BERTopic: Neural Topic Modeling with a class-based TF-IDF procedure

Megha Manoj **Naganathan Meenakshi Sundareswaran** **Rohan Pujari** **Vishnupriya Santhosh**
mm2773@njit.edu nm749@njit.edu rp992@njit.edu vs263@njit.edu

Abstract

This paper presents a comprehensive comparative analysis of two prominent topic modeling techniques: BERTopic and Latent Dirichlet Allocation (LDA). Topic modeling plays a pivotal role in extracting meaningful insights from large text corpora, aiding in information retrieval and knowledge discovery. BERTopic, leveraging BERT embeddings and c-TF-IDF, stands as a state-of-the-art Python library for topic modeling, while LDA, a generative statistical model, has been a cornerstone in the field. We delve into the methodologies of both approaches, emphasizing their underlying assumptions, techniques for topic representation, and dimensionality reduction. The study evaluates the models' performance on diverse datasets, including 20NewsGroups and BBCNews, utilizing critical metrics such as Topic Word Scores. Additionally, we employ topic coherence as a benchmark to assess the interpretability and relevance of the generated topics. Our findings aim to provide valuable insights into the strengths and limitations of each approach, aiding researchers and practitioners in selecting suitable models for diverse applications.

1 Introduction

In the ever-expanding realm of natural language processing, the quest for effective topic modeling techniques persists. This paper focuses on comparing two influential methodologies—BERTopic and Latent Dirichlet Allocation (LDA)—that stand out in their approaches to unraveling latent topics within textual datasets. BERTopic harnesses the power of BERT embeddings and contextualized TF-IDF, offering a nuanced understanding of word semantics and document structures. On the other hand, LDA, a generative probabilistic model, relies on a different paradigm, assuming documents are mixtures of topics with associated word probabilities. Through a systematic exploration of these models, we aim to provide valuable insights into their comparative performance, usability across diverse datasets, and implications for applications requiring efficient topic modeling. This investigation not only contributes to the ongoing discourse in the field but also aids practitioners and researchers in selecting the most suitable technique based on their specific requirements and objectives.

2 Related Works

The related work for the presented document would likely delve into the broader field of topic modeling, natural language processing (NLP), and techniques for analyzing large text corpora. Specifically, prior research might explore alternative approaches to topic modeling, such as Latent Semantic Analysis (LSA) and Latent Dirichlet Allocation (LDA), emphasizing their strengths and limitations. Additionally, related work could discuss advancements in word embedding techniques beyond BERT embeddings, considering methods like Word2Vec or GloVe.

Comparative studies between BERTopic and other state-of-the-art topic modeling tools, along with investigations into different dimensionality reduction and clustering algorithms, would be relevant. Moreover, research on evaluation metrics for topic coherence and performance, as highlighted in the document, would contribute to understanding the effectiveness of various topic modeling methodologies. Overall, the related work section would aim to position BERTopic within the broader landscape of text analysis techniques, providing context and insights into the evolution of topic modeling methodologies.

3 BERTopic

BERTopic is a state-of-the-art Python library that simplifies the topic modeling process using various embedding techniques and c-TF-IDF to create dense clusters, allowing for easily interpretable topics while keeping important words in the topic descriptions. It organizes large amounts of text data by grouping similar topics together. It uses BERT embeddings, which are powerful representations of words in the context of sentences, to find similarities and differences between pieces of text. Then, it groups similar pieces into topics. BERTopic helps you make sense of large text collections by automatically identifying and organizing topics, making it easier to analyze and explore the content.

The three steps through which BERTopic generates topic representations are:

1. Converting each document to its embedding representation using a pre-trained language model
2. The dimensionality of the resulting embeddings is reduced and then clustered.
3. Lastly, topic representations are extracted using a custom class-based variation of TF-IDF.

To identify the latent description of a corpus (text or document), two assumptions are made: each document (text or corpus) consists of a mixture of topics, and each topic consists of a collection of words. The LSA (Latent Semantic Analysis) approach uses a document-term matrix representing the frequency of terms in each document. LDA (Latent Dirichlet Allocation) is a generative statistical model that assumes documents are made up of words that aid in determining the topics. Thus, documents are mapped to a list of topics by assigning each word in the document to different topics. This model ignores the order of words occurring in a document and treats them as a bag of words. Both the above approaches lack the semantic or contextual meaning of the words and that's where BERTopic has an edge by leveraging BERT embeddings and c-TF-IDF.

Both the above approaches lack the semantic or contextual meaning of the words and that's where BERTopic has an edge by leveraging BERT embeddings and c-TF-IDF.

3.1 Embedding representation

Embeddings are short dense vectors of real numbers in a high-dimensional space to represent words in documents. Words with similar meaning have the same representations. Word embeddings provide an effective path to recognize and capture the semantic relationships between words. Additionally, embeddings can be used for clustering similar words or visualizing relationships between words in a meaningful way which is an essential step in the process of topic modeling.

By default, the BERTopic model uses sentence transformer, all-MiniLM-L6-v2. However, there are a whole range of options of sentence transformers available that can be chosen based on the application. Further, there are a number of models available from which any one to be chosen as an embedding model. The flexibility of the BERTopic to choose an appropriate model or even better, to build a customized, own model is called *Modularity* by the author.

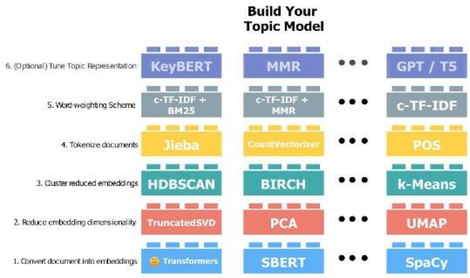

Figure.1: Modularity

The options for embedding models include, but not limited to, SBERT, Spacy, Huggingface Transformers, Cohere, Universal Sentecne Encode (USE) and even OpenAI backend.

3.2 Dimensionality reduction

The extracted, input embeddings tend to be of larger dimensions depending on the number of documents and hence might be difficult to process due to the curse of dimensionality. BERTopic applies dimensionality reduction as a default in its pipeline. UMAP is the default algorithm applied as it can capture both the local and global high-dimensional space in lower dimensions. UMAP is a nonlinear dimensionality reduction technique, which means it can capture complex relationships and structures that may not be well-represented by linear methods like principal component analysis. Since it has no computational restrictions on embedding dimensions, UMAP can be used across language models with differing dimensional space.

As represented in Figure 1, there are other solutions such as PCA (Pricipal Component Analysis) and TruncatedSVD are also available for reducing dimensionality. cuML can be used to speed up UMAP through GPU acceleration since, at times, there may be difficulty in handling large amounts of data.

3.3 Clustering

Since there is plurality of topics, and text is distributed, topic modeling is not essentially aiming to find similarities in documents, but rather a specific topic representing a cluster of documents.

As a default, BERTopic uses HDBSCAN (Hierarchical Density-Based Spatial Clustering of Applications with Noise) to perform its clustering. The algorithm combines elements of density-based and hierarchical clustering and is particularly useful when dealing with data containing clusters of varying shapes and densities.

Though later the features of BERTopic will be explained, Figure 2 shows the cluster of documents and related topics.

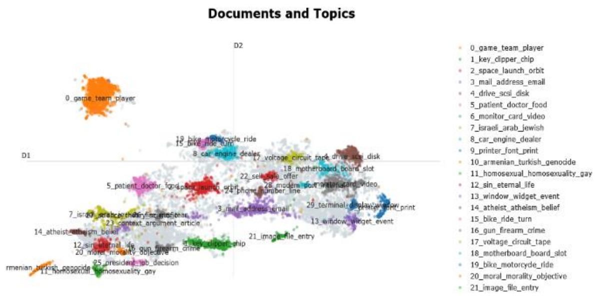

Figure 2: Visualization of cluster of documents and topics.

It can be observed that a topic can represent multiple documents and a document may also be distributed across topics with different probabilities.

Apart from HDBSCAN, BERTopic supports K-Means, Agglomerative clustering, cuML HDBSCAN, BIRCH models for clustering.

3.4 Topic representation using c-TF-IDF

A topic representation is usually a set of most important words specific to this topic and not others. They are modeled based on the documents in each cluster where each cluster will be assigned one topic. To find out what distinguishes one topic from another based on the words in its cluster, a class-based TF-IDF is applied. It considers the term frequency within a class (topic) and inversely weighs it by the frequency of the term in other classes. The original TF-IDF formula measures the representation of the importance of a word to a document, while the adaptation measures the representation of a term’s significance to a topic instead. The purpose of class-based TF-IDF is to provide the same class vector for all documents of the same class. In this context, each cluster is considered a class.

Each cluster is converted to a single document instead of a set of documents. The frequency of each word x is extracted for each class c. This results in the class-based tf representation. This representation is L1-normalized to account for the differences in topic sizes.

Logarithm is taken to one plus the average number of words per class A divided by the frequency of word x across all classes. We add plus one within the logarithm to force values to be positive. This results in our class-based idf representation.

$$W_{x,c} = \left|\left|tf_{x,c}\right|\right| . \log\left(1 + \frac{A}{f_x}\right)$$

$tf_{x,c}$: frequency of word w in class c fx: frequency of all word w across all classes

A: Average number of words per class.

The class-based TF-IDF procedure models the importance of words in clusters instead of individual documents. This allows us to generate topic-word distributions for each cluster of documents.

3.5 Features and options in BERTopic

One aspect that repeats in all those pipeline steps is *Modularity*. As the author puts it, in practice, there is not one correct way of creating embeddings, reducing dimensions, clustering, and creating topic representations. Thus, BERTopic is the state-of-the-art library in providing flexibility to choose any available module than the default for any of the tasks - embeddings, dimensions, clustering, tokenizing, and topic representations. Figure3 is a snapshot of a code block that shows the BERTopic model having options for each of the tasks.

```
from sklearn.cluster import KMeans
from sklearn.decomposition import PCA
from sklearn.feature_extraction.text import CountVectorizer
from sentence_transformers import SentenceTransformer
from bertopic.vectorizers import ClassTfidfTransformer

topic_model = BERTopic(

    # 1. Extract Embeddings
    embedding_model=SentenceTransformer("paraphrase-multilingual-mpnet-base-v2")

    # 2. Reduce Dimensionality
    umap_model=PCA(n_components=5),

    # 3. Cluster Documents
    hdbscan_model=KMeans(n_clusters=50),

    # 4. Tokenize Topics
    vectorizer_model=CountVectorizer(stop_words=my_dutch_stopwords),

    # 5. Extract Representative Words
    ctfidf_model=ClassTfidfTransformer(bm25_weighting=True),
)
```

Figure 3: BERTopic model declaring with options to choose modules for each of the tasks.

Having seen the capability of the BERTopic model in terms of embeddings, clustering, topic representation, below are some of the visualizations to understand the topic extractions. The BERTopic model provides features to visualize topics, terms, documents and their relation, distribution and a possible hierarchy. Figure 4 shows intertopic distance to show clusters of topics with similar words. For e.g., a document about genetics topic is more likely to be about infectious diseases topic than astronomy or politics and hence clusters of topics and documents. c-TF-IDF representations of the topics are embedded in 2D using UMAP and then visualized the two dimensions using plotly such that an interactive view is possible.

A similarity matrix by simply applying cosine similarities through those topic embeddings created by c-TF-IDF and embeddings can be created. The result will be a matrix indicating how similar certain topics are to each other. To visualize the heatmap as shown in Figure 5.

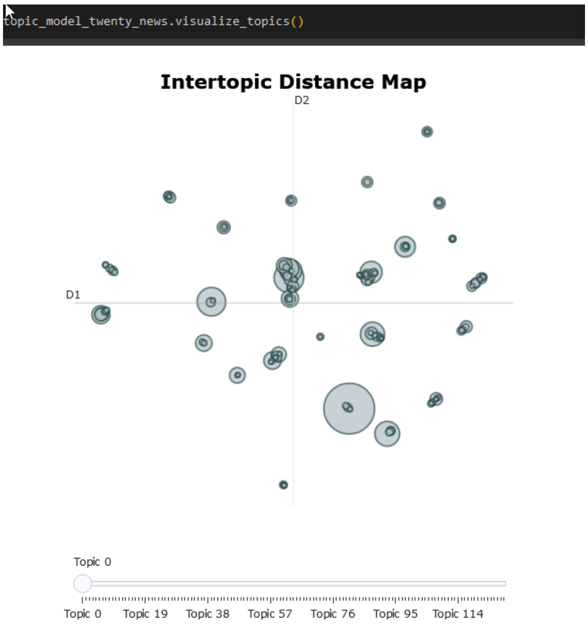

Figure 4: Inter-topic distance

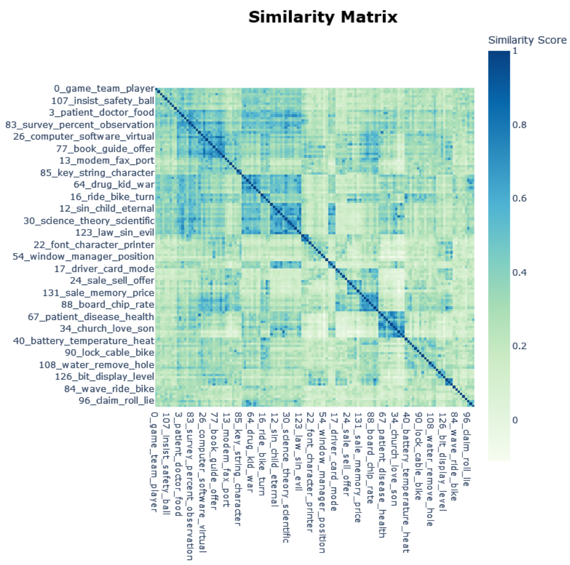

Figure 5: Heatmap – similarity matrix

A more fine-grained approach where visualization of the documents inside the topics to see if they were assigned correctly or whether they make sense is possible as already shown in Figure 2.

It is possible to visualize the selected terms for a few topics by creating bar charts out of the c-TF-IDF scores for each topic representation. Insights can be gained from the relative c-TF-IDF scores between and within topics which are shown under the section model performance.

Further, the BERTopic model provides options for Hierarchical topic modeling - to model the possible hierarchical nature of the topics created to understand which topics are similar to each other; Dynamic topic modeling (DTM) - to understand how a topic is represented across different times, BERTopic allows for DTM by calculating the topic representation at each timestep without the need to run the entire model several times.

Hierarchical representation of topics and documents are shown in Figure 6.

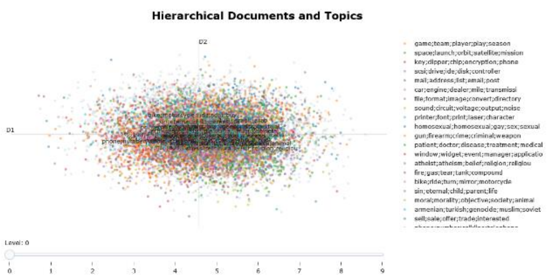

Figure 6: Hierarchical topic modeling

So far whatever has been discussed and explained are part of the features of the BERTopic model. The model provides even more options and is constantly enhanced to integrate the latest language models. The below Figure 7 represents the potential of the model with its modularity.

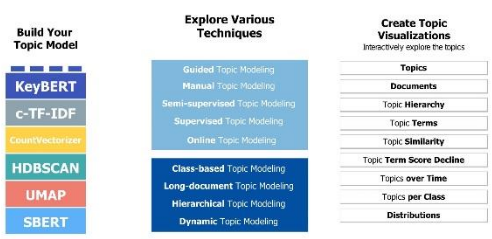

Figure 7: Potential of BERTopic model

4 LDA

Latent Dirichlet Allocation (LDA) is a popular topic modeling technique to extract topics from a given corpus. The term latent conveys something that exists but is not yet developed. In other words, latent means hidden or concealed. Each document is made up of various words, and each topic also has various words belonging to it. LDA aims to find topics a document belongs to, based on the words in it.

Let's say we have 5 documents each containing the words listed in front of them (ordered by frequency of occurrence). What we want to figure out are the words in different topics, as shown in the table below.

	Word1	word2	word3	word4	.....
Topic1	0.01	0.23	0.19	0.03	
Topic2	0.21	0.07	0.48	0.02	
Topic3	0.53	0.01	0.17	0.04	

Figure.8: A sample word and topic map for LDA

Each row in the table represents a different topic and each column a different word in the corpus. Each cell contains the probability that the word(column) belongs to the topic(row). The 2 parts in LDA are words that belong to a document, that we already know and the words that belong to a topic or the probability of words belonging to a topic, that we need to calculate.

LDA uses Dirichlet distributions to model the mixtures of topics in documents and the mixtures of words in topics. The model makes use of the following hyperparameters:

Alpha: A parameter that represents the document-topic density. Higher values of alpha will lead to documents being composed of more topics, and lower values will lead to documents being composed of fewer topics.

Beta: A parameter that represents the topic-word density. Similar to alpha, higher values of beta will lead to topics being composed of a larger number of words, and lower values will lead to topics being composed of fewer words.

We have used an alpha of 0.1 and specified the number of topics for each specific dataset while building and testing our model on the selected datasets.

5 Results and Conclusions

5.1 Model Performance

The BERTopic model was executed on two distinct datasets: 1) 20NewsGroups and 2) BBCNews. The resultant top topics, ranked by their respective Topic Word Scores—a critical metric in BERTopic modeling—are presented herewith for both datasets. The Topic Word Score holds paramount significance in the BERTopic framework for identifying and characterizing topics.

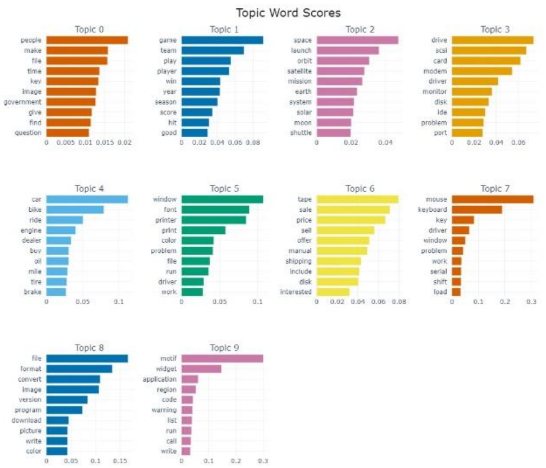

Figure.9: Topic word scores for 20 News Groups

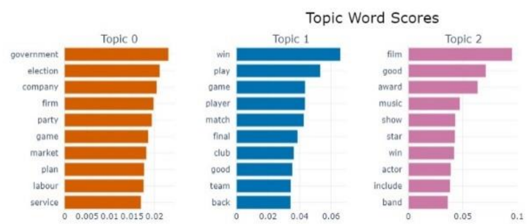

Figure.10: Topic word scores for BBC News

5.2 Evaluation Metric – Coherence Metric

Topic coherence is a measure used to evaluate the interpretability and meaningfulness of topics generated by a topic modeling algorithm. It provides a quantitative metric to assess the quality of the topics discovered from a corpus of text. The significance of topic coherence in topic modeling lies in its ability to help researchers and practitioners gauge the coherence and relevance of the identified topics.

Significance of topic coherence in topic modeling:

Interpretability Assessment:

Topic coherence serves as a tool for evaluating how interpretable and coherent the identified topics are. It helps in understanding whether the words within a topic are semantically related and form a meaningful cluster.

Model Evaluation:

Topic coherence provides a quantitative measure for comparing different topic models or variations of a model. Researchers can use it to assess which models produce topics that are more coherent and thus more likely to align with human understanding.

Optimizing Hyperparameters:

In the context of probabilistic topic models, such as Latent Dirichlet Allocation (LDA), tuning hyperparameters is crucial. Topic coherence can guide the selection of optimal hyperparameters by identifying parameter values that lead to more coherent topics.

Ensuring Meaningful Representation:

High topic coherence implies that the words within a topic are closely related and convey a clear theme. This is essential for ensuring that the topics extracted from the text are meaningful and can be easily interpreted by users.

Improving Model Robustness:

Models with higher topic coherence tend to be more robust and reliable in capturing meaningful patterns in the data. By aiming for high coherence, practitioners can enhance the robustness of the topic modeling results.

Facilitating Topic Labeling:

Coherent topics are easier to label and describe. Topic coherence metrics can guide the process of assigning meaningful labels to topics, aiding in the interpretation and communication of results.

User Satisfaction:

Ultimately, the goal of topic modeling is to provide valuable insights to users. Coherent topics contribute to user satisfaction by delivering results that are not only accurate but also easily understandable.

In summary, topic coherence is a vital aspect of topic modeling as it allows for the quantitative assessment of the quality and interpretability of the identified topics, guiding model selection, parameter tuning, and overall model evaluation.

The performances of the models were compared using the coherence scores and the results are presented in table (Table 1). The scores clearly state that BERTopic is a better model when compared to the LDA model for both the datasets.

	BERTopic	LDA
20NewsGroups	**0.572**	0.054
BBC News	**0.664**	-0.017

Table 1: Coherence Matrix

References

1. https://arxiv.org/pdf/2203.05794.pdf
2. https://maartengr.github.io/BERTopic/index.html
3. https://www.sbert.net/docs/pretrained_models.html
4. https://maartengr.github.io/BERTopic/getting_started/embeddings/embeddings.html
5. https://maartengr.github.io/BERTopic/index.html#modularity
6. https://maartengr.github.io/BERTopic/index.html#sentence-transformers