Topic Modeling: Contextual Token Embeddings Are All You Need

Dimo Angelov
University of Ottawa
dimo.angelov@gmail.com

Diana Inkpen
University of Ottawa
diana.inkpen@uottawa.ca

Abstract

The goal of topic modeling is to find meaningful topics that capture the information present in a collection of documents. The main challenges of topic modeling are finding the optimal number of topics, labeling the topics, segmenting documents by topic, and evaluating topic model performance. Current neural approaches have tackled some of these problems but none have been able to solve all of them. We introduce a novel topic modeling approach, Contextual-Top2Vec, which uses document contextual token embeddings, it creates hierarchical topics, finds topic spans within documents and labels topics with phrases rather than just words. We propose the use of BERTScore to evaluate topic coherence and to evaluate how informative topics are of the underlying documents. Our model outperforms the current state-of-the-art models on a comprehensive set of topic model evaluation metrics.

1 Introduction

The most widely used topic modeling approach is Latent Dirichlet Allocation (LDA) (Jelodar et al., 2019), which is a probabilistic generative model that models documents as a mixture of topics and each topic as a mixture of words. Selecting the optimal number of topics is one of the primary challenges with LDA and many other topic modeling approaches. Additionally, LDA uses bag-of-words (BOW) representations of documents that ignore word semantics and syntax. LDA models the underlying word distribution of documents, which necessarily makes uninformative words the most probable in topics, leading to poor topic interpretability. The authors of the LDA paper make it clear: "We refer to the latent multinomial variables in the LDA model as topics, so as to exploit text-oriented intuitions, but we make no epistemological claims regarding these latent variables beyond their utility in representing probability distributions on sets of words." (Jelodar et al., 2019).

Neural topic models use deep learning techniques to capture the syntax and semantics of text by leveraging word and document embeddings instead of relying on statistical inference on BOW representations.

The Embedded Topic Model (ETM) (Dieng et al., 2020) combined the power of LDA with word embeddings. This approach overcomes the limitations of BOW representation of documents and allows for richer semantic representation of words in a document. However, ETM still ignores the syntax of words and requires the number of topics to be determined.

The Top2Vec model (Angelov, 2020) introduced a novel method that leverages joint document and word embeddings to find topic vectors. It automatically finds the number of topics, it does not require stop-word removal and it also produces hierarchical topics. This approach has been shown to produce more informative and interpretable topics (Karas et al., 2022), (Egger and Yu, 2022), (AKBAY, 2022). Top2Vec captures word syntax with document embeddings and word semantics with word embeddings. Its main limitation is that each document is assigned only one topic.

The BERTopic model (Grootendorst, 2022) uses BERT (Devlin et al., 2019) document embeddings to find clusters of documents that are labeled with class-based TF-IDF. The main limitations of this model are that it uses BOW representation of document clusters which ignore word semantics, it does not assign topics to all documents, and it only assigns one topic per document.

Contextual Embedding Models (CTM) (Bianchi et al., 2021a), (Bianchi et al., 2021b) extend Neural Product-of-Experts LDA (ProdLDA) (Srivastava and Sutton, 2017) by adding contextualized document embeddings. This model uses vector representation of words and documents so it therefore captures word semantics and syntax. Its main limitation is that it requires the number of topics to be

Findings of the Association for Computational Linguistics: EMNLP 2024, pages 13528–13539
November 12-16, 2024 ©2024 Association for Computational Linguistics

determined.

The main advantage common to all the models discussed is that they improve topic coherence compared to LDA. The main disadvantage is that they do not segment topics within a document; instead, they only give distributions of topics present in a document, with the exception of Top2Vec and BERTopic which only assign one topic per document. Another common disadvantage is that a topic distribution does not give information about the relevance of documents to a topic, it only gives the proportion of the document that is relevant.

Most topic evaluation metrics measure topic coherence, but little attention is given to how well the topics represent the underlying documents, which can lead to misleading results (Bhatia et al., 2017). Accessing how well a topic represents the underlying documents is essential to evaluate the performance of a topic model (Doogan and Buntine, 2021). Lastly, coherence measures designed for older models may not accurately represent the performance of novel neural methods (Doogan and Buntine, 2021), (Hoyle et al., 2021).

For additional information on related work see appendix A.

Contributions We propose a new topic modeling approach, Contextual-Top2Vec (C-Top2Vec), which automatically finds the number of topics in the embedding space of document contextual token embeddings and shows the locations of the topics in the document. Our approach supports hierarchical topic reduction which aids in exploring the optimal topic granularity for downstream use-cases.

Previous methods do not specify where a topic occurs in a document or how relevant a document section is to the topic, they only provide the proportion of a document that belongs to a topic. Our approach segments each document into topic spans allowing for much more granular topic discovery. It also produces a per-topic relevance score for each topic span within a document which allows for ranking the relevance of document segments for a topic.

Previous models use ranked lists of words to label topics which can be difficult to interpret. We label our topics with phrases that lead to improved interpretability and topic coherence.

Most previous evaluations focus on topic coherence without evaluating how well topics represent the documents specifically assigned to a topic, leaving a gap in topic model evaluation. We propose using BERTScore (Zhang et al., 2020) to evaluate the coherence of topics and simultaneously evaluate how representative topics are of the topic documents.

Our implementation will be available as a Python library at the following link: https://github.com/ddangelov/Top2Vec

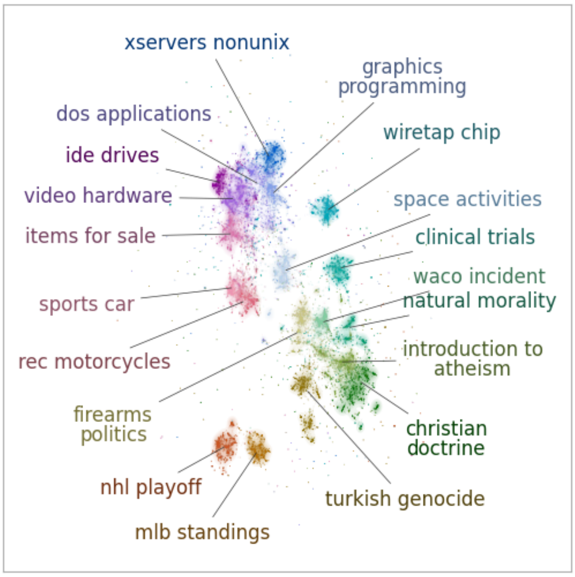


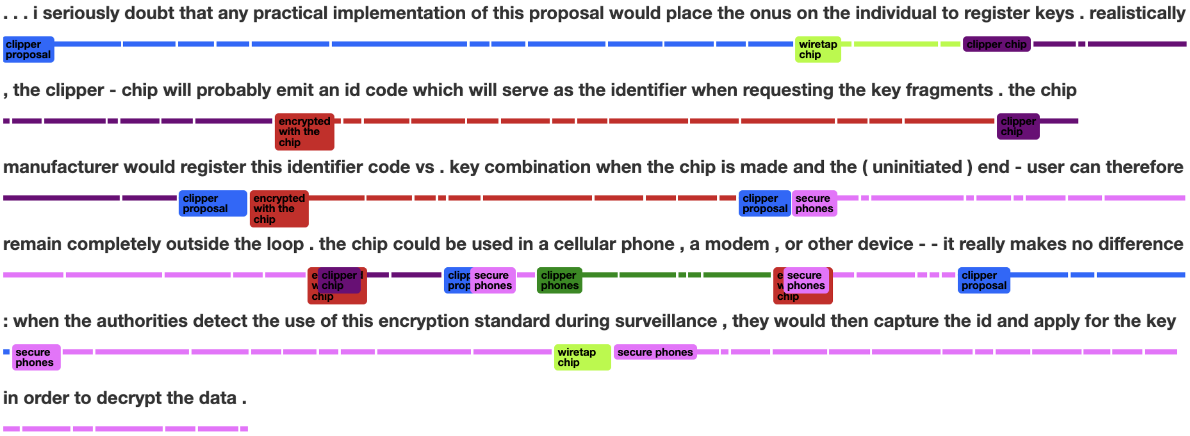
Figure 1: 2D UMAP projected sub-document vectors and top phrase topic labels discovered by C-Top2Vec on the *20newsgroups* dataset.

2 Model

2.1 Contextual Token Embeddings

The self-attention mechanism introduced in the Transformer (Vaswani et al., 2017) and later used by BERT (Devlin et al., 2019) led to contextualized token embeddings. Unlike traditional word embeddings which have a single vector representation, contextualized embeddings have different vectors depending on their context. Sentence-BERT (SBERT) (Reimers and Gurevych, 2019) uses the BERT architecture to train embedding models that ensure semantically similar documents are close in the embedding space. The sentence embeddings are created by pooling the contextualized token embeddings of the document.

The first step in our algorithm is to create contextualized token embeddings for each document. This is done by embedding the documents with an SBERT model and taking the contextual token embeddings before the pooling layer. Any embedding model can be used in this step as long as it has



. . . i seriously doubt that any practical implementation of this proposal would place the onus on the individual to register keys . realistically , the clipper - chip will probably emit an id code which will serve as the identifier when requesting the key fragments . the chip manufacturer would register this identifier code vs . key combination when the chip is made and the (uninitiated) end - user can therefore remain completely outside the loop . the chip could be used in a cellular phone , a modem , or other device - - it really makes no difference : when the authorities detect the use of this encryption standard during surveillance , they would then capture the id and apply for the key in order to decrypt the data .

Figure 2: Example of labelled document topic spans.

Topic Labels	BERTScore$_R$
C-Top2Vec	
turkish genocide, victims of the turkish, genocide of the muslims	0.564
sports car, sports cars, mustang gt, toyota celica, sport coupe, honda accord	0.515
christian doctrine, bible contradictions, biblical interpretation, biblical contradictions	0.412
introduction to atheism, religious sects, atheist position, religious persecution	0.383
Top2Vec	
graphics, graphical, freeware, tiff, bitmap, software, toolkits, fortran, adobe, pixmap	0.500
xmu, xterm, libxmu, xdm, xfree, openwindows, gui, xm, interface, xloadimage	0.448
propaganda, discussions, discussion, threatened, debate, rushdie, argument, arguments	0.359
prices, price, deals, sale, purchasing, sells, interested, offers, selling, buyer	0.351
CTM	
key, secure, keys, security, chip, encryption, government, use, algorithm, secret	0.480
dos, ftp, graphics, code, software, pub, available, files, unix, pc	0.457
hr, suggested, un, remain, frequently, abuse, covered, bj, structure, capable	0.316
hr, suggested, dod, un, remain, changes, consistent, connected, capable, ordered	0.315

Table 1: Top 2 and bottom 2 topics based on topic BERTScore$_R$ on the *20newsgroups* dataset.

contextual tokens, it uses average pooling, and it was trained for semantic similarity.

2.2 Multi-vector Document Representation

Most neural models represent documents as single vectors. However, single vector document representations cannot capture all contextual information, especially in long documents or ones with diverse topics (Luan et al., 2021), (Zhang et al., 2022a).

The second step of our algorithm is to create multiple vectors for each document in order to capture topical information from each part of the document. This is done by using a sliding fixed-sized window with mean pooling over the contextualized token embeddings of each document. This operation aggregates information from multiple contextualized token embeddings and creates a single vector for each position. The pooled vectors from each position represent the topical information from that segment of the document. The size of the window determines the granularity of the representation of the document topic. We use a window size of 50 and stride of 40. See table 7 for window size comparison.

2.3 Topic Vectors

The main premise of the algorithm is that the embedding space represents semantic similarity and density in that space represents a common underlying topic (Angelov, 2020). Using our multi-vector representation of documents we find dense areas of those sub-document vectors. The dense areas are representative of an underlying topic that is common to them. The centroid of each dense area, being the central point, is the most representative of the common topic of the sub-document vectors in that region. Thus, to create topic vectors, we find the centroids of each dense area.

Finding dense areas of vectors that have high dimensions is a challenge due to the curse of dimensionality, which makes finding density computationally expensive and less effective. We use Uniform Manifold Approximation and Projection for Dimension Reduction (UMAP) (McInnes et al., 2018) to find a low-dimensional representation of the vectors. We use UMAP since it preserves the local and global structure of the vectors, allowing us to find dense areas in their low-dimensional representation.

We use Hierarchical Density-Based Clustering (HDBSCAN) (McInnes et al., 2017) on the low-dimensional vectors to find dense areas. We use HDBSCAN since it can find clusters of varying densities and does not require the number of clusters to be specified. For each dense area identified by HDBSCAN we calculate the centroid of the vectors in their original embedding dimension. The resulting centroids become the topic vectors.

The number of initial topics is determined by the number of dense areas found by HDBSCAN. Hierarchical topic reduction can be used to reduce the number of topics to any desired number using the same method used by Top2Vec. Hierarchical topic reduction is performed with the below algorithm until the selected number of topics is reached:

1. For each sub-document vector d_i in the set of sub-document vectors $D = \{d_1, d_2, ..., d_n\}$:
    - Assign d_i to the nearest topic vector in the set of topic vectors $T = \{t_1, t_2, ..., t_m\}$.
    - The size of each topic vector in T is determined by the number of sub-document vectors assigned to it.
2. Find the topic vector t_{min} in T with the smallest size.
3. Find the nearest topic vector $t_{nearest}$ to t_{min}, based on cosine similarity.
4. Merge t_{min} and $t_{nearest}$ by taking their size-weighted mean.

2.4 Topic Assignment and Segmentation

We assign topics to document segments at the token level. The token level topic assignment is done using the contextualized token embeddings of each document and the topic vectors as follows:

1. For each document contextualized token vector c_i in the set $C = \{c_1, c_2, ..., c_n\}$:
    - Using a centered window of size 3, average token vector c_i with adjacent tokens
    - Calculate cosine similarity of pooled c_i with each topic vector t.
    - Assign c_i to the topic vector t with the highest similarity.
2. For each document d:
    - Count the occurrences of each topic based on tokens' assigned topics.
    - Create a topic distribution for document d by dividing the count of each topic by the total number of tokens in d.
3. For each topic t in a document d:
    - Compute the mean cosine similarity between each contextualized token vector c_i assigned to t. This average represents the relevance of topic t in document d.

2.5 Topic Labelling

To create labels for each topic we use phrases extracted from the entire set of documents. We use pointwise mutual information to find n-gram collocations (Bouma, 2009) (Rehurek and Sojka, 2011) that are present across the documents.

The main assumption is that a topic vector is the most central point of a topic area and therefore near phrases are most representative of the topic. To create labels for each topic we select the most similar phrases based on cosine similarity to the topic vector.

Using the phrases we create topic labels for each topic with the below algorithm:

1. Embed all phrases to create a set of phrase vectors $P = \{p_1, p_2, ..., p_n\}$.
2. For each topic vector t_i in the set of topic vectors $T = \{t_1, t_2, ..., t_m\}$:
    - Find the N nearest phrase vectors in P to the topic vector t_i based on cosine similarity.
3. Assign the N nearest phrases found for each topic vector t_i as the labels for that topic.

3 Experiments

3.1 Datasets

We evaluated the models in two datasets: *20Newsgroups* (Pedregosa et al., 2011) and *Yahoo! Answers* (Zhang et al., 2015a), selected for their large number of labelled topics. Both datasets contain posts on various topics with labels, 20 topics for *20newsgroups* and 10 topics for *Yahoo! Answers*. Due to the size of *Yahoo! Answers* we take a stratified sample of 50,000 documents, and we use all 18,846 from *20Newsgroups*. We filter the datasets by removing any empty documents and documents shorter than 35 characters. For additional information on datasets see Appendix B.

3.2 Metrics

Normalized Pointwise Mutual Information C_{NPMI} is used to evaluate topic coherence by measuring co-occurrence of the top topic words within documents from the actual corpus. It is a widely used metric for topic coherence evaluation that has been shown to correlate with human judgements (Röder et al., 2015).

Topic Coherence C_V also looks at the top topic words of a topic but it calculates NPMI of words using a sliding window rather than at the document level. (Röder et al., 2015).

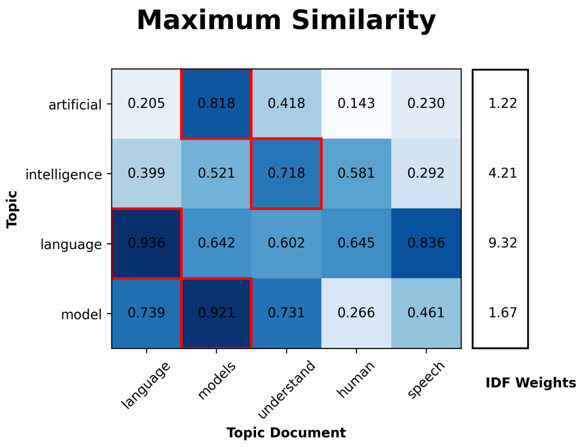
Word Embedding Coherence C_{WE} We use pre-trained `word2vec` (Mikolov et al., 2013) embeddings trained on a one-hundred-billion word corpus. Using the embeddings we find the average pairwise cosine similarity of top topic words as proposed by (Ding et al., 2018) and used by (Bianchi et al., 2021a). This is intended to evaluate the coherence of the words relative to an external corpus.

SBERT Word Embedding Coherence C_{SBERT} We also propose a modification the the word embedding coherence by using an SBERT (Reimers and Gurevych, 2019) model to embed words instead of `word2vec`. This leverages the advancements in neural architectures and also allows for embedding phrases rather than just words. We use this aproach to also calculate the average pairwise cosine similarity of top topic words and phrases.

Adjusted Rand Index ARI proposed by (Hubert and Arabie, 1985) measures the similarity between two clusterings by comparing the cluster labels only. We use it to evaluate how well topic models assign documents to topics compared to the ground-truth labels from the reference dataset. The reference clusters are the true topic labels of each document from the labelled dataset which are compared against the clusters formed by the topic model topic assignment of each document.

Adjusted Mutual Information AMI proposed by (Vinh et al., 2009) measures the similarity of two clusterings by comparing cluster labels only. It can be more effective when dealing with unbalanced cluster sizes and small clusters as compared to ARI. We use it in the exact same way as ARI to evaluate document topic assignments with the true labels as a reference.

BERTScore Most topic model evaluations focus on topic coherence and do not directly evaluate how representative topics are of documents of that topic. A topic could be coherent but the wrong documents could be assigned to that topic or it may not represent topical content accurately. In order to address this gap we propose using BERTScore (Zhang et al., 2020) to evaluate both topic coherence and how representative the topics are of their underlying documents.



$$Score = \frac{(0.818 \times 1.22) + (0.718 \times 4.21) + \ldots}{1.22 + 4.210 + 9.32 + 1.67}$$

Figure 3: BERTScore maximum similarity

BERTScore was initially proposed to evaluate text generation and has been shown to correlate well with human judgements (Zhang et al., 2020).

It has also been used to evaluate abstract summaries of topics represented as phrases or multiple words (Mrini et al., 2021). Given a reference and a candidate sequence, BERTscore uses contextualized token embeddings of each to measure the similarity between candidate and reference tokens.

BERTScore computes a recall, BERTScore$_R$, precision, BERTScore$_P$, and F1 score, BERTScore$_{F1}$. We use the recall as a measure of topic coherence as it evaluates how well each of the topic words is represented in the documents. We use precision to measure how representative the topic is of the document as it evaluates how well the contents of the documents are captured by the topic words.

To evaluate topics we create contextualized embeddings for the top N topic words of each topic. We also produce contextualized token embeddings for documents of each topic. For each topic, we calculate the cosine similarity between its contextualized token embeddings and the embeddings of the tokens from each topic document.

BERTScore uses maximum similarity as shown in Figure 3, so for each topic word we take the most similar token from the topic document to calculate BERTScore. The greedy matching allows BERTScore to be used even though topic descriptions have fewer tokens than documents and allows it to handle documents of varying lengths. To compute BERTScore, a weighted average of the max scores is taken using the Inverse Document Frequency (IDF), which puts less weight on stopwords and more weight on informative words.

BERTScore$_R$ is computed by taking the maximum of cosine similarities for each of the topic's contextualized tokens. This approach ensures that for each token in the topic, its best representation in the document is recognized. BERTScore$_P$ is computed by taking the maximum of cosine similarities for each of the document's contextualized tokens. This ensures that each token in the document is matched with its closest counterpart in the topic, reflecting how accurately the document tokens are represented by the topic tokens. We average these scores over all topics and their documents to evaluate a model.

3.3 Evaluated Models

We evaluate LDA, ETM, CTM, `Top2Vec`, BERTopic and our proposed model. To level the playing field between the neural methods we use the same SBERT model, `all-mpnet-base-v2`, which is based on MPNet (Song et al., 2020) for all of them except ETM which uses `word2vec`. We use OCTIS (Terragni et al., 2021), a framework for topic model evaluation, to evaluate models with C$_{\text{NPMI}}$, C$_{\text{V}}$, and C$_{\text{WE}}$. For the experimental setup, see Appendix C.

3.4 Results

We train LDA, ETM, CTM, `Top2Vec`, BERTopic and `C-Top2Vec` on the *20newsgroups* dataset for 20, 60, 100 and 140 topics and on the *Yahoo! Answers* dataset for 10, 60, 100 and 140 topics. We average results over the multiple runs for each dataset.

Topic Coherence Evaluation To evaluate topic coherence we use the top 10 words of each topic for LDA, ETM, CTM, `Top2Vec`, and BERTopic. For our approach, we used the top 10 phrases of each topic. To evaluate the phrases with C$_{\text{NPMI}}$, C$_{\text{V}}$, and C$_{\text{WE}}$ we split up the ordered phrases into words and take the top 10, as these methods require single-word descriptions. For C$_{\text{SBERT}}$ we use the entire phrases for our approach.

Table 2 shows that `C-Top2Vec` outperforms all other models on C$_{\text{NPMI}}$ and C$_{\text{SBERT}}$ while remaining competitive in the other metrics, demonstrating that `C-Top2Vec` produces more coherent topics. `Top2Vec` has better performance than our approach on C$_{\text{WE}}$ results, this is likely because we split up the phrases into single words and take only the top 10 words. This reduces the coherence as can be demonstrated by the C$_{\text{SBERT}}$ score which allows for measuring the coherence using the entire phrases.

Model	**C$_{\text{NPMI}}$**	**C$_{\text{V}}$**	**C$_{\text{WE}}$**	**C$_{\text{SBERT}}$**
	20newsgroups			
LDA	-0.068	0.460	0.026	0.244
ETM	-0.004	0.508	0.042	0.244
CTM	0.032	0.615	0.046	0.269
BERTopic	0.027	0.540	0.043	0.260
Top2Vec	-0.070	0.561	**0.126**	0.480
C-Top2Vec	**0.138**	**0.684**	0.116	**0.514**
	YahooAnswers			
LDA	-0.086	0.374	0.023	0.235
ETM	-0.010	0.426	0.022	0.249
CTM	0.050	**0.601**	0.038	0.294
BERTopic	0.028	0.486	0.052	0.290
Top2Vec	-0.068	0.494	**0.106**	0.473
C-Top2Vec	**0.094**	0.532	0.090	**0.506**

Table 2: Topic coherence scores.

Topic Assignment Evaluation To evaluate how well the models group documents into topics we compare their topic assignment with the original topic labels of the datasets. For LDA, ETM, and CTM we use the document topic distribution and assign each document to the topic with the highest probability. `Top2Vec` and BERTopic only assign documents to a single topic so we just use those assignments. For our approach, we combine the document topic probability and the document topic relevance by taking their element-wise multiplication and then assigning each document to the topic with the highest resulting score.

The results in Table 3 show the AMI and ARI results averaged over 5 runs comparing the true topic labels to assigned topic labels for each dataset. Our results are tied with `Top2Vec` with a very slight edge, over all other models.

Model	**AMI**	**ARI**
	20newsgroups	
LDA	0.261±0.018	0.096±0.024
ETM	0.447±0.019	0.266±0.016
CTM	0.210±0.012	0.102±0.008
BERTopic	0.420±0.010	0.138±0.013
Top2vec	0.574±0.008	0.437±0.011
C-Top2Vec	**0.582±0.008**	**0.442±0.008**
	YahooAnswers	
LDA	0.147±0.017	0.083±0.019
ETM	0.247±0.002	0.195±0.002
CTM	0.153±0.005	0.101±0.005
BERTopic	0.217±0.015	0.054±0.006
Top2Vec	0.371±0.018	0.314±0.017
C-Top2Vec	**0.376±0.020**	**0.318±0.025**

Table 3: Topic assignment evaluation.

Topic BERTScore Evaluation To evaluate the model's topic coherence and how well the topics represent their underlying documents we use the top 10 topic words for LDA, ETM, CTM, BERTopic and `Top2Vec` as the references. For `C-Top2Vec` we use the top 10 phrases as the references. For the candidates, we use the top 50 documents of each topic. For LDA, ETM, CTM we select the top 50 candidates by taking the 50 documents with the highest probability of each topic. For BERTopic and `Top2Vec` we select the 50 documents with the highest similarity score to each topic.

In order to select the top 50 candidates for our model, we find the top 50 most relevant documents to each topic by using the element-wise multiplication of the document topic probability and the document topic relevance. Further, to evaluate our topic segmentation, we use only the segments from the document that are assigned to the topic rather than using the entire document.

To compute the contextualized token embeddings of topics and documents, as suggested by the official BERTScore Github[1], we use the `microsoft/deberta-xlarge-mnli` model (He et al., 2021) which has the best correlation with human judgments.

To compute BERTScore recall, BERTScore$_R$, we create contextualized token embeddings for each topics top 10 words and its top 50 documents. We then compute pairwise cosine similarity between the embeddings of the topic's top 10 words and the embeddings of the topic documents. We calculate BERTScore$_R$ for each topic by taking the maximum of similarity scores for each of the topic's contextualized tokens and doing a weighted sum using the IDF values of each word as shown Figure 3. We then average all the scores from each topic to have a single BERTScore$_R$.

The results in Table 4 show that our approach outperforms all other models. The BERTScore$_R$ demonstrates that our topics are significantly more coherent in relation to their underlying documents. The BERTScore$_P$ shows that the documents are well represented by the topics.

Model	**BERTScore$_R$**	**BERTScore$_P$**	**BERTScore$_{F1}$**
	20newsgroups		
LDA	0.384	0.316	0.344
ETM	0.378	0.250	0.300
CTM	0.398	0.294	0.336
BERTopic	0.356	0.290	0.318
Top2Vec	0.410	0.309	0.351
C-Top2Vec	**0.456**	**0.374**	**0.409**
	YahooAnswers		
LDA	0.365	0.347	0.352
ETM	0.392	0.265	0.315
CTM	0.414	0.322	0.359
BERTopic	0.351	0.308	0.326
Top2Vec	0.387	0.344	0.362
C-Top2Vec	**0.439**	**0.399**	**0.416**

Table 4: Topic BERTScore evaluation.

[1] https://github.com/Tiiiger/bert_score

Qualitative Analysis We show our sub-document vectors in a 2D UMAP plot along with the topic label assigned by `C-Top2Vec` for 20 topics on the *20newsgroups* dataset in Figure 1. In Figure 2, we show the topic spans assigned by `C-Top2Vec` visualized with `displacy` (Honnibal and Montani, 2017). In Table 1, we show the top 2 and the bottom 2 topics based on their $\text{BERTScore}_\text{R}$ from the top performing models.

Execution Time Our model uses contextual token embeddings instead of a single vector per document however the embedding generation time remains unchanged. Table 5 shows a comparison of the running times for all the evaluated models on 20 topics for the *20newsgroups* dataset.

Model	Runtime
LDA	1 min
ETM	16 min
CTM	5 min
BERTopic	1 min
Top2Vec	2 min
C-Top2Vec	4 min

Table 5: Model run time comparison.

Topic Span Effect Analysis We assess our topic span segmentation with BERTScore using three variations: the complete document, solely the topic segments, and the non-topic segments. The same datasets and configuration as in the BERTScore Evaluation section are used. The results shown in Table 6, demonstrate that `C-Top2Vec` correctly selects the topic segments as the topic span $\text{BERTScore}_\text{R}$ remains high when non-topic spans are removed and is much lower when non-topic spans are used.

Configuration	BERTScore_R
20newsgroups	
Whole Document	0.457
Topic Spans	0.456
Non-topic Spans	0.365
YahooAnswers	
Whole Document	0.440
Topic Spans	0.439
Non-topic Spans	0.313

Table 6: Topic span evaluation.

Window Size Effect Analysis We evaluate the impact of window size on our multi-vector document representation by calculating the Silhouette Score (Rousseeuw, 1987) for UMAP-reduced vectors, using the dataset's topic labels as a reference. Table 7 shows that when using all tokens, equivalent to a single document vector, or very small window sizes, the Silhouette scores are low, indicating poor cluster separation. In contrast, a window size of around 50 yields the highest Silhouette score and thus provides the best clustering. This demonstrates that the multi-vector document representation is better for generating topic vectors than the single-vector representation.

Window size	Silhoutte Score
20newsgroups	
15	0.140
25	0.182
50	**0.194**
100	0.192
All tokens	0.133
YahooAnswers	
15	0.144
25	0.152
50	**0.162**
100	0.127
All tokens	0.114

Table 7: Effect of the window size.

4 Conclusion

We propose a novel topic modeling approach that leverages contextual token embeddings to create multi-vector document representations that capture topical information from each segment of a document. Our method finds topic vectors and labels each topic using coherent and easily interpretable phrases. `C-Top2Vec` supports hierarchical topic reduction, enabling it to handle topics at various levels of granularity. Our model segments documents into topic spans, enabling detailed and granular analysis of topics. Our model produces a document topic distribution and a document topic relevance score which allows for ranking of document segments according to their relevance to a specific topic. We propose the use of BERTScore for evaluating both topic coherence and how informative topics are of their underlying documents overcoming previous gaps in evaluation. Our findings, based on a comprehensive set of topic modeling evaluation metrics, demonstrate that our model outperforms current state-of-the-art models.

Limitations

Using pre-trained embedding models in neural topic modeling offers significant benefit by bringing external knowledge; however, these models may not fully grasp the nuances of a specific corpus. Additionally, the models can introduce external biases as they are pre-trained on diverse datasets, potentially affecting the accuracy of the topic models.

Topic model evaluation measures have limitations as they cannot fully capture the nuances of natural language. The effectiveness of a model is highly dependant on the specific use case and dataset therefore a custom approach should always be used for model selection and evaluation.

Ethical Statement

In order to ensure the reproducibility of our research and in the spirit of fostering openness and transparency, we will make the source code associated with this paper available on GitHub.

In our research, we used datasets commonly used in evaluating topic models and do not use or infer any sensitive information.

Using pre-trained embedding models does inherently introduce bias into topic modeling, thus it is very important to understand what datasets the embedding models were trained on.

In general, the risk of possible abuse of `C-Top2Vec` is low.

References

Aly Abdelrazek, Yomna Eid, Eman Gawish, Walaa Medhat, and Ahmed Hassan. 2023. Topic modeling algorithms and applications: A survey. *Information Systems*, 112:102131.

Tuncer AKBAY. 2022. Modeling education studies indexed in web of science using natural language processing. *Instructional Technology and Lifelong Learning*, 3(2):129–143.

Dimo Angelov. 2020. Top2vec: Distributed representations of topics. *arXiv preprint arXiv:2008.09470*.

Shraey Bhatia, Jey Han Lau, and Timothy Baldwin. 2017. An automatic approach for document-level topic model evaluation. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 206–215, Vancouver, Canada. Association for Computational Linguistics.

Federico Bianchi, Silvia Terragni, and Dirk Hovy. 2021a. Pre-training is a hot topic: Contextualized document embeddings improve topic coherence. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 759–766, Online. Association for Computational Linguistics.

Federico Bianchi, Silvia Terragni, Dirk Hovy, Debora Nozza, and Elisabetta Fersini. 2021b. 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.

Federico Bianchi, Silvia Terragni, Dirk Hovy, Debora Nozza, and Elisabetta Fersini. 2021c. 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.

Gerlof Bouma. 2009. Normalized (pointwise) mutual information in collocation extraction. *Proceedings of the Biennial GSCL Conference 2009*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Adji B Dieng, Francisco JR Ruiz, and David M Blei. 2020. Topic modeling in embedding spaces. *Transactions of the Association for Computational Linguistics*, 8:439–453.

Ran Ding, Ramesh Nallapati, and Bing Xiang. 2018. Coherence-aware neural topic modeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 830–836, Brussels, Belgium. Association for Computational Linguistics.

Thanh-Nam Doan and Tuan-Anh Hoang. 2021. Benchmarking neural topic models: An empirical study. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4363–4368.

Caitlin Doogan and Wray Buntine. 2021. Topic model or topic twaddle? re-evaluating semantic interpretability measures. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3824–3848, Online. Association for Computational Linguistics.

Roman Egger and Joanne Yu. 2022. A topic modeling comparison between lda, nmf, top2vec, and bertopic to demystify twitter posts. *Frontiers in sociology*, 7:886498.

Derek Greene and Pádraig Cunningham. 2006. Practical solutions to the problem of diagonal dominance in kernel document clustering. In *Proceedings of the 23rd international conference on Machine learning*, pages 377–384.

Maarten Grootendorst. 2022. Bertopic: Neural topic modeling with a class-based tf-idf procedure. *arXiv preprint arXiv:2203.05794*.

Philip J. Hayes and Steven P. Weinstein. 1990. CONSTRUE/TIS: a system for content-based indexing of a database of news stories. In *Second Annual Conference on Innovative Applications of Artificial Intelligence*.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. {DEBERTA}: {DECODING}-{enhanced} {bert} {with} {disentangled} {attention}. In *International Conference on Learning Representations*.

Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear.

Alexander Hoyle, Pranav Goel, Andrew Hian-Cheong, Denis Peskov, Jordan Boyd-Graber, and Philip Resnik. 2021. Is automated topic model evaluation broken? the incoherence of coherence. *Advances in neural information processing systems*, 34:2018–2033.

Lawrence Hubert and Phipps Arabie. 1985. Comparing partitions. *Journal of classification*, 2:193–218.

Hamed Jelodar, Yongli Wang, Chi Yuan, Xia Feng, Xiahui Jiang, Yanchao Li, and Liang Zhao. 2019. Latent dirichlet allocation (lda) and topic modeling: models, applications, a survey. *Multimedia Tools and Applications*, 78:15169–15211.

Bradley Karas, Sue Qu, Yanji Xu, and Qian Zhu. 2022. Experiments with lda and top2vec for embedded topic discovery on social media data—a case study of cystic fibrosis. *Frontiers in Artificial Intelligence*, 5:948313.

Yi Luan, Jacob Eisenstein, Kristina Toutanova, and Michael Collins. 2021. Sparse, dense, and attentional representations for text retrieval. *Transactions of the Association for Computational Linguistics*, 9:329–345.

Leland McInnes, John Healy, and Steve Astels. 2017. hdbscan: Hierarchical density based clustering. *Journal of Open Source Software*, 2(11):205.

Leland McInnes, John Healy, and James Melville. 2018. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc.

Khalil Mrini, Can Liu, and Markus Dreyer. 2021. Rewards with negative examples for reinforced topic-focused abstractive summarization. In *Proceedings of the Third Workshop on New Frontiers in Summarization*, pages 33–38, Online and in Dominican Republic. Association for Computational Linguistics.

Shirui Pan, Jia Wu, Xingquan Zhu, Chengqi Zhang, and Yang Wang. 2016. Tri-party deep network representation. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*, pages 1895–1901. IJCAI/AAAI Press.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830.

Radim Rehurek and Petr Sojka. 2011. Gensim–python framework for vector space modelling. *NLP Centre, Faculty of Informatics, Masaryk University, Brno, Czech Republic*, 3(2).

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Conference on Empirical Methods in Natural Language Processing*.

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*, pages 399–408.

Peter J Rousseeuw. 1987. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. Mpnet: masked and permuted pre-training for language understanding. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA. Curran Associates Inc.

Akash Srivastava and Charles Sutton. 2017. Autoencoding variational inference for topic models. In *International Conference on Learning Representations*.

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. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Nguyen Xuan Vinh, Julien Epps, and James Bailey. 2009. Information theoretic measures for clusterings comparison: is a correction for chance necessary? In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, page 1073–1080, New York, NY, USA. Association for Computing Machinery.

Xiaobao Wu, Thong Nguyen, and Anh Tuan Luu. 2024. A survey on neural topic models: Methods, applications, and challenges. *Artificial Intelligence Review*, 57(2):1–30.

Shunyu Zhang, Yaobo Liang, Ming Gong, Daxin Jiang, and Nan Duan. 2022a. Multi-view document representation learning for open-domain dense retrieval. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5990–6000, Dublin, Ireland. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015a. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28.

Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. 2015b. Character-level convolutional networks for text classification. In *NIPS*.

Zihan Zhang, Meng Fang, Ling Chen, and Mohammad Reza Namazi Rad. 2022b. Is neural topic modelling better than clustering? an empirical study on clustering with contextual embeddings for topics. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3886–3893, Seattle, United States. Association for Computational Linguistics.

A Additional Related Work

Recent interest in neural topic modeling methods has created a rapidly evolving landscape that is difficult to cover exhaustively. This appendix attempts to highlight additional relevant neural topic modeling approaches that are not detailed in the main text. The purpose is to acknowledge the depth of ongoing research and to guide the reader to other relevant methods.

In their work, (Zhang et al., 2022b) explore clustering methods for topic modeling and follow a similar approach to BERTopic (Grootendorst, 2022) for topic labelling. They demonstrate competitive topic coherence compared to neural topic modeling approaches. The limitation of their approach is the BOW representation of clusters for topic labeling.

Another notable contribution is from (Bianchi et al., 2021c) introduce cross-lingual neural topic models that can be trained on one language and applied to another. This paper uses contextual document representations instead of BOW representations of documents which allows for zero-shot cross-lingual topic modeling.

In their study, (Ding et al., 2018) propose adding a training objective to maximize topic coherence. They show that their approach increases topic coherence while maintaining a similar level of perplexity as baseline models.

The authors of (Doan and Hoang, 2021) evaluate several neural topic models and compare them to traditional probabilistic models. Their results show that neural models are better at finding coherent topics and creating representations useful for downstream tasks; however, they also conclude that traditional models are strong baselines and are sometimes better at modeling the documents.

For further reading, we suggest "A survey on neural topic models: Methods, applications, and challenges" (Wu et al., 2024) and "Topic modeling algorithms and applications: A survey" (Abdelrazek et al., 2023).

B Additional Datasets

The most commonly used dataset for topic model evaluation is the *20newsgroups* dataset (Pedregosa et al., 2011). It contains around 18,000 posts from 20 newsgroups that are each on a different topic. It is popular due to the diverse range of topics and associated labels.

Another very popular dataset is *Reuters-21578* (Hayes and Weinstein, 1990) which contains around 20,000 news articles from Reuters newswire and it contains multiple labels for each document.

The *AG news* topic classification dataset (Zhang et al., 2015b) contains roughly 130,000 news articles with 4 different topic labels.

The *M10* dataset (Pan et al., 2016) contains roughly 10,000 documents that are scientific publications with 10 distinct topic labels.

The *BBC News* (Greene and Cunningham, 2006) dataset contains roughly 2,000 articles with 5 distinct topic labels.

The *Yahoo! Answers* topic classification dataset (Zhang et al., 2015a) contains roughly 1,400,000 questions and their answers. The original dataset can be acquired from the Yahoo! Webscope program.

Our goal was to evaluate not just topic coherence, but also how representative topics are of their underlying documents. An essential criterion of our dataset selection process was to have ground truth on topic labels so that we can perform cluster evaluation. There are many other available datasets but based on our review of the literature and datasets *20newsgroups* and *Yahoo! Answers* best met our criteria. We chose the two datasets due to their size and number of topic labels. All other available datasets were either too small, contained too few labels, or did not have a single label per document.

C Experimental Setup

Model Training We trained LDA and ETM using the OCTIS framework (Terragni et al., 2021) for standardization and reproducibility. We trained CTM, BERTopic and Top2Vec using thier respective Github implementations[2][3][4].

Due to the computational cost and time required for training multiple models across a range of topics and evaluating with BERTScore and other evaluation metrics, we opted to train each model once for each configuration.

Model Parameters To train LDA and ETM we used OCTIS (Terragni et al., 2021) and their suggested model parameters. For CTM we use the suggested parameters from the paper (Bianchi et al., 2021a) and their Github[2]. For BERTopic (Grootendorst, 2022), due to the little information on parameters in the paper we use the default values from their Github[3]. For Top2Vec (Angelov, 2020) we use the suggested parameters from the paper and their Github[4]. All relevant evaluated model parameters are shown in Table 8.

Model Parameters

Parameter	Value
LDA	
decay	0.5
gamma_threshold	0.001
iterations	50
ETM	
num_epochs	100
t_hidden_size	800
t_hidden_size	800
rho_size	300
embedding_size	300
activation	relu
dropout	0.5
lr	0.005
optimizer	adam
batch_size	128
wdecay	1
CTM	
embedding_model	`all-mpnet-base-v2`
hidden_sizes	(100, 100)
dropout	0.2
learn_priors	True
batch_size	64
lr	2×10^{-3}
momentum	0.99
solver	adam
num_epochs	100
BERTopic	
embedding_model	`all-mpnet-base-v2`
top_n_words	10
umap_n_neighbours	15
umap_n_components	5
hdbscan_min_cluster_size	10
Top2Vec	
embedding_model	`all-mpnet-base-v2`
umap_n_neighbours	15
umap_n_components	5
hdbscan_min_cluster_size	15
min_count	50
C-Top2Vec	
embedding_model	`all-mpnet-base-v2`
umap_n_neighbours	50
umap_n_components	5
hdbscan_min_cluster_size	15
min_count	50
window_size	50
stride	40
smoothing_window_size	3

Table 8: Model parameters for all the evaluated models.

[2] https://github.com/MilaNLProc/contextualized-topic-models

[3] https://github.com/MaartenGr/BERTopic

[4] https://github.com/ddangelov/Top2Vec