, Y. Y. Luo, L. Richard, and D. Guo, "Experimental comparison of three topic modeling methods with LDA, Top2 Vec and BERTopic," in *Proc. Int. Symp. Artif. Intell. Robot.*, 2024, pp. 376–391.

[34] S. C. Turan, K. Yildiz, and B. Büyüktanir, "Comparison of LDA, NMF and BERTopic topic modeling techniques on Amazon product review dataset: A case study," in *Proc. Int. Conf. Comput. Intell. Data Anal.*, 2024, pp. 23–31.

[35] D. Stammbach, V. Zouhar, A. Hoyle, M. Sachan, and E. Ash, "Revisiting automated topic model evaluation with large language models," 2023, *arXiv:2305.12152*.

[36] Y. Mu, C. Dong, K. Bontcheva, and X. Song, "Large language models offer an alternative to the traditional approach of topic modelling," 2024, *arXiv:2403.16248*.

[37] P. Sahoo, A. Kumar Singh, S. Saha, V. Jain, S. Mondal, and A. Chadha, "A systematic survey of prompt engineering in large language models: Techniques and applications," 2024, *arXiv:2402.07927*.

[38] J. D. M.-W. C. Kenton and L. K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proc. NAACL-HLT*, J. Burstein et al., Eds., Association for Computational Linguistics, 2019, pp. 4171–4186.

[39] *Umap Documentation*. Accessed: Feb. 5, 2024. [Online]. Available: https://umap-learn.readthedocs.io/en/latest/parameters.html

[40] M. Röder, A. Both, and A. Hinneburg, "Exploring the space of topic coherence measures," in *Proc. 8th ACM Int. Conf. Web Search Data Mining*, Feb. 2015, pp. 399–408.

[41] R. Řehuřek and P. Sojka, "Software framework for topic modelling with large corpora," in *Proc. LREC Workshop New Challenges NLP Frameworks*, May 2010, pp. 45–50. [Online]. Available: http://is.muni.cz/publication/884893/en

[42] S. Terragni, E. Fersini, B. G. Galuzzi, P. Tropeano, and A. Candelieri, "OCTIS: Comparing and optimizing topic models is simple!" in *Proc. 16th Conf. Eur. Chapter Assoc. Comput. Linguistics, Syst. Demonstrations*, 2021, pp. 263–270.

[43] S. García, A. Fernández, J. Luengo, and F. Herrera, "Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power," *Inf. Sci.*, vol. 180, no. 10, pp. 2044–2064, May 2010.

WANNES JANSSENS received the M.S. degree (summa cum laude) in business engineering: data analytics from Ghent University, in 2024. He is currently a Ph.D. Researcher with the Data Analytics Research Group, Ghent University. He is with FlandersMake@UGent—corelab CVAMO. His research interests include natural language processing, predictive and prescriptive analytics, and machine learning, with applications in remanufacturing and strategic decision-making.

MATTHIAS BOGAERT is currently an Associate Professor of data analytics with Ghent University, where he teaches social media and web analytics and predictive and prescriptive analytics. He is also affiliated with FlandersMake@UGent—corelab CVAMO, and a Visiting Professor with the University of Namur. He has published in journals, such as *European Journal of Operational Research*, *Machine Learning*, and *Decision Sciences*. His research interests include the applications of descriptive, predictive, and prescriptive analytics in social media, CRM, sports, and production and manufacturing.

DIRK VAN DEN POEL (Senior Member, IEEE) received the master's degree in business engineering and the Ph.D. degree from KU Leuven. He is currently a Senior Full Professor of data analytics, big data, and AI with Ghent University, Belgium. He teaches courses, such as big data and analytical customer relationship management. In 1999, he co-founded the advanced Master's of Science in marketing analysis, the first (predictive) analytics master's program in the world, and the Master's of Science in statistical data analysis and the Master's of Science in business engineering/data analytics. He has co-authored more than 150 peer-reviewed research publications in journals, such as *Journal of Marketing*, *Journal of Statistical Software*, *Journal of Product Innovation Management* (Best Paper Award), in 2017, *Computers and Chemical Engineering*, *European Journal of Operational Research*, IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, *Decision Support Systems*, and IEEE TRANSACTIONS ON POWER SYSTEMS. He has graduated more than 20 Ph.D. students. Several of them are continuing their careers in U.S. He helped more than 100 for-profit and non-profit organizations in applying the aforementioned methods to real-life business cases.