OTMISC: Our Topic Modeling Is Super Cool!
A Systematic Comparison of Topic Modeling Algorithms on Short and Long Text Datasets

Berk Sudan
Department of Informatics
berk.sudan@tum.de

Ferdinand Kapl
Department of Mathematics
ferdinand.kapl@tum.de

Yuyin Lang
Department of Informatics
yuyin.lang@tum.de

Abstract

Topic modeling helps to classify documents into different topics and choose some words to represent the extracted topics. In this work, we created a topic modeling pipeline to evaluate different topic modeling algorithms, including their performance on short and long text, preprocessed and not preprocessed datasets, and with different embedding models (for embedding-based algorithms). Finally, we summarized the results and suggested how to choose algorithms based on the task. Our code is accessible on GitLab[1].

1 Introduction

Over the last few years, the availability of extensive unstructured data has risen to unseen heights. Especially text data, in the form of tweets, comments, news articles, customer reviews, and job postings, is everywhere, more than ever. This creates the need to be able to analyze these sizeable unstructured text datasets in order to derive insights from them. Topic Modeling is one of the domains that help humans understand these large corpora. Its goal is to cluster documents (in whatever form) into groups that share an underlying topic and then represent or associate these topics by a collection of so-called topic words that should be most descriptive of the topics at hand. The use-cases for Topic Modeling are manifold: online recommender systems trying to rank more relevant articles higher, extracting and mapping the right features of candidates to individual job postings, and last but not least, organizing extensive email collections, customer reviews as well as social media postings (see Basmatkar and Maurya, 2022).

In general, Topic Modeling (in the following abbreviated as TM) algorithms can be divided into two categories: traditional probability-based approaches like Latent Dirichlet Allocation (LDA) (Blei et al., 2003) or Non-negative Matrix Factorization (NMF) (Lee and Seung, 2000) and more modern embedding-based approaches like Top2Vec (Angelov, 2020), BERTopic (Grootendorst, 2022) or Cross-lingual Contextualized Topic Models (CTM) (Bianchi et al., 2021). These probability-based methods have some well-known shortcomings (see Basmatkar and Maurya, 2022 and Grootendorst, 2022): for example, they need a predefined number of topics since they cannot dynamically create an estimate by "looking" at the dataset. Even more detrimental, they operate on bag-of-words inputs, disregard information encoded in the sequence of words, and, consequently, might lose semantic meaning. Newer algorithms try to solve these issues by employing pretrained embedding models. For example, BERTopic mainly uses embeddings from the Sentence-BERT framework (Reimers and Gurevych, 2019) to encode the semantic information contained in the documents and then use the fact or assumption that vectors close in embedding space are semantically similar. Furthermore, hybrid approaches combine both worlds by concatenating LDA with embedding vectors, for example, LDA-BERT from Basmatkar and Maurya (2022).

All available datasets are listed in Section 4, and the algorithms are described in depth in Section 5. The main goal of this work is to give a thorough overview of currently used, state-of-the-art TM algorithms and compare their performance on different short and long text datasets to suggest which approach should be favored for a given dataset.

2 Related Work

Currently, one of the biggest challenges in TM research is the lack of a golden standard for evaluating outputs from different algorithms. Even though all the existing methods work in different

[1] https://gitlab.lrz.de/practical-courses/nlp-lab-course-ss22/topic-modeling-advancements

ways, all of them usually produce two things: An assignment of documents to topics/clusters[2] and a representation of every topic in the form of topic words with associated word scores that are metrics of the importance of that word for that topic. In Angelov (2020), Top2Vec is only compared to the older methods of LDA and PLSA with the metric of *Topic Information Gain* that tries to measure how informative topic words are in their respective documents. The outputs of the aforementioned three algorithms are compared on Yahoo Answers and 20 Newsgroups datasets. Finally, some nonmetric findings are presented: Topics together with their respective topic words and visualizations of the semantic space by using UMAP to reduce the dimension of the embedding space to 2D and coloring the "points" by either real label[3] or assigned topic.

Before we continue, let us first introduce the most popular metrics for evaluating topic models: *Topic Coherence* (abbreviated TC) and *Topic Diversity* (TD). In this regard, various techniques exist for measuring the respective concepts. In general, coherence can be understood, as stated by Hoyle et al. (2021), as "An intangible sense, available to human readers, that a set of terms, when viewed together, enable human recognition of an identifiable category.". On the other hand, topic diversity is simply a measure of how diverse the topic words are, ranging from all topics described by the same words to all being different/unique.

For example, in Basmatkar and Maurya (2022), LDA-BERT is compared to LDA, TF-IDF with K-Means, and BERT with K-Means by using C_V, a measure for topic coherence described in depth later, and silhouette score, a measure how well the clusters are separated, on a single dataset of Amazon product reviews. Furthermore, they only list the topic words of some resulting clusters for LDA-BERT.

Finally, Grootendorst (2022) sets an excellent example of how one can evaluate different algorithms. First, BERTopic is compared to LDA, NMF, Top2Vec, and CTM, thus including newer methods, on three different datasets composed of long text (20 Newsgroups, BBC News) and a short text dataset (Trump's tweets). Then, TC and TD measures are used, i.e., NPMI and percentage of unique topic words, but unfortunately, no visualizations or examples of topic words are given other than a graphic of the computation times of all algorithms.

Our observations on this small subset of current methods are supported by the findings in Hoyle et al. (2021): There exists a *validation gap* since automated coherence measures that were validated in the past on topics produced by classical TM algorithms are not validated yet on the outputs of current (neural) methods. Furthermore, they also recognize a *standardization gap*. This is because most works on TM do not use the same benchmarking data, preprocessing steps, and hyperparameter tuning procedures. To add, experiments by Hoyle et al. (2021) show that human assessment of coherence differs from automatic measures, e.g., NPMI and C_V, in the sense that automatic scores declare a clear winner of the topic model when humans do not. They explain this difference by multiple factors: metric-based evaluation favors esoteric topics by producing high NPMI scores for topic words that are very specific and only occur in a narrow context, and by giving low scores to words that are related (for humans) but do not frequently appear in a small context window (NPMI: 10-word tokens). Therefore it makes sense to prefer C_V over NPMI as a coherence measure as it uses a larger window size; details for that choice are given later. In general, there needs to be a reconsideration of how to evaluate topic models, both by humans and automatic metrics. Possible directions for further research are given in Section 8.

Finally, our solution for evaluating topic models is an exhaustive combination of different topic coherence, diversity, and cluster metrics together with many visualizations which facilitate better comprehension and evaluation of the outputs of topic models by humans. The used metrics consist of the following:

- **Normalized Pointwise Mutual Information:** NPMI is a measure for the coherence of topic words in [-1,1], with 1 being perfect association.

- **$\mathbf{C_V}$:** It is also a coherence measure but uses a larger sliding window (110-word tokens) over the text with indirect cosine similarity based on NPMI; in [0,1] with 1 being perfect association. This metric is motivated by the survey by Röder et al. (2015) of "all" sensible

[2] This can either be deterministic, as in one document belongs to one topic or probabilistic, i.e., there is a distribution over the topics for every document.

[3] The labels were derived from 20 Newsgroups or Yahoo Answers.

combinations of existing and new coherence measures where it achieved the best results in terms of highest correlation with human judgment.

- **Topic Diversity:** A measurement for diversity as a percentage of unique topic words, i.e., in [0,1] with 1 meaning all different topic words.

- **Inverted Rank-Biased Overlap:** Also a measure for diversity as a rank-weighted percentage of unique topic words where words at higher ranks are penalized less. Again in the range of [0,1] with 1 meaning all different topic words.

- **Rand Index:** Similarity measure for the two clusterings given by the topic model and the real labels (details later in Section 4), again in [0,1] with 1 representing a perfect match.

The first four, so all topic coherence and diversity metrics are implemented using the framework OCTIS[4], an open-source python package (Terragni et al., 2021), and the rand index from scikit-learn.[5]

Additionally, depending on the algorithm, as many as five different visualizations are available for the output of a topic model. Details are given in Section 3.

At the end of this section, we briefly mention other works comparing recent topic models. For example, in Egger and Yu (2022), four TM algorithms are compared on a dataset of tweets consisting of covid travel-related hashtags. Namely, they compare the performance of LDA, NMF, Top2Vec, and BERTopic by topic coherence of the best scoring model for the respective algorithm. The following approach is suggested for determining the number of topics for a given dataset. First, choosing an *optimal* number is hard or even impossible because topics frequently overlap, and topics are often mixtures of background "themes". Nevertheless, they recommend choosing a small number to allow for manual (human) inspection of the resulting topics, for example, 10, and then going from there. Finally, their conclusion determines BERTopic as a winner on this short text dataset and lists the strengths as well as weaknesses of the respective algorithms. As far as we know, this work is one of the only ones comparing current TM models with older baselines, other than works on new algorithms such as Grootendorst (2022). Another relevant work for our experiments is Sethia et al. (2022). Many combinations of LDA, Word2Vec, Doc2Vec, and BERT with dimension reduction methods (PCA, t-SNE, Autoencoder) and the clustering algorithm K-Means were compared on the 20 Newsgroups by using Normalized Mutual Information as an evaluation metric. The best performing composition was what we call LDA-BERT, a hybrid approach of LDA together with Sentence BERT followed by an Autoencoder and K-Means. However, they only evaluated LDA-BERT on a subset of one-fourth of the whole dataset due to, as they write, computation time constraints.

3 Architectural Design

Now we want to describe the architecture of our pipeline for running the experiments, whose results are shown later in Section 6. The name of our framework is an homage to OCTIS, and we called it OTMISC: Our Topic Modeling Is Super Cool! and as can be seen in appendix A in figure 1, our framework currently contains eight different datasets, described in Section 4, sixteen preprocessing methods, six TM algorithms and for evaluation five visualizations as well as the five already mentioned automatic metrics. The list of available preprocessing functions can be seen in Appendix C in Table 7 (applied to raw text) and Table 8 (applied to tokenized text). Furthermore, the available visualizations consist of:

- **UMAP 2D Scatter Plot:** Visualizes document embeddings colored by their topic cluster.

- **Topic Words Bar Chart:** Plots the top words of every topic with their word scores.

- **Labels per Topic:** Shows the distribution of real labels of documents (see Section 4) per topic cluster.

- **Topic Similarity Matrix:** Matrix colored by cosine similarity between topic embeddings for similarity of topics in embedding space.

- **Representative Documents:** Colored dataframe with representative documents per topic.

[4] https://github.com/MIND-Lab/OCTIS

[5] https://scikit-learn.org/stable/modules/generated/sklearn.metrics.rand_score.html

The UMAP 2D scatter plot and topic similarity matrix visualization are only available for Top2Vec, BERTopic, and LDA-BERT. Example outputs will be shown later.

4 Datasets

The datasets used here can be divided into two different categories: short and long text datasets. Dividing them into these categories is justified because a dataset with short text has inherently different characteristics than an extended text dataset. For example, tweets from *CRISIS #01* (short for CRISIS NLP - Resource #01) often contain fewer than 30 words and many URLs, hashtags, and typos or abbreviations. In contrast, long text datasets like 20 Newsgroups often have documents with more than 100 words and are more readable and more sophisticated as they are often news articles. All datasets, their categories, number of documents, and number of real labels as given by the sources are shown in Table 1. Their respective sources can be accessed in the footnotes. For example, the 20 Newsgroups dataset is the by-date version[6], Yahoo Answers is a subset of 60k documents from the full dataset[7], the CRISIS NLP - Resources are only the labeled versions[8], and the AG News Titles are the title of various news[9]. Note that, due to its size, the dataset "AG News Titles + Text" , which is a concatenation of news titles and their contents, is ignored in the initial experiments.

Name	Type of Text	# Docs	Topic Count
20 Newsgroups (By Date)	Long	18846	20
Yahoo Answers (60K)	Long	60000	10
AG News Titles + Text	Long	127600	4
CRISIS #01	Short	20514	4
CRISIS #07	Short	10941	2
CRISIS #12	Short	8007	4
CRISIS #17	Short	76484	10
AG News Titles	Short	127600	4

Table 1: Available datasets and their characteristics

For our experiments, we initially explored how unprocessed and preprocessed versions of the same dataset affect the results. Even for embedding-based TM algorithms such as Top2Vec and BERTopic, we observed that a small amount of preprocessing (removal of stop-words and non-informative words) resulted in better topic words. Therefore, to compare different algorithms/datasets fairly, we used similar preprocessing methods in each run.

Moreover, to enable an easier human-based comparison and good visualizations of the outputs, we chose a low number of topics per dataset as given by the number of unique labels in the respective datasets (see Table 1).

5 Algorithms

In this section, we introduce six algorithms in our tool in more detail. Here, we present how the algorithms work, what parameters we use, and provide visualization examples for the algorithms.

5.1 LDA and NMF

First, we introduce LDA and NMF together. As the only probability-based approaches in this work, they have in common that they both use the BOW (Bag of Words) model. More specifically, LDA iterates two probabilities – P(topic t | document d) and P(word w | topic t) through words to obtain topic words and topic documents. On the other hand, NMF uses matrix factorization to convert a high-dimensional representation into the product of two low-dimensional representations to achieve the same goal.

We use OCTIS for the implementation. We set the parameter alpha for LDA as asymmetric; for NMF, we use the default settings in OCTIS. We predefine the topic numbers for both algorithms as the topic count in Table 1 and set a random state for them for reproducibility. We show the labels per topic visualization of NMF on Crisis 12 dataset in Appendix B.1.

5.2 BERTopic

Next, we briefly describe how BERTopic creates topics and topic representations in the form of topic words. The algorithm can be split into three phases. In *phase 1*, embeddings are created for each document in the data by leveraging a pretrained embedding model. By default, any model from the Sentence BERT framework is supported (Reimers and Gurevych, 2019) but generally, which is one of the biggest strengths of BERTopic, any pretrained language model that was fine-tuned on semantic similarity can be used by passing it the calculated embeddings. Then in *phase 2*, these embeddings are reduced in dimension by UMAP (McInnes et al.,

[6] http://qwone.com/~jason/20Newsgroups

[7] https://github.com/LC-John/Yahoo-Answers-Topic-Classification-Dataset

[8] https://crisisnlp.qcri.org/

[9] https://huggingface.co/datasets/ag_news

2018), and then the lower-dimensional representations are clustered into topics by a hierarchical clustering algorithm with a soft-clustering approach HDBSCAN (Malzer and Baum, 2020). This clustering method allows documents to be classified as noise which might improve the generation of concise topics. However, in practice, we found that when using a low number of topics[10], a large portion of the files tends to be classified as noise, up to 60% of the full dataset. Here each document is assigned to one topic (cluster), so if one wants to assign every document to a "real", non-noise cluster, our experiments showed that the most simple and elegant solution is to use K-Means instead of HDBSCAN, which of course requires us to specify the number of desired topics, also producing good results. Other options like changing the parameters of the HDBSCAN algorithm and assigning documents to the most likely non-noise cluster when the probability is more significant than a specified threshold came with additional problems and performance drops. Finally, in *phase 3*, topic words per cluster are extracted by creating topic-word distributions with a class-based TF-IDF approach and taking the top n words with the highest probability (here: ten words per topic). A topic similarity matrix visualization example can be found in Appendix B.2.

5.3 LDA-BERT

As previously mentioned, LDA-BERT is a hybrid approach combining a classical TM algorithm with embeddings. First, for every document, LDA generates a vector with the probabilities of this document belonging to a topic, so if we choose several topics equal to 10, this vector has dimension 10 for every document. Then the LDA vector is concatenated with the embedding vector; we use an embedding model again from the Sentence BERT framework, using a hyperparameter gamma weighting the respective importance of the two vectors. Next, an autoencoder produces a latent space representation of this concatenated vector which is finally clustered using K-Means, and the topic words for each cluster are simply the most frequent words per cluster. Therefore this method and our implementation benefit highly from preprocessing.[11]. Appendix B.5 illustrates how representative documents of the detected topics look like after running LDA-BERT.

5.4 Top2Vec

There are several similarities between Top2Vec and BERTopic. They both use pretrained embedding models to create document and word embeddings. Embedding dimensions are then reduced by UMAP and clustered by HDBSCAN. Unlike BERTopic, Top2Vec adopts a slightly different approach to generating topic words, assigning all documents to a topic. Therefore, Top2Vec outputs do not contain *noisy* documents which are not assigned to a topic. Moreover, Top2Vec deals with the outlier documents problem by simply not taking them into account while computing topic vectors, although still assigning them to the closest topic vectors afterward. Outlier documents can be observed by the naked eye, thanks to the UMAP-2D visualization method. An example UMAP-2D output can be found in Appendix B.4. In addition, Angelov (2020) concludes that Top2Vec consistently finds more informative topics and representative of the corpus than probabilistic generative models like LDA and PLSA for varying sizes of topics and number of top topic words. We strongly agree with this conclusion since we also noticed a similar pattern in our experiments.

We used the library from the original author of the Top2Vec paper with some adjustments and improvements[12]. It was suggested that some preprocessing methods, namely stop-words removal and lemmatization, are noneffective as creating word embeddings can overcome these issues (Angelov, 2020). However, after applying the aforementioned preprocessing methods, we observed minor improvements in the results, especially for short text tweet datasets. We reduced the number of detected topics to the number of real topics using *hierarchical topic reduction*. Top2Vec and BERTopic do not need the number of topics to be provided in advance, although we did to make a fair comparison according to the evaluation metrics. It also helped us to see whether each detected topic could be matched with a different real label. We argue that the main disadvantage of hierarchical topic reduction is the requirement of a *balance* in the number of documents per topic because there is a higher chance for the topics of similar sizes to merge. If the dataset is highly unbalanced, this can

[10] here, “low” means in comparison to the number of topics found by the algorithm before reducing it to a given number of topics

[11] Our implementation is adapted from `www.kaggle.com/code/dskswu/topic-modeling-bert-lda`

[12] `https://github.com/ddangelov/Top2Vec`

lead to the merging of unrelated documents.

5.5 CTM

As the newest algorithm in our task, Contextualized Topic Models (CTM) also uses embedding models. This algorithm is based on variational autoencoders. Instead of inputting BOW, CTM uses pretrained multilingual representations from SBERT, thus taking the context information into account. In this way, CTM overcomes the biggest drawback of BOW representation: BOW has no information about the context.

We use OCTIS for the implementation of CTM. In this work, we tried three different embedding models for CTM (“bert-base-nli-mean-tokens”, “all-MiniLM-L6-v2”, “paraphrase-multilingual-MiniLM-L12-v2”). Since CTM computations take much time, we did not use preprocessing to save time. In Appendix B.3, we present the topic words bar chart visualization to show its performance.

6 Comparison

In this section, we report and compare the results of our algorithms, including performance results on short and long datasets, performance results on preprocessed and unprocessed datasets, and impacts of different embedding models. We further discuss the result in Section 7.

6.1 Performance on Short Datasets

First, we compare the performance of different algorithms on short datasets. As stated before, short datasets in our task include four crisis datasets and one news dataset (AG News Titles). Table 2 shows the results of our algorithms on all five evaluation metrics. The scores are averaged over all five datasets. The best-performing algorithm is marked in bold. Moreover, the number in the parenthesis indicates the ranking of each algorithm.

	TD	TD	TC	TC	Cluster
Algorithm	**Unique**	**Inv. RBO**	**NPMI**	**C_V**	**Rand**
NMF	0.718(5)	0.715(6)	-0.014(4)	0.388(5)	0.614(5)
LDA	0.700(6)	0.810(4)	0.010(2)	0.435(3)	0.625(4)
LDA-BERT	0.743(4)	0.778(5)	-0.001(3)	0.421(4)	0.727(3)
BERTopic	0.826(3)	0.901(3)	**0.050(1)**	**0.493(1)**	0.474(6)
Top2Vec	0.897(2)	0.956(2)	-0.279(6)	0.388(5)	0.739(2)
CTM	**0.966(1)**	**0.994(1)**	-0.100(5)	0.486(2)	**0.746(1)**

Table 2: Performance on short datasets

As Table 2 suggests, CTM outperforms other algorithms in TD and Cluster Rand, while BERTopic achieves the best score in TC. It is also interesting to note that CTM obtains a good result in TC (C_V), and the gap with BERTopic is not much.

6.2 Performance on Long Datasets

The comparison on long datasets is quite similar to the short ones. For example, we have the 20 Newsgroups dataset and Yahoo Answers dataset; the result is the average score of these two datasets. The number in the parenthesis indicates the ranking of each algorithm. The result is shown in Table 3.

	TD	TD	TC	TC	Cluster
Algorithm	**Unique**	**Inv. RBO**	**NPMI**	**C_V**	**Rand**
NMF	0.565(3)	0.805(5)	0.035(4)	0.508(4)	0.846(4)
LDA	0.525(5)	0.878(3)	0.038(3)	0.538(3)	0.792(5)
LDA-BERT	0.397(6)	0.661(6)	0.055(2)	**0.594(1)**	0.880(2)
BERTopic	0.637(2)	0.828(4)	**0.081(1)**	0.577(2)	0.425(6)
Top2Vec	**0.819(1)**	0.902(2)	-0.113(6)	0.436(5)	**0.906(1)**
CTM	0.552(4)	**0.909(1)**	-0.056(5)	0.404(6)	0.861(3)

Table 3: Performance on long datasets

The situation for long datasets is much more complicated than for short datasets. A total of four algorithms received first place in at least one evaluation metric. Top2Vec wins first place twice: In TD (Unique) and Cluster Rand; it also wins second place in TD (Inv. RBO) with a small gap to first place. As for the two metrics in TC, LDA-BERT and BERTopic each receive first and second place, outperforming other algorithms.

6.3 Performance on Preprocessed and Unprocessed Datasets

The preprocessing of datasets strongly influences the performance of the algorithms. For LDA and NMF, because of the BOW representation, preprocessing is essential. Otherwise, the model assigns immense importance to those words which appear quite often in all documents, including stop words and hashtags. Even for embedding-based methods in our work, if we do not apply preprocessing methods, the extracted topic words would contain too many stop words. Therefore, it is interesting to see how preprocessing influences performance based on various evaluation metrics. Here, we choose one probability-based algorithm (NMF) and one embedding-based algorithm (BERTopic). To keep the table not too big, we present the performance on 20 Newsgroups dataset (a long dataset) and CRISIS #01 (a short dataset) on the five evaluation metrics. The result is shown in Table 4 and Table 5. "pre" in the parenthesis means the result is obtained on the preprocessed dataset.

	TD	TD	TC	TC	Cluster
Dataset	**Unique**	**Inv. RBO**	**NPMI**	**C_V**	**Rand**
20 Newsgroups (pre)	0.640	0.960	0.103	0.646	0.736
20 Newsgroups	0.370	0.822	0.023	0.470	0.836
Crisis #01 (pre)	0.825	0.878	0.089	0.492	0.522
Crisis #01	0.600	0.740	-0.008	0.388	0.586

Table 4: Performance on preprocessed and not preprocessed datasets (NMF)

	TD	TD	TC	TC	Cluster
Dataset	**Unique**	**Inv. RBO**	**NPMI**	**C_V**	**Rand**
20 Newsgroups (pre)	0.837	0.981	0.173	0.778	0.630
20 Newsgroups	0.430	0.670	0.017	0.513	0.508
Crisis #01 (pre)	0.900	0.929	0.135	0.582	0.473
Crisis #01	0.742	0.846	-0.090	0.334	0.506

Table 5: Performance on preprocessed and not preprocessed datasets (BERTopic)

The scores in almost all the evaluation metrics have greatly improved. Only the score for Cluster Rand with NMF has dropped. The most significant gap appears in TD (Diversity) on 20 Newsgroups with BERTopic, which amounts to 0.397.

6.4 Performance of Embedding-based Algorithm using Different Embedding Models

Different embedding models vary in many aspects, such as embedding dimension, model size, data on which the model is pretrained, and so forth. In this work, we want to see whether the models perform differently when using different embedding models. In Table 6, we present the performance of Top2Vec on four different embedding models and evaluate it on our five evaluation metrics. The number in the parenthesis indicates the ranking of each algorithm.

	TD	TD	TC	TC	Cluster
Embedding model	**Unique**	**Inv. RBO**	**NPMI**	**C_V**	**Rand**
all-MiniLM-L6-v2	0.907(3)	0.957(2)	-0.284(4)	0.388(2)	**0.864(1)**
doc2vec	0.912(2)	0.931(4)	-0.276(3)	**0.485(1)**	0.552(4)
paraphrase-multilingual -MiniLM-L12-v2	**0.930(1)**	**0.987(1)**	**-0.252(1)**	0.376(3)	0.777(3)
universal-sentence-encoder	0.860(4)	0.941(3)	-0.267(2)	0.366(4)	0.827(2)

Table 6: Performance with different embedding models

As the biggest embedding model in our work, "paraphrase-multilingual-MiniLM-L12-v2" received first place three times in the evaluation, including both TD evaluations. In TC, it performs similarly with doc2vec as both embedding models received first and third place. Finally, the embedding model "all-MiniLM-L6-v2" performs the best in Cluster Rand.

7 Discussion

In this section, we discuss the results shown in Section 6. Then, we draw conclusions based on our experiments and provide insight into TM.

7.1 Can we suggest a golden algorithm for TM?

As can be seen in Table 2 and Table 3, no algorithm outperforms others in all five evaluation metrics. This applies to both short and long datasets. Therefore, it is impossible to claim which algorithm is the best for TM. However, this does not mean we cannot give some suggestions about choosing proper algorithms because the choice of algorithms heavily depends on the specific task. For example, in some tasks, we expect the topic words to be more diverse; in others, we expect the coherence of topic words to be higher. Hence, it makes sense to suggest an algorithm that serves the purpose better for the given task.

For short datasets, CTM would be the first choice when TD and Cluster Rand are crucial for the task; but BERTopic should be considered first when we want to obtain a high TC score. Moreover, as the time complexity of CTM is high, Top2Vec is also a good choice if the computation resources are limited and achieving high TD/Cluster scores is the primary goal.

For long datasets, we encourage to use Top2Vec when it comes to TD. That is because compared to CTM, Top2Vec received a slightly lower score than CTM on TD (Inv. RBO), ranking second, but a much higher score on TD (Unique). Also, the high ranking in Cluster Rand made Top2Vec more comprehensive on long datasets. On the other hand, in the TC metric, we recommend both LDA-BERT and BERTopic as their performances are similar. Furthermore, we noticed that in the four embedding-based algorithms, if one algorithm performs exceptionally well in TC or TD, it is likely to perform poorly in another evaluation metric, which should be considered when choosing the algorithm.

7.2 Should we use preprocessed data?

Data preprocessing should not be avoided for probability-based algorithms like LDA and NMF because such algorithms otherwise cannot read the context. Therefore, a considerable boost in performance is expected when using these algorithms. However, it is surprising that such a boost can also be seen when feeding preprocessed datasets to embedding-based algorithms such as BERTopic. Why does that happen? We argue that although embedding models help better understand the documents (this can be seen in Table 2 and Table 3 as embedding-based algorithms perform much better than probability-based ones), they have no information about which words are representative. That is to say, just like probability-based algorithms, embedding-based algorithms assign high scores to stop words. Therefore, the strength of embedding models lies in obtaining the association between the topic words. Therefore, we strongly recommend using preprocessing methods when doing TM tasks.

7.3 Does it make much difference to use different embedding models?

Usually, the bigger the model size is, the more information a model can extract. However, a bigger model size does not necessarily produce a higher score. Moreover, it can be seen that in the three evaluation metrics where "paraphrase-multilingual-MiniLM-L12-v2" wins the first place, its score was not much higher than the last place. Therefore, embedding models do not influence much to the result, but only slightly.

One should also remember that bigger models will lead to longer computation time. In our work, "all-MiniLM-L6-v2" and "doc2vec" are much faster to compute. Since their performances are not bad, it is worth considering them if the computation resources are limited.

8 Future Work

Two areas of TM that need to be improved the most are topic representations and evaluation of them or topic models in general. The current state of topic representations is using topic words, and as already highlighted in Hoyle et al. (2021), there is a need for something better. Humans usually describe a topic with a short paragraph, often including examples, so a simple list of (topic) words is just not enough to comprehend a topic in a collection of documents. Furthermore, the evaluation of topic models is in dire need of new metrics or improved human-based scoring to better compare existing and upcoming algorithms, especially for neural, non-classical methods.

One of the most exciting avenues of further research is the composition of already used methods. Many newer approaches consist of the following building blocks: a pretrained language model for the embedding, a dimension reduction, and clustering algorithm, and a method to extract topic representations of the existing clusters. For example, further experiments on LDA-BERT are of interest: combining different representations of documents, embeddings, or probabilistic vectors, with newer dimensions and clustering solutions (UMAP & HDBSCAN) with improved topic word representations (class-based TF-IDF). Similar experiments for newly developed dimension reduction and cluster algorithms are imaginable, including building full pipelines to fine-tune a given dataset.

Finally, we would like to look more into the noise clusters BERTopic produces: Are they helpful by letting the existing topics be more concise and descriptive, or are they just an obstacle in understanding a collection of documents?

9 Conclusion

TM has been more and more applied in our daily life, and the need to extract topics from unstructured data is also increasing daily. In our work, we created a pipeline for TM, including choosing datasets, algorithms, and evaluation metrics. Apart from that, we also realized five visualization methods to help humans evaluate the model performance intuitively.

In our experiments, we found that no algorithm beats others in all the evaluation metrics; thus, a golden algorithm does not exist. However, one can choose the best algorithm according to the task. Moreover, even though embedding-based models perform better than probability-based ones, using preprocessed data as input is still beneficial. Finally, different embedding models can bring some performance differences, but not much. Here, the longer computation time caused by a bigger embedding model should be considered.

Acknowledgements

We want to thank Miriam Anschütz and Ahmed Mosharafa for their assistance and valuable feedback at every project stage.

References

Dimo Angelov. 2020. Top2vec: Distributed representations of topics.

Pranjali Basmatkar and Mahesh Maurya. 2022. An overview of contextual topic modeling using bidirectional encoder representations from transformers. In *Proceedings of Third International Conference on Communication, Computing and Electronics Systems*, pages 489–504, Singapore. Springer Singapore.

Federico Bianchi, Silvia Terragni, Dirk Hovy, Debora Nozza, and Elisabetta Fersini. 2021. Cross-lingual contextualized topic models with zero-shot learning. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1676–1683, Online. Association for Computational Linguistics.

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

Roman Egger and Joanne Yu. 2022. A topic modeling comparison between lda, nmf, top2vec, and bertopic to demystify twitter posts. *Frontiers in Sociology*, 7.

Maarten Grootendorst. 2022. Bertopic: Neural topic modeling with a class-based tf-idf procedure.

Alexander Hoyle, Pranav Goel, Denis Peskov, Andrew Hian-Cheong, Jordan Boyd-Graber, and Philip Resnik. 2021. Is automated topic model evaluation broken?: The incoherence of coherence.

Daniel Lee and H Sebastian Seung. 2000. Algorithms for non-negative matrix factorization. *Advances in neural information processing systems*, 13.

Claudia Malzer and Marcus Baum. 2020. A hybrid approach to hierarchical density-based cluster selection. In *2020 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*. IEEE.

Leland McInnes, John Healy, and James Melville. 2018. Umap: Uniform manifold approximation and projection for dimension reduction.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks.

Michael Röder, Andreas Both, and Alexander Hinneburg. 2015. Exploring the space of topic coherence measures. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*, WSDM '15, page 399–408, New York, NY, USA. Association for Computing Machinery.

Kashi Sethia, Madhur Saxena, Mukul Goyal, and R.K. Yadav. 2022. Framework for topic modeling using bert, lda and k-means. In *2022 2nd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE)*, pages 2204–2208.

Silvia Terragni, Elisabetta Fersini, Bruno Giovanni Galuzzi, Pietro Tropeano, and Antonio Candelieri. 2021. OCTIS: Comparing and optimizing topic models is simple! In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 263–270, Online. Association for Computational Linguistics.

A Appendix - Architectural Design

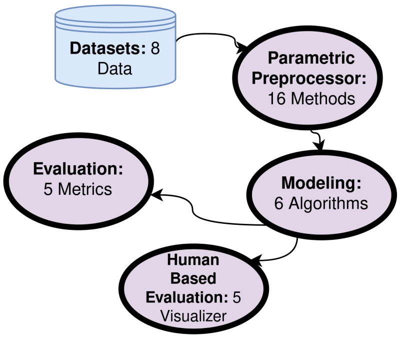


Figure 1: Architectural Design of OTMISC Pipeline

B Appendix - Visualizations

In this section, we present five different visualization methods with examples.

B.1 Labels Per Topic Visualization

This visualization (Figure 2) shows the performance of NMF in grouping the documents that belong to the same real label into the same detected topic. For example, in topic 1, most of the documents assigned to this topic have the real label *earthquake*. We conclude that the model performs better when different real labels dominate each detected topic. As it can be seen, the distinction here is primarily obvious, while there are some *mixed* topics, such as topic 2.

B.2 Topic Similarity Matrix

Figure 3 shows the topic similarity matrix for BERTopic with 20 Newsgroups. Only five topics are shown here as a subset of 20 to avoid too big a figure. The numbers in the matrix indicate the cosine similarity of different topics. Low values indicate that the two topics are irrelevant, while high values suggest the opposite. For example, topic five and topic twelve can be considered similar topics in politics.

B.3 Top Topic Words Bar Chart

This visualization (Figure 4) shows the performance of CTM on CRISIS #12 about the topic words. It is clear in the figure that the topic words belong to four different topics: Wildfire, cyclone, flood, and earthquake.

B.4 UMAP 2D Scatter Plot

A UMAP-2D visualization output can be seen in Figure 5 for Top2Vec with the dataset Crisis #12. The outlier documents are marked with white circles, and each color represents a different detected topic (not real label). Since it directly uses document embeddings, this visualization method is only available for embedding-based algorithms (Top2Vec, BERTopic, and LDA-BERT). Leveraging the UMAP algorithm reduces the number of document dimensions to 2 and visualizes the resulting vectors in a 2D coordinate system.

B.5 Representative Documents

Figure 6 shows the performance of LDA-BERT on the CRISIS #12 dataset. It is convenient to see the sentences (or documents) that obtain the highest scores; thus, this visualization is crucial for human-based evaluation.

C Appendix - Preprocessing Functions

Here, we show the preprocessing functions used in our work. Table 7 shows the functions for raw texts. These functions could be used on the entire sentence. Table 8 shows the functions for tokenized texts, which means these functions are specific for words, for example, deleting or changing the form of a word.

The preferred methods in our work are "to_lowercase", "standardize_accented_chars", "remove_url", "expand_contractions", "remove_mentions", "remove_hashtags", "keep_only_alphabet", "remove_english_stop_words" and "lemmatize_noun".

Function	Description
to_lowercase	text to lowercase
standardize_accented_chars	standard representation for accented characters
remove_url	remove www and http(s) links
expand_missing_delimiter	insert space between a lower- and uppercase letter
remove_mentions	remove @user
remove_hashtags	remove #hashtag
keep_only_alphabet	keep only letters from a-z
remove_new_lines	remove newline characters
remove_extra_spaces	delete additional spaces
remove_html_tags	remove all <...> tags

Table 7: Available preprocessing functions applicable to non-tokenized text

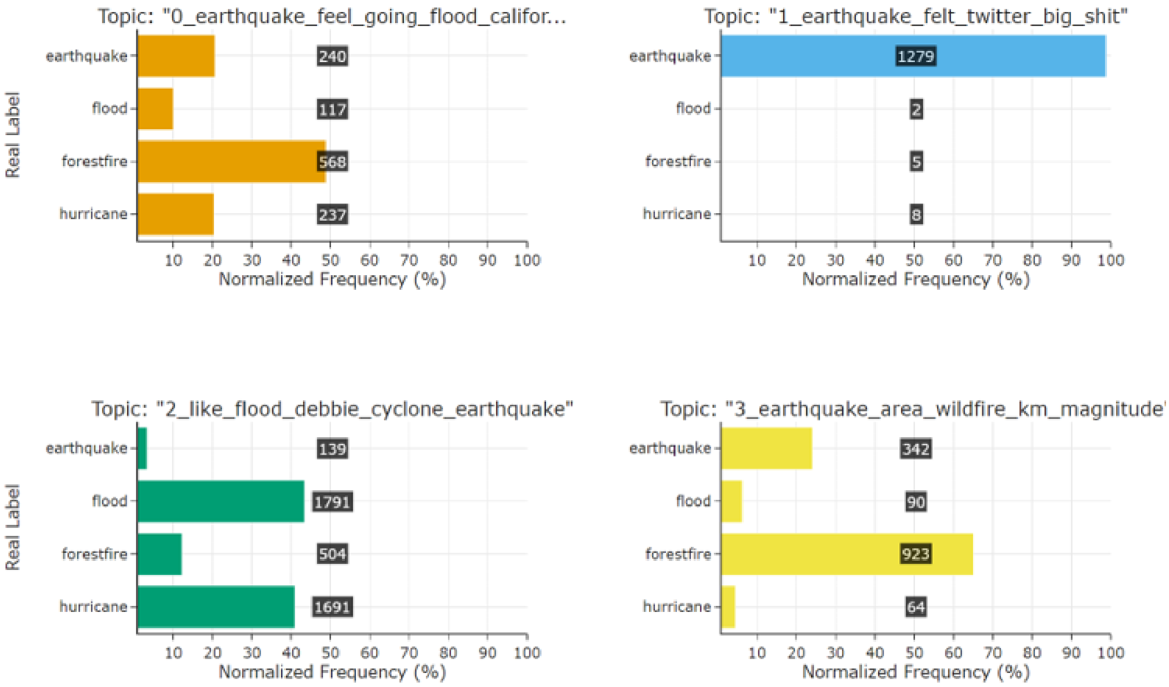


Figure 2: Labels per Topic Visualization for NMF

Function	Description
expand_contractions	example: do not, I will, ...
remove_english_stop_words	example: I, is, and, ...
lemmatize_noun, lemmatize_verb, lemmatize_adjective	lemmatize words with respective part of speech tag
correct_typo	substitutes words with detected typo correction

Table 8: Available preprocessing functions applicable to tokenized text

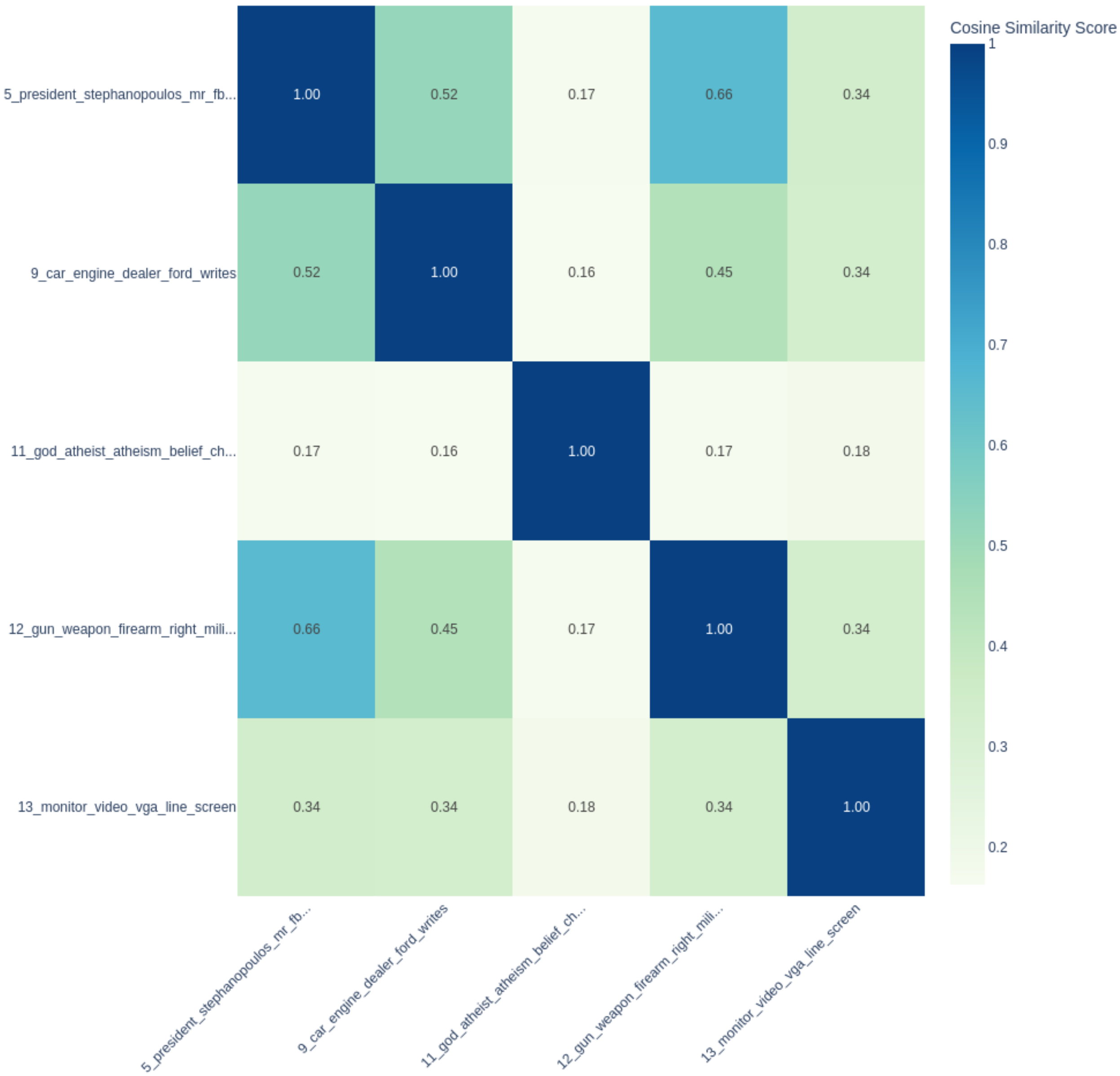


Figure 3: Topic Similarity Matrix Visualization for BERTopic

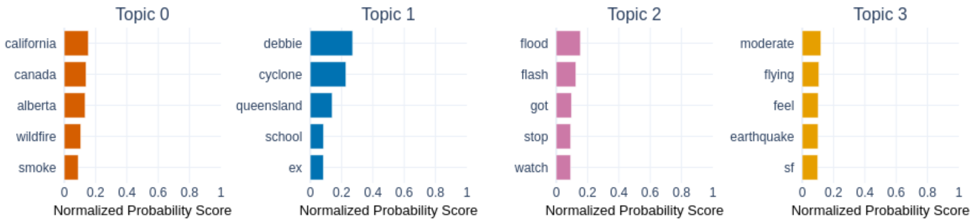


Figure 4: Top Topic Words Visualization for CTM

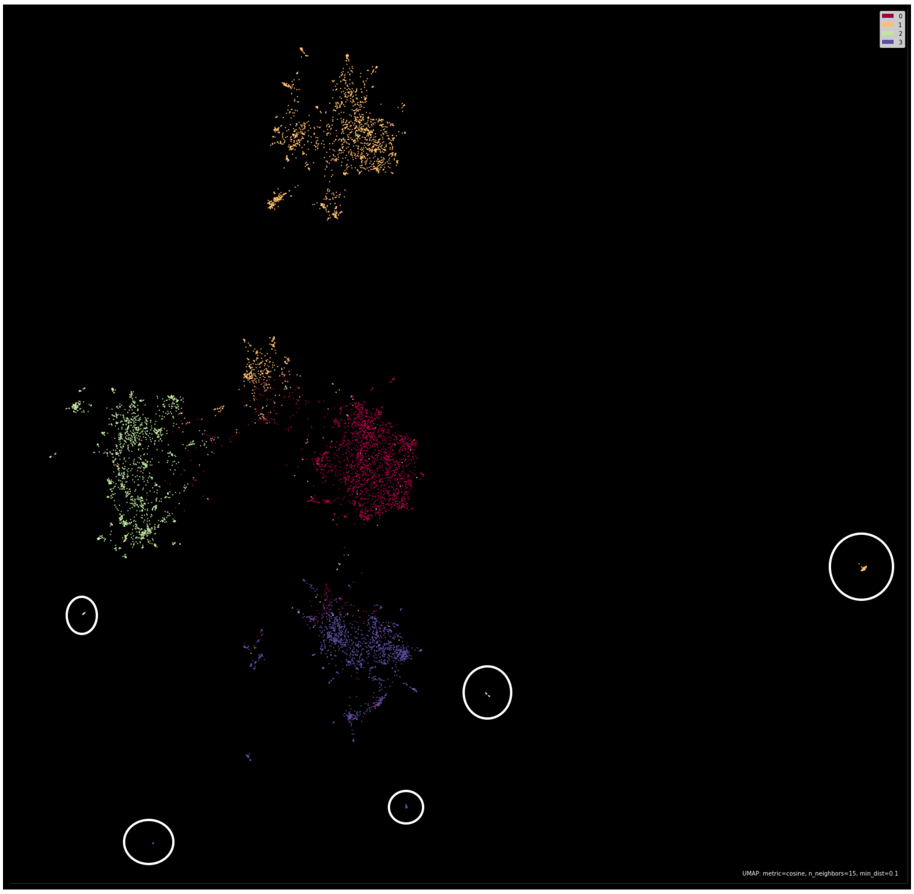
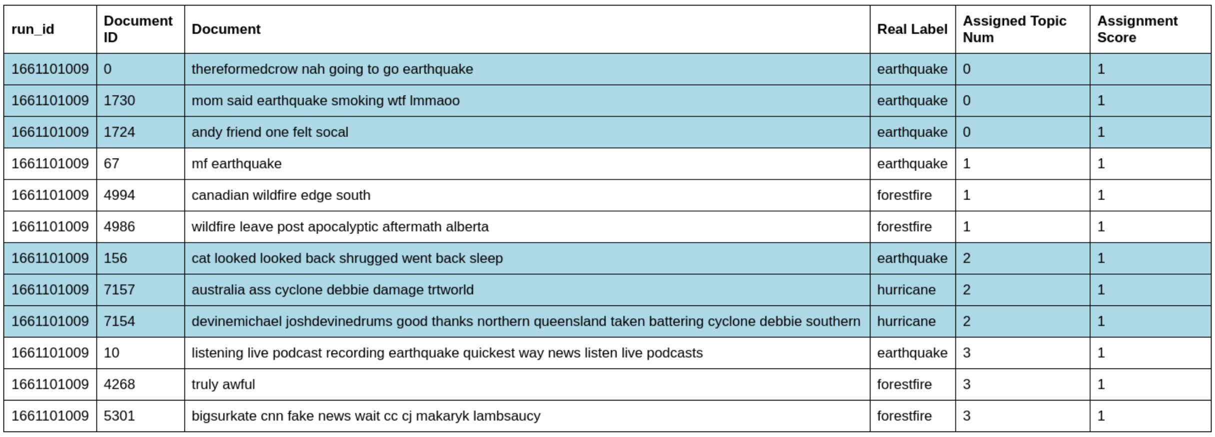


Figure 5: UMAP 2D Scatter Plot Visualization for Top2Vec



run_id	Document ID	Document	Real Label	Assigned Topic Num	Assignment Score
1661101009	0	thereformedcrow nah going to go earthquake	earthquake	0	1
1661101009	1730	mom said earthquake smoking wtf lmmaoo	earthquake	0	1
1661101009	1724	andy friend one felt socal	earthquake	0	1
1661101009	67	mf earthquake	earthquake	1	1
1661101009	4994	canadian wildfire edge south	forestfire	1	1
1661101009	4986	wildfire leave post apocalyptic aftermath alberta	forestfire	1	1
1661101009	156	cat looked looked back shrugged went back sleep	earthquake	2	1
1661101009	7157	australia ass cyclone debbie damage trtworld	hurricane	2	1
1661101009	7154	devinemichael joshdevinedrums good thanks northern queensland taken battering cyclone debbie southern	hurricane	2	1
1661101009	10	listening live podcast recording earthquake quickest way news listen live podcasts	earthquake	3	1
1661101009	4268	truly awful	forestfire	3	1
1661101009	5301	bigsurkate cnn fake news wait cc cj makaryk lambsaucy	forestfire	3	1

Figure 6: Representative Documents per Topic Visualization for LDA-BERT