An Exploratory Analysis of GSDMM and BERTopic on Short Text Topic Modelling

Adarsh K N, Abhinandan Udupa, Anvitha Aravinda, Neelam Godihal

Department of Computer Science and Engineering

B M S College of Engineering
Bengaluru, Karnataka 560019, India

{adarshkn, abhinandan, anvitha}.cs19@bmsce.ac.in, neelamh.ec19@bmsce.ac.in

Dr. Kayarvizhy N

Associate Professor
Department of Computer Science and Engineering
B M S College of Engineering
Bengaluru, Karnataka 560019, India

kayarvizhy.cse@bmsce.ac.in

***Abstract* - Topic models may be a useful tool for locating latent subjects in collections of documents. Short text clustering has become a more important task as social networking sites like Twitter, Google+, and Facebook have gained popularity. It is a challenging problem because of its sparse, high-dimensional, and large-volume characteristics. Two of the most well-known short text modelling algorithms are BERTopic and the Gibbs Sampling Dirichlet Multinomial Mixture Model (GSDMM). We found that GSDMM can infer the count of clusters automatically with a good balance between the completeness and homogeneity of the clustering results, and is fast to converge.BERTopic is a topic model that extends this technique by extracting coherent topic representations via the creation of a class-based form of TF-IDF. We compare the two algorithms in this research to determine which can be utilised for short text topic modelling the most effectively.**

Index Terms - GSDMM, BERTopic, Coherence.

1. Introduction

Short texts, such as news headlines, status updates, web page excerpts, tweets, question/answer pairs, and so on, have become a major source of information. As a result, short text analysis has gained traction in recent years. Effective and efficient models infer latent themes from brief texts, which can aid in the discovery of latent semantic structures in a set of documents. Short text topic modelling techniques are used in a variety of applications, including subject identification, categorization, comment summarization, and user interest profiling. Short text modelling has many problems associated with it. As there are fewer words in a sentence than in a lengthy paragraph, TF-IDF scores are not very useful as all the scores will be equal to. And dealing with high-dimensional data, using vector space leads to sparsity. The result is a high computation and memory storage need. As a result of the above issues, clustering the text into topics becomes increasingly difficult, and we lose the capacity to generate coherence topics.

1.1 Dirichlet Distribution

The Dirichlet distribution is essentially a multidimensional Beta distribution (documents). A Beta distribution is basically a probability distribution that represents the previous state chance of a document entering a cluster as well as its resemblance to the cluster. The form of both the Beta distribution is controlled by two factors ,i.e, alpha and beta.

1.2 Neural Topic Modelling

Recently, neural topic modelling has gained significant attention since it can combine the benefits of probabilistic topic models with neural networks.This method has three key benefits over traditional ones. Firstly, unlike the traditional technique, which necessitates dealing with a challenging optimization task, its inference is compounded and hence far more computationally straightforward. Second, the effectiveness of some deep learning frameworks like Pytorch, Tensorflow, and Flux.jl helps to narrow the gap between prototype and deployment procedures. Third, pre-trained word and text embeddings, which are prevalent and have shown to be quite beneficial, such as GPT-3 and BERT, are simple to incorporate with neural topic models.

2. Definitions

2.1 Gibbs Sampling For Dirichlet Multinomial Mixture (GSDMM)

GSDMM can automatically infer the count of clusters while maintaining a good balance between the completeness and homogeneity of the clustering findings, and it is rapid to

converge. GSDMM can also handle the sparse and high-dimensional issue of brief texts, obtaining the representative words of each cluster. The model promises to tackle the sparsity problem of brief text clustering while also presenting word themes like LDA. GSDMM is simply a modified LDA (Latent Dirichlet Allocation) that assumes a document (such as a tweet or text) has just one subject. GSDMM can automatically calculate the count of clusters and has a perfect balance between the completeness and homogeneity of grouping findings, as well as a quick convergence time, making it more successful than LDA at extracting hidden subjects from short texts.

2.2 BERTopic

Through the construction of TF-IDF, a topic model that extends the process of detecting latent topics in a corpus by extracting coherent topic representation. Traditional methods, such as Latent Dirichlet Allocation (LDA) and Non-Negative Matrix Factorization (NMF), characterise a document as a collection of words and represent each document as a collection of latent themes. One disadvantage of these models is that they neglect semantic links between words by using bag-of-words representations. The bag-of-words input may fail to correctly represent documents since these representations do not account for the context of words in a phrase.Text embedding approaches have gained popularity in the field of natural language processing as a solution to this problem. Word Embedding is a numeric vector input that depicts a word in a document as a dense matrix in a lower-dimensional space. It enables similar-sounding words to have comparable representations. Three stages are used by BERTopic to produce topic representations. Each document is first transformed using a trained language model into its embedding representation. The dimensionality of the generated embeddings is then decreased prior to clustering these embeddings in order to improve the clustering procedure. Finally, topic representations are retrieved from the document clusters using a custom class-based form of TF-IDF.

3. Approach

3.1 GSDMM

3.1.1 Movie Group Approach

It is analogous to the challenge of short text clustering. This example is utilised throughout the study and can assist in understanding both the short text clustering problem and the GSDMM method.

We may assume that the lecturer of a film conversation class intends to split the students into groups. She anticipates that the kids in the very same group will have seen more of the same movies, giving them additional topics to talk about. The professor instructs the pupils to jot down the movies they have seen in the last few minutes. (Since the students may not have enough time, the list will be short, and they will most likely jot down movies they have recently seen or movies they adore.) Each student can now be represented by a list of videos.The lecturer must discover a means to divide the pupils into groups. Students in the same class should have comparable interests (similar film lists), but students in separate groups should have diverse interests.The same thing is considered with respect to documents, where each document is represented as a student. Movies represent words in the document.

Common similarity-based methods for text clustering, such as K-means and HAC, use the Vector Space Model to describe the documents. Every document (student) is defined as a V-length vector. Each vector element represents the weight of the relevant word. Because of the scarcity of text documents, most words in the articles have TF=1, implying that TF is almost worthless in the depiction of short texts. Despite the fact that each brief text contains only a few words, it is represented by a vector of size V. It may envision the professor inviting the students to a large restaurant and randomly assigning them to K tables. Then she instructs the pupils to re-select a table one at a time. We may anticipate that a student will select a table depending on the aforementioned two rules:

- Rule 1: Select a table with a larger number of pupils.
- Rule 2: Select a table with kids who have comparable interests (i.e., have viewed more films of the same genre) as him.

As the process progresses, certain tables will expand in size while others will disappear. We may anticipate that, eventually, only a portion of the tables will contain students, and that the students in each table will have comparable interests. In other terms, the lecturer can divide the students into different groups in this manner.

3.1.2 Dirichlet Multinomial Mixture

DMM as mentioned before comprises few key parameters in the model. They are:

The parameter alpha influences the form of probability distribution. More crucially, alpha is calculated from the likelihood of a document being categorised into a

cluster. In the film instance, this is the likelihood of a student selecting a table.

$$p(d|z=k) = \prod_{w \in d} p(w|z=k)$$

The other shape factor for distribution is beta. The term beta refers to the resemblance of words in one text to those in another. In terms of film groups, beta is the likelihood that a student would join a table with comparable movie preferences. If beta is 0, the student will only connect tables that have movies in common. This may not be the greatest approach. Maybe two pupils like suspense movies, but they didn't list the same ones. We really want the pupils to end up in the same thriller-loving group. Phi is the multinomial distributions of groupings over words with k clusters (mixtures), such that p(w|z = k) = phi, where w = words and z = cluster label. Similarly, because theta is a multinomial distribution that includes alpha, p(d|z=k) = theta, where d = document. These factors add up to the likelihood that a document (d) is created by a cluster (k), assuming Dirichlet priors. The study makes the assumption of symmetric Dirichlet priors. This implies that in the outset, the identical alphas and betas are assumed. Alpha denotes that the same groups are equally essential, whereas beta denotes that the same phrases are equally as important.

3.1.3 Gibbs Sampling for DMM

The collapsed Gibbs Sampling approach for the Dirichlet Multinomial Mixture model (abbreviated GSDMM), which is identical to the Movie Group Process (MGP), is explained in this section. "In the initial step of algorithm documents are randomly assigned to k clusters,and following information is taken down:$\rightarrow z$ (cluster labels of each document), mz (number of docu- ments in cluster z), nz (number of words in cluster z), and n^w (number of occurrences of word w in cluster z). Then we traverse the documents for I iteration. In each iteration, we re-assign a cluster for each document d in turn according to the con- ditional distribution: $p(z_d = z \mid \rightarrow z_{\neg d}, d \rightarrow)$, where $\neg d$ means the cluster label of document d is removed from $\rightarrow z$. Each time we re-assign a cluster z to document d, the corresponding information in $\rightarrow z$, mz , nz , and n^w are updated accordingly. Finally, only a part of the initial K clusters will remain non-empty, in other words, GSDMM can cluster the documents into several groups. We know that the fraction of clusters that are not empty found by GSDMM can be near the true number of groups as long as K is larger than the true number. GSDMM is also a soft clustering model like Gaussian Mixture Model (GMM), since we can get the probability of each document belonging to each cluster from $p(z_d = z|\rightarrow z_{\neg d}, d\rightarrow)$. We can derive $p(z_d = z|z\rightarrow_{\neg d}, d\rightarrow)$ from the Dirichlet Multinomial Mixture (DMM) model, and find that it conforms to the two rules of MGP introduced."

3.2 BERTopic

3.2.1 Document Embedding

In this approach, documents are embedded using neural networks such that their representations in the vector space are semantically comparable to each other. We make the assumption that documents with the same subject matter are semantically related. Sentence-BERT (SBERT) is a framework that BERTopic employs to carry out the embedding stage. Using pre-trained language models, this system enables users to transform phrases and paragraphs into dense vector representations. On several sentence embedding tasks, it performs at the cutting edge. However, rather than directly producing the topics, these embeddings are mostly used to group documents that share comparable semantic properties. The language model producing the document embeddings can be turned on semantic similarity to produce more coherent topics of higher quality. As new and improved language models are created, the quality of clustering in BERTopic will therefore improve. This enables BERTopic to advance along with the state-of-the-art embedding technologies.

3.2.2 Document Clustering

Document clustering is a technique for identifying structure in a set of documents so that related documents may be categorised. It has been demonstrated that the hellinger distance (the distance metric used by UMAP) to the nearest dataset, which is represented as a point in the vector space, gets closer to the distance to the furthest dataset as data becomes more dimensional. As a result, in strong space, the concept of spatial proximity is ill-defined, and distance measures differ little. Although there are clustering methods for overcoming this dimensionality curse, reducing the dimensionality of embeddings is a simpler solution. Despite the fact the PCA and t-SNE are very good algorithms for reducing dimensionality, UMAP has proved that it can retain more of the local and global properties which are characteristic of high dimensional data projected in lower dimensions. Because there are no computational constraints on the embedding size, UMAP may be used with language models with varying spatial dimensions.Therefore, in order to lower the produced document embeddings' dimensionality the clustering method uses reduced embeddings.HDBSCAN is a DBSCAN extension that looks for clusters of different sizes by transforming DBSCAN into a hierarchical clustering

method.A soft-clustering method, HDBSCAN models clusters while permitting noise to serve as outlier models. This approach tends to avoid allocating unrelated documents to any cluster and enhances the topic representations. Additionally, it has been shown that lowering high dimensional embeddings with UMAP, well-known clustering methods like the k-Means approach and the HDBSCAN approach may both perform better in terms of time and clustering precision.

3.2.3 Topic Representation

The documents in each cluster are used to model the topic representations, and each cluster will be given a specific topic. Based on the cluster-word distribution of each topic, to distinguish one topic from the other we may choose to alter the TF-IDF score, which is a common metric that is used for representing the relevance of a word to its parent document, in a way that it might instead reflect the significance of a term to a topic. Term frequency and inverse document frequency are two statistics that are combined in the traditional TF-IDF process.

$$W_{t,d} = tf_{t,d} \cdot \log(\frac{N}{df_t})$$

Where the term frequency $tf_{t,d}$, models the occurrence of the term t in the document d. The inverse document frequency—a measure of how much knowledge a word adds to a document—is calculated by dividing the log of the number of documents in a corpus N by the total number of documents containing the term t. This approach is extended to document clusters. First, we view each item in a cluster as a single document by simply concatenating them. Then, by transforming documents to groups, TF-IDF is updated to take this representation into account:

$$W_{t,c} = tf_{t,c} \cdot \log(1 + \frac{A}{tf_t})$$

The term frequency $tf_{t,c}$ models the frequency of the term t in a class c. The group of documents that have been merged together to form a single document that represents each cluster of the class c in that instance. Then, to gauge how much knowledge a word contributes to a class, the TF-IDF score is replaced by the class-based TF-IDf score. It is calculated by dividing the average number of words in the class c by the logarithm of the frequency of the term t across all classes. We multiply the division of the logarithm by one in order to produce only positive numbers.In order to represent the relevance of words in clusters rather than in individual texts, this class-based TF-IDF approach is used. In this way, topic-word distributions for each cluster of documents can be produced. We may also limit the number of topics that are generated to a number that the user can specify. This can be achieved by recursively merging the resultant class-based TF-IDF representations of the smallest topic (the one with the smallest cluster size) with its most comparable one.

4. Experimental Setup

Here, we describe the parameter setting of the two models that we have chosen for the short text topic modelling, the datasets we have used and the evaluation metrics used. The experiments were performed on a Google Colab GPU Instance which runs on a Intel Xeon Haswell CPU having 2 cores at 2 Ghz and 12GB of RAM and a Nvidia K80 GPU with 12GB of vRAM. We have set the number of iterations for both models at 40 unless otherwise expressly stated, using the settings suggested by the respective authors. We utilised "*all-MiniLM-L6-v2*", a pre-trained BERT based model, as the default BERT model for BERTopic. This BERT based model represents sentences in a 384 dimensional dense vector space . It is trained and tuned for applications related to semantic search and clustering. Both models have been configured to generate 15 topics for each dataset.

4.1 Datasets

We use the following three datasets to compare and contrast the two algorithms, GSDMM and BERTopic. Following preprocessing these datasets, we summarise their essential details in Table I, where N denotes the total number of samples (documents) in each dataset and Count denotes the mean number of words and the maximum number of words of each sample.

TABLE I
Details of Dataset

Dataset	N	Count
Trump Twitter Archive Dataset	931	18.8/55
ABC News Headlines Dataset	1000	6.5/10
Stack Overflow Dataset	1000	8.6/29

4.1.1 Trump Twitter Archive Dataset

Donald Trump, a former US president, was well-known for using Twitter often. After the violent rioting at the US Capitol building on January 6, 2021, the platform decided to ban his account on January 8th, claiming "the potential of future provocation of violence." Trump's social media activity serves as a significant record of the US political and cultural discourse's rising polarisation in the second decade of the 2000s. Trump's tweets from November 2019 to

December 2019 are included in this dataset. It was taken from "The Trump Archive," a website that handled all the effort of occasionally scanning Trump's Twitter account until his suspension in 2021.

4.1.2 ABC News Headline Dataset

This data set contains news headlines issued across a number of years. ABC News, a reputable Australian news source (Australian Broadcasting Corporation). This comprises all of the articles published on the abcnews website. With a daily volume of 200 articles and a strong emphasis on international news, we can be quite assured that every significant event has been covered in this dataset. By diving into the keywords, one can observe all of the major events that shaped the last decade and how they evolved over time.

4.1.3 Stack Overflow Dataset

Stack Overflow is the largest online community for programmers to study, share information, and further their careers. This BigQuery dataset comprises Stack Overflow material such as posts, polls, labels, and badges. This dataset, which can be accessed via the Stack Exchange Data Explorer, has been updated to match the Stack Overflow content on the Internet Archive.

4.2 Parameter Setting

GSDMM: We set k = 300, $\alpha = 0.1$ and $\beta = 1$, n = 40 as declared in the paper (GSDMM).

BERTopic: We set n_gram range as 2-3, MMR (Diversity) as None or 0.25 unless explicitly specified and consider only the top 10 words from each topic.

4.3 Evaluation Metric - Coherence

If a group of claims or facts corroborate one another, the group is said to be coherent. As a result, a coherent set of facts can be interpreted in a setting that includes all or the majority of the facts. A coherent fact set might include statements like "Bitcoin is an electronic currency" "It is decentralised electronic cash that does not rely on banks," and "the usage of bitcoin is increasing across the world".

By determining the semantic similarity measure between the topic's high-scoring words, Topic Coherence evaluates a single topic. These metrics assist in separating subjects that are artefacts of statistical inference from topics that are semantically interpretable.

The *c_v* measure of coherence uses a sliding window, one-set segmentation of the top words that are present in a topic, and also a confirmation measure that uses the cosine similarity and normalised pointwise mutual information (NPMI).

We picked the coherence metric based on Pointwise Mutual Information (PMI) because it provided the highest correlations with human ratings.

5. Analysis

5.1 GSDMM

This section will examine how GSDMM performs when the alpha α, beta β, and number of iterations are varied. The count of clusters corresponds to the number of topics identified by GSDMM. Therefore, the count of clusters is a valid measure to evaluate its performance by varying the parameters like alpha α, beta β, and number of iterations.

5.1.1 Effect of Alpha

This section aims to study the impact of α on the count of clusters identified by GSDMM. For all datasets, we set $\beta = 1$, the count of clusters to 300, and the number of iterations to 10.

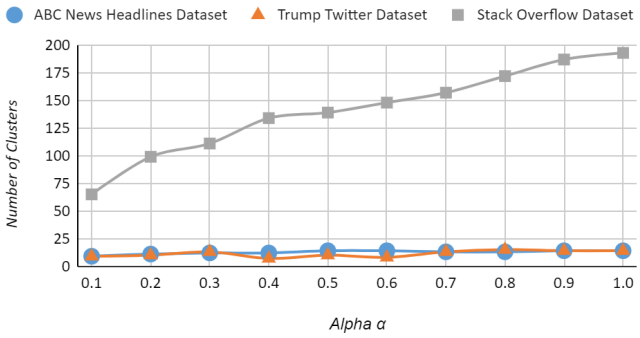

Fig. 1 Effect of α

The count of clusters identified by GSDMM on the three datasets with various values of α is depicted in the above figure. We can see that as α increases on the Stack Overflow Dataset, a greater fraction of clusters that are not empty is identified by GSDMM. We examine the clustering outcomes for this dataset and discover that larger α tends to produce clusters that contain just one document. With the aid of the Movie Group Process (MGP), we can explain this occurrence. A student will be more likely to select an empty table as α increases. However, due to MGP's "richer gets richer" feature, if a table is empty, there is a low likelihood that it will continue to grow large. As a result, as α increases, more clusters containing only one document will be identified by GSDMM and the fraction of clusters that are not empty with only one document will also grow. We can see that in both the

Trump Twitter Dataset and the ABC News Headlines Dataset, the fraction of clusters that are not empty identified by GSDMM tends to stay stable. In accordance with the Movie Group Process analogy, this is explained by the fact that, for these datasets, students are more likely to select a table based on their interests, decreasing the likelihood that the empty tables would be chosen.

5.1.2 Effect of Beta

This section aims to study the impact of β on the count of clusters identified by GSDMM. For all datasets, we set α = 0.1, the count of clusters to 300, and the number of iterations to 10.

The count of clusters identified by GSDMM on the three datasets with various values of β is depicted in the above figure. We can observe that as we increase β, the count of clusters identified by GSDMM drops. In accordance with the Movie Group Process analogy, this is explained by the fact that a student selects a table whose members have interests in common with him. When β is small, the likelihood that a student will select a table depends more on the frequency of the word w in cluster z.

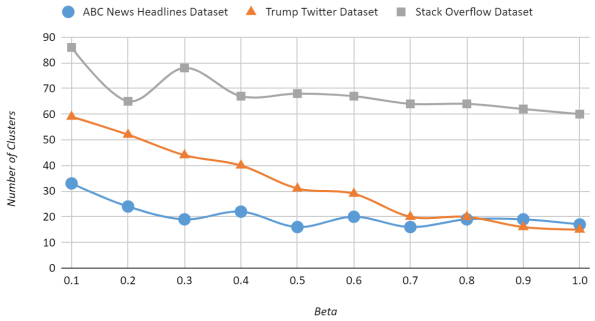

Fig. 2 Effect of β

To put it another way, when β is small, GSDMM places more focus on the student's interests, and students are therefore more likely to select a table based on their interests than on how popular it is. Therefore, when β is small, GSDMM will produce more non-empty tables (clusters). Similar to this, GSDMM will produce fewer non-empty tables when β is high.

5.1.3 Effect of Number of Iterations

This section aims to study the impact of the number of iterations on the count of clusters identified by GSDMM. We set α = 0.1, β = 1 for all datasets and the initial number of clusters at 300.

The count of clusters identified by GSDMM on the three datasets with different numbers of iterations is depicted in the figure below. We can observe that the count of clusters rapidly decreases and nearly flattens after fifteen iterations. In order to understand this phenomenon, we use the ABC News Headlines Dataset as an example. GSDMM randomly distributes the dataset's 1,000 documents (students) into 300 clusters during the initialization phase (tables). GSDMM counts the students for each iteration and allows them to choose a table again in accordance with the two rules of Movie Group Process (MGP) that were previously discussed. Some clusters grow quickly at first because, once chosen by a student, they are more likely to be chosen by other students with similar interests. Similarly, some clusters vanish instantly. On the ABC News Headline Dataset, we can see that after five iterations, the fraction of clusters that are not empty decreases from 300 to roughly 42, andafter ten iterations, the number decreases to approximately 28. The allocation of documents will become steady with many iterations.

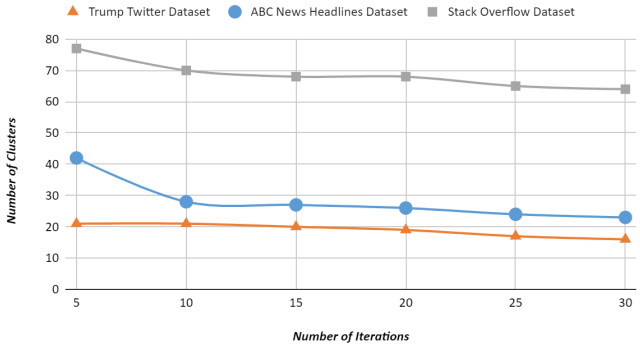

Fig. 3 Effect of Number of Iterations

5.2 BERTopic

In this section we analyse three important parameters for BERTopic - the N-gram Range, Diversity/MMR and the top `n` words to be chosen. For each of the parameters we are varying the values and observing the coherence scores. Coherence scores allow us to arrive at an objective metric for evaluating the effectiveness of the algorithm.
The three parameters were chosen as they are subjectively speaking analogous to how a human would extract topics from any given text - we tend to use phrases, choose the most common words and try to represent as much from the text as possible.

The scores obtained are the averages over 10 runs for each dataset and for each valid value of the parameters chosen. The N-gram range has been varied from (1, 1) to (6, 6). We

chose not to go further as our dataset for ABC news has on average 6.5 words per document. After this we choose the N-gram range with the best coherence scores of around 0.65 for testing the remaining two parameters [(1, 6) for the ABC News Dataset, (1, 6) for the Stackoverflow Dataset and (1, 5) for the Trump Twitter dataset)].

The value for diversity was varied from 0.00 to 1.00 in steps of 0.1 with best the N-gram range for that dataset. The averages were calculated for 10 runs as mentioned. The value for choosing the top *n* words was varied from 1 to 10 in steps of 1. Going above 10 will not be practically useful as the calculations will need more time.

5.2.1 Effect of the N-gram Range

In the domain of computational linguistics, the n-gram is a consecutive chain of n elements from the given text or speech. Depending on the approach used, the n-gram can be syllables, base pairs, words, or phonemes. They are generally extracted from a corpus of spoken or written language. One may also term them as shingles when the elements in them are words.

By using a good n-gram range we are able to drastically improve the quality of classifications obtained as with multiple words we are better able to capture the complexity of a composition of words. BERTopic being a neural topic model which can use the semantic meaning behind the words will perform better if the N-gram Range is wide. This way, the context can be more effectively captured.

In our testing we found that an N gram range gives us the best coherence scores (closest to 0.65). Below is a graph representing our findings.

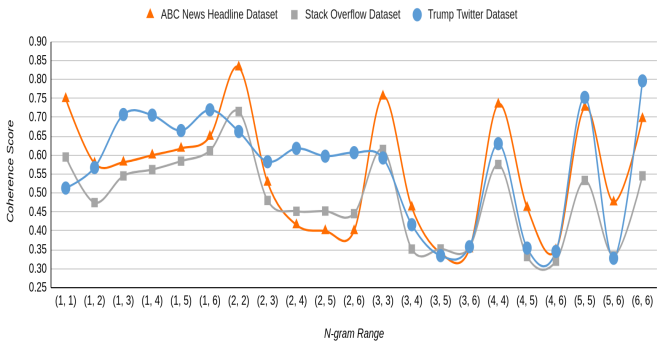

Fig. 4 Effect of N-gram range

In general we can see that for N-grams of the type (x, x) - where the minimum and maximum n-grams are equal, the coherence scores are the highest. They are on average greater than or at least equal to 0.7. This indicates that the model is being overfit. There is a high chance that common phrases are picked up by the algorithm to represent the topics. This is in line with what was subjectively observed.

We can also see that the coherence scores for an n-gram of the range (x, y) decreases dramatically as the values for *y* increase as *x* remains constant. This indicates that as more words are used to represent a topic the semantic similarity between the words and phrases within that topic decreases for the given dataset. Furthermore, it intuitively makes sense that for short text which usually consists of just a few words, representing its topic as the text itself is not very helpful or productive.

We can observe from the graph that for the ABC news dataset and the Stack Overflow datasets an N-gram range of (1, 6) gives the best coherence scores. And for the Trump Twitter dataset an N-gram range of (1, 5) gives the best coherence scores.

5.2.2 Effect of the Diversity/MMR

Maximum Marginal Relevance (MMR) seeks to minimise result duplication while preserving query relevance for already ranked articles, phrases, etc. MMR uses the novelty of the words representing a topic in the ranking of documents and eliminates partially or fully duplicate information. It is a diversity based ranking approach in which sentences are ranked according to their dissimilarity to a set of other sentences - dissimilar sentences will be ranked higher than similar ones. BERTopic uses the cosine similarity to calculate the MMR rank of the phrases/words.

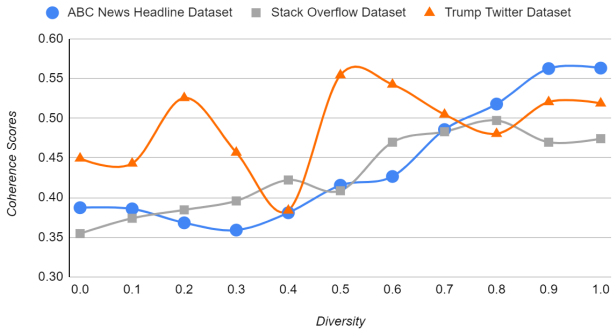

Fig. 5 Effect of Diversity/MMR

From the above figure, one can observe that the coherence score tends to increase with an increase in diversity. For the Trump Twitter Dataset the trend is not clear. But on averaging the scores we find that it too follows a similar trend with respect to diversity. From our observation these values correspond to better topics being generated. An interesting observation is that the coherence scores are lower than those without using MMR to rank the words.

This seems to suggest that diversity of words may have a negative impact on the coherence of the topics generated. This intuitively makes sense as coherence is a measure of similarity among the words of a topic.

5.2.3 Effect of choosing the top `n` words

BERTopic chooses the top `n` words/phrases from each topic based on the c-TF-IDF scores to represent the topic. The c-TF-IDF is calculated as mentioned in section 3.C. By varying this parameter we generate different phrases which could represent the topic of the text by better context and semantics.

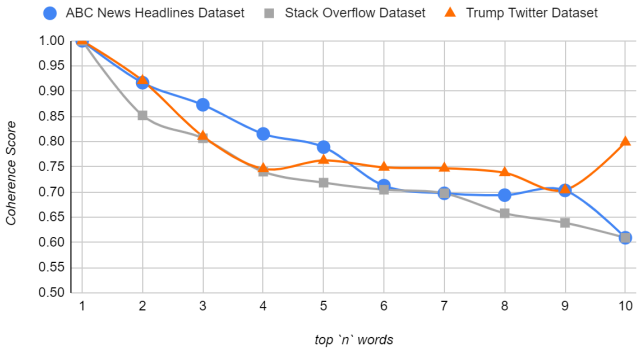

Fig. 6 Effect of Top 'n' words

From the above figure, we observe very high values for coherence (= 1) when BERTopic is set to choose only the best word/phrase for each topic. This is as expected as coherence is a measure of similarity between the words/phrases in a topic. The coherence tends to decrease as the number of words/phrases chosen increases. For values less than or equal to 7, the coherence scores are more than 0.7 indicating overfitting of the model. This reflects what is observed - for these values the topics generated have many phrases which have similar meaning. But as we go higher the coherence score comes close to the best score for coherence. The topics generated are subjectively meaningful phrases from which one can decide the topic reliably which is reflected in the diversity of phrases in the topics generated. We achieved the best scores for coherence when considering the values of 9 and 10 for this parameter.

Using higher values for this parameter decreases the coherence score beyond the acceptable score.

5.3 Comparison based on Topic Coherence

The figure below shows the optimal coherence scores of GSDMM and BERTopic when applied to all three datasets.

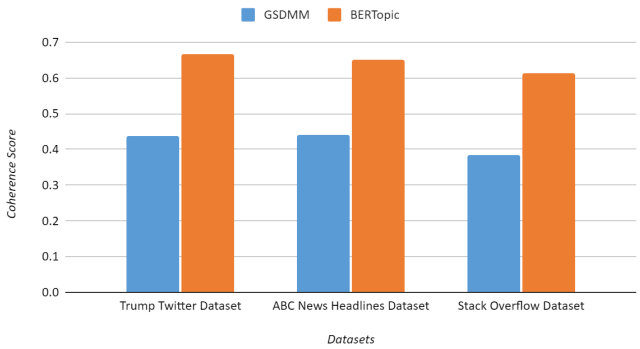

Fig. 7 Comparison of Coherence Scores

With a score of about 0.6, we can observe that BERTopic outperforms GSDMM in terms of coherence on all the three datasets with the latter having scores only around 0.4.

This is because BERTopic leverages language models to produce more coherent topics which also more effectively capture the context of the text. It is able to use pre-trained BERT models to cluster together similar topics.

6. Conclusion

In this paper, we discussed the Dirichlet Multinomial Mixture model for short text topic modelling using the Gibbs Sampling algorithm (GSDMM). The Movie Group Process (MGP) was also considered as an analogy for GSDMM that might clarify the parameters of the model's operation as well as how and why it operates. We discovered that GSDMM is quick to converge and can automatically infer the count of clusters (topics). Short texts present a sparse and high-dimensional challenge that can be handled by GSDMM in order to identify the key words for each cluster. A thorough experimental research demonstrates that GSDMM can perform significantly better than the traditional approaches.

With the help of cutting-edge language models and a class-based TF-IDF technique, the BERTopic, a new topic model leverages the cluster embedding approach to produce topic representations. By separating the processes of classifying documents and constructing subject representations, this paradigm gains a significant amount of flexibility and usability.

In this paper, we give a thorough examination of BERTopic through evaluation experiments that use conventional topic coherence measures. Our research indicates that BERTopic acquires the coherent linguistic patterns present in short text and performs competitively and consistently better than GSDMM.

7. References

[1] Aggarwal, Charu C., and ChengXiang Zhai. "A survey of text clustering algorithms." In Mining text data, pp. 77-128. Springer, Boston, MA, 2012.

[2] Yin, Jianhua, and Jianyong Wang. "A dirichlet multinomial mixture model-based approach for short text clustering." In Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 233-242. 2014.

[3] Grootendorst, Maarten. "BERTopic: Neural topic modeling with a class-based TF-IDF procedure." arXiv preprint arXiv:2203.05794 (2022).

[4] Qiang, Jipeng, Zhenyu Qian, Yun Li, Yunhao Yuan, and Xindong Wu. "Short text topic modeling techniques, applications, and performance: a survey." IEEE Transactions on Knowledge and Data Engineering 34, no. 3 (2020): 1427-1445.

[5] Jónsson, Elıas, and Jake Stolee. "An evaluation of topic modelling techniques for twitter." In Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing (short papers), pp. 489-494. 2015.

[6] Zhao, He, Dinh Phung, Viet Huynh, Yuan Jin, Lan Du, and Wray Buntine. "Topic modelling meets deep neural networks: A survey." arXiv preprint arXiv:2103.00498 (2021).

[7] Berkhin, Pavel. "A survey of clustering data mining techniques." In Grouping multidimensional data, pp. 25-71. Springer, Berlin, Heidelberg, 2006.

[8] Grootendorst, Maarten. "BERTopic: Neural topic modelling with a class-based TF-IDF procedure." arXiv preprint arXiv:2203.05794 (2022).

[9] Angelov, Dimo. "Top2vec: Distributed representations of topics." arXiv preprint arXiv:2008.09470 (2020).

[10] Beyer, Kevin, Jonathan Goldstein, Raghu Ramakrishnan, and Uri Shaft. "When is “nearest neighbor” meaningful?." In International conference on database theory, pp. 217-235. Springer, Berlin, Heidelberg, 1999.

[11] Allaoui, Mebarka, Mohammed Lamine Kherfi, and Abdelhakim Cheriet. "Considerably improving clustering algorithms using UMAP dimensionality reduction technique: a comparative study." In International Conference on Image and Signal Processing, pp. 317-325. Springer, Cham, 2020.

[12] Bianchi, Federico, Silvia Terragni, and Dirk Hovy. "Pre-training is a hot topic: Contextualized document embeddings improve topic coherence." arXiv preprint arXiv:2004.03974 (2020).

[13] Bianchi, Federico, Silvia Terragni, Dirk Hovy, Debora Nozza, and Elisabetta Fersini. "Cross-lingual contextualized topic models with zero-shot learning." arXiv preprint arXiv:2004.07737 (2020).

[14] Bouma, Gerlof. "Normalized (pointwise) mutual information in collocation extraction." Proceedings of GSCL 30 (2009): 31-40.

[15] Carbonell, Jaime, and Jade Goldstein. "The use of MMR, diversity-based reranking for reordering documents and producing summaries." In Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval, pp. 335-336. 1998.

[16] McInnes, Leland, John Healy, and James Melville. "Umap: Uniform manifold approximation and projection for dimension reduction." arXiv preprint arXiv:1802.03426 (2018).

[17] Reimers, Nils, and Iryna Gurevych. "Sentence-bert: Sentence embeddings using siamese bert-networks." arXiv preprint arXiv:1908.10084 (2019).

[18] Sia, Suzanna, Ayush Dalmia, and Sabrina J. Mielke. "Tired of topic models? clusters of pretrained word embeddings make for fast and good topics too!." arXiv preprint arXiv:2004.14914 (2020).

[19] Habibabadi, Sedigheh Khademi, and Pari Delir Haghighi. "Topic modelling for identification of vaccine reactions in twitter." In Proceedings of the Australasian Computer Science Week Multiconference, pp. 1-10. 2019.

[20] Blekanov, Ivan S., Svetlana S. Bodrunova, Nina Zhuravleva, Anna Smoliarova, and Nikita Tarasov. "The ideal topic: Interdependence of topic interpretability and other quality features in topic modelling for short texts." In International Conference on Human-Computer Interaction, pp. 19-26. Springer, Cham, 2020.

[21] Lin, Lihui, Hongyu Jiang, and Yanghui Rao. "Copula guided neural topic modelling for short texts." In Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 1773-1776. 2020.

[22] Wang, Bo, Maria Liakata, Arkaitz Zubiaga, and Rob Procter. "A hierarchical topic modelling approach for tweet clustering." In International Conference on Social Informatics, pp. 378-390. Springer, Cham, 2017.