MATHÉMATIQUES
VISION
APPRENTISSAGE

INTERNSHIP REPORT
Master MVA
2021/2022

Dynamic Topic Modeling

Research intern :
David SOTO

Under the supervision of :
Alain Rakotomamonjy, Criteo
Alberto Lumbreras, Criteo
Michael Arbel, ENS Paris-Saclay

école
normale
supérieure
paris—saclay

CRITEO

Table of content

Abstract

Topic modeling analyzes documents to learn meaningful patterns of words. For documents collected in sequence, dynamic topic models capture how these patterns vary over time. In this research work, we develop a dynamic topic model that leverages contextual embeddings to capture the evolution of topics. The proposed model combines a semantic space of embedded documents, a semantic space of word vectors and a distance metric to uncover topics from a stream of text data and analyze the evolution of topics over time in an efficient way. The semantic space of embedded documents allows a thorough and meaningful detection of topics. The semantic space of embedded words and the Chamfer distance enables the model to capture topic trajectories. On three different corpora - a collection of Trump's tweets, United Nations debates and the 20 NewsGroups dataset - we found that our model outperforms BERTopic, a state-of-the-art dynamic topic model, on the detection of ground-truth topics over time.

1 Introduction

With the exponential growth of the data collection, most of them will never be directly seen by a human. As these data are likely to contain valuable information for many applications, it is crucial to develop tools capable of handling the retrieve of information from data collection. Among all data collected and processed, textual data is a type of data that contains a lot of information that could be useful to many, starting with companies. However, analysing text data is a real challenge because most of the time text data are unstructured and given in huge amounts, due to the growth of data collection. Therefore, it is patent that data analysis is a task that cannot be supervised by humans, let alone be done by humans. For example, data analysis of social media posts, emails, chats, news articles, and more, is not an easy task, and less so when delegated to humans alone. This kind of tasks needs to be automated and be done by machines, but these tasks need to be performed efficiently to extract the most relevant and valuable information contained in data. For instance, for a company that processes hundreds of thousands of customer interactions every day, it is of paramount importance to analyze the textual content in an efficient way to understand the need or behavior of customers.

It is for these reasons that many are eager about the implications artificial intelligence could have on their day-to-day tasks, as well as on businesses as a whole. AI-powered text analysis uses a wide variety of methods or algorithms to process language naturally, one of which is topic modeling – used to automatically detect topics and themes from a text. By using machine learning techniques, businesses are now able to analyze enormous amount of text data in a short period of time which helps these companies to be more productive.

1.1 Topic modeling

Topic modeling is an unsupervised machine learning technique that identifies hidden relationships in text data. It is capable of scanning sets of documents, detect word and phrase patterns within them, and automatically clustering word groups and similar expressions that best characterize a set of documents. Topic modeling works in an exploratory manner, looking for the topics that lie within a set of text data, and discover topics using a probabilistic framework to infer the themes within the data based on the words observed in the documents. Being unsupervised, topic modeling doesn't need labeled data and can be applied directly to a set of text documents to extract information. Topic modeling is a versatile way of making sense of an unstructured collection of text documents. It can be used to automate the process of sifting through large volumes of text data and help to organize and understand it. It is for these reasons that companies such as Criteo use topic modeling techniques to analyze text data collection. Topic modeling is used when a large collection of text cannot be reasonably read and sorted through by a person.

Topic models are useful tools for the statistical analysis of document collections (Blei et al. [4] [3]). They have been applied to documents from many fields, including marketing, sociology, political science, and the digital humanities; see Boyd-Graber et al.[6] for a review. One of the most common topic models is latent Dirichlet allocation (LDA) [4], a probabilistic model that represents each topic as a distribution over words and each document as a mixture of the topics. LDA has been extended in different ways, for example to capture correlations among the topics (Lafferty and Blei [17]), to classify documents (Blei and McAuliffe [19]), or to analyze documents in different languages (Mimno et al. [21]).

1.2 Dynamic topic modeling

Dynamic topic modeling refers to the introduction of a temporal dimension into the topic modeling analysis. It consists in modeling the evolution of prevalent themes in literature, online media, and other forms of text data over time. The introduction of a temporal aspect in topic modeling includes the analysis of the evolution of topics over time. The latter is a challenging field in dynamic topic modeling and one of the problems we tackle in our work. The dynamic aspect of topic modeling is a growing area of research and has seen many applications, including semantic time-series analysis, unsupervised document classification, and event detection.

Overall, dynamic topic modeling consists in discovering topics from a stream of text data over time, which includes the analysis of topics' quality as they arrive in flow and the analysis of the detection of new emerging topics, that is topics that appear progressively over time until becoming prominent topics.

1.3 Organization of the report

This report is organized as follows : the first section is a general introduction to dynamic topic modeling, then we explain the chief goal of our research work which is to enhance state-of-the-art dynamic topic models. In particular, we want to enhance the detection of emerging topics. The next section is a small presentation of Criteo and we talk about the utility of our reasearch work for the company (which is to propose a state-of-the-art dynamic topic model that can be used for Criteo's applications). Then, we introduce important notions to understand our work in the background section. Specifically, we introduce some popular techniques in dynamic topic modeling and point out their limitations and explain why we do not use them and why we will try to improve some of them. The next section is dedicated to presenting our dynamic topic model. The sixth section is dedicated to the experiments we conducted. Lastly, we finish with a general conclusion, without forgetting some appendices mainly related to our experiments and bibliography explanations related to our project.

2 Problem statement

Dynamic topic modeling is a growing area of research, and even though there has been some impressive improvement in the last years [28][12][2][8][9], dynamic topic modeling remains a promising area of research. There are many challenges faced in dynamic topic modeling. One main challenge faced in this field is the tracking of topics over time. Indeed, due to the dynamic aspect of topic modeling, we are not just interested in discovering topics but also in analyzing their evolution over time. The topic evolution analysis is at the core of dynamic topic modeling and remains one of the main challenges of this area of natural language processing. A subfield of topic evolution that is of particular interest to us is the discovery of emerging topics as well as the discovery of disappearing topics. In our work, we tackle this challenge by proposing a method for analyzing the evolution of topics that enables to identify new topics and disappearing topics. The dynamic aspect of topic modeling brings another big challenge, which is the variation of the number of topics to be detected. Indeed, the evolution of topics, in particular the appearance of news topics and disappearance of topics brings alterations in the number of topics to be discovered. Dynamic topic models need to capture these changes in the number of topics to be detected in order to be efficient. In our work we will propose a methodology to catch the variations in the number of topics to detect over time.

The first objective of our work is to analyze the state-of-the-art in dynamic topic modeling and, if possible, try to ameliorate what has been done until now. The main goal of our research work is to propose an efficient and practical approach to perform dynamic topic modeling that allows to analyze the evolution of topics over time efficiently. In particular, we aim at proposing an approach that enables to detect emerging topics and vanishing topics with the least delay. This is because the information gained in the detection of new topics and topics disappearing in a collection of documents over time is significant and essential for understanding and analyzing text data.

3 About Criteo

3.1 Presentation of Criteo

Criteo is a global technology company that provides marketers and media owners around the world with reliable and impactful advertising through the Commerce Media Platform, a package of products that enables the world's largest set of business data to achieve better business results. The company helps thousands of brands, publishers and retailers reach and monetize audiences, moreover Criteo is committed to supporting a fair and open Internet that enables discovery, innovation and choice. Technically this means to produce bids and displays for around 15,000 retailers with a typical number of bids around 1 million per seconds which is larger than the number of searches on Google which is estimated at 70,000 per second. Criteo was founded and is headquartered in Paris and has offices around the world where a great team works together to create an open and inclusive environment.

3.2 Presentation of Criteo AI Lab

The Criteo AI Lab merges the Criteo Research and Machine Learning Platform Engineering teams, two major branches of Criteo's RD hub. It is made of 90 permanent members with approximately 30 researchers, other being engineers. Usually, traditional research labs revolve around a single group whose sole mission is to publish studies, or create teams to solve strategic problems. The AI Lab stands out by placing Criteo's expertise at the service of the development of an innovative AI at the cutting edge of technology: both in its integration into production systems and in the deepening of knowledge of Artificial Intelligence techniques. Supported by an investment of 20 million euros over three years, Criteo's AI Lab plays a major role in the company's mission. The research pole of the AI Lab is organized into 4 teams:

- **Exploration, Exploitation and Learning :** this team works mainly on Reinforcement Learning and Bandit problems, but also on the Robustness of Deep Learning models, as well as the Mechanism Design of Markets.

- **Causal Learning :** this team focuses on the study of Causal inference, Counterfactual prediction.

- **Recommendation :** this team works mainly on the development of Recommendation systems.

- **Learning From Text and Structured Data :** the line of research of this team revolves around Deep Learning models as well as Transfer Learning.

3.3 Criteo's interest for our research work

As one can imagine, Criteo has to analyze enormous amounts of text data everyday. Most of these data come from web pages that need to be analyzed in order to understand the content of these web pages and retrieve valuable information. The company exploits these information to understand users' behaviors and preferences in order to perform an efficient ad targeting.

As a consequence, Criteo uses topic modeling and dynamic topic modeling techniques to analyze streams of text data. However, the company wants to enhance the process of text data analysis.

In particular, Criteo is interested in the discovery of emerging topics in flows of text data. This is because being able to detect emerging topics with the least delay would improve the company's performance in ad targeting. Indeed, the detection of emerging topics in a set of web pages visited by users allows to anticipate future popular topics, that is topics that will be of interest to internet users. The detection of emerging topics can ameliorate the company's online ad targeting. The potential enhancement brought by our work in dynamic topic modeling can help Criteo's ad targeting to be more efficient and therefore contribute to increasing the company's profits.

4 Background

In this section we review some important notions we will mention in the report.

4.1 State-of-the-art models

First of all, we introduce two state-of-the-art dynamic topic models that will help us to understand where our approach came from. In particular, we introduce the method that inspired the model we propose in the next section. We start by reviewing LDA and BERTopic; both are non-dynamic topic models. We then review D-LDA, the dynamic extension of LDA, as well as the extension of BERTopic to dynamic topic modeling.

Consider a corpus of D documents, where the vocabulary contains V distinct terms. Let $w_{dn} \in \{1, ..., V\}$ denote the nth word in the dth document.

4.1.1 LDA & Dynamic-LDA

Latent Dirichlet Allocation (LDA) is a probabilistic generative model of documents (Blei et al. [4]). It considers K topics $\beta_{1:K}$, each of which is a distribution over the vocabulary. It further considers a vector of topic proportions θ_d for each document d in the collection; each element θ_{dk} expresses how prevalent the kth topic is in that document. In the generative process of LDA, each word is assigned to topic k with probability θ_{dk}, and the word is then drawn from the distribution β_k. The generative process for each document is as follows:

1. Draw topic proportions $\theta_d \sim \text{Dirichlet}(\alpha_\theta)$.
2. For each word n in the document:
    a. Draw topic assignment $z_{dn} \sim \text{Cat}(\theta_d)$.
    b. Draw word $w_{dn} \sim \text{Cat}(\beta_{z_{dn}})$.

Here, $\text{Cat}(\cdot)$ denotes the categorical distribution. LDA also places a Dirichlet prior on the topics, $\beta_k \sim \text{Dirichlet}(\alpha_\beta)$. The concentration parameters α_β and α_θ of the Dirichlet distributions are model hyperparameters.

Dynamic Latent Dirichlet Allocation (D-LDA) is an extension of LDA to dynamic topic modeling. D-LDA allows topics to vary over time to analyze time-series corpora (Blei and Lafferty [17]). The generative model of D-LDA differs from LDA in that the topics are time-specific, i.e., they are $\beta^{(t)}_{1:K}$, where $t \in \{1, ..., T\}$ indexes time steps. Moreover, the prior over the topic proportions θ_d depends on the time stamp of document d , denoted $t_d \in \{1, ..., T\}$. The generative process for each document is :

1. Draw topic proportions $\theta_d \sim \mathcal{N}(\eta_{t_d}, a^2 I)$.
2. For each word n in the document:
    a. Draw topic assignment $z_{dn} \sim \text{Cat}(\theta_d)$.
    b. Draw word $w_{dn} \sim \text{Cat}(\beta^{(td)}_{z_{dn}})$.

Here, a is a model hyperparameter and η_t is a latent variable that controls the prior mean over the topic proportions at time t. To encourage smoothness over the topics and topic proportions, D-LDA places random walk priors over $\beta_{1:K}^{(t)}$ and η_t,

$$\begin{aligned} \tilde{\beta}_k^{(t)} \mid \tilde{\beta}_k^{(t-1)} &\sim \mathcal{N}(\tilde{\beta}_k^{(t-1)}, \sigma^2 I) \text{ and } \beta_k^{(t)} = \text{softmax}(\tilde{\beta}_k^{(t)}) \\ \eta_t \mid \eta_{t-1} &\sim \mathcal{N}(\eta_{t-1}, \delta^2 I) \end{aligned}$$

The variables $\beta_k^{(t)} \in \mathbb{R}^V$ are the transformed topics; the topics $\beta_k^{(t)}$ are obtained after mapping $\tilde{\beta}_k^{(t)}$ to the simplex. The hyperparameters σ and δ control the smoothness of the Markov chains.

4.1.2 BERTopic

BERTopic is a state-of-the-art topic model with an extension to dynamic topic modeling. Its approach to topic modeling and dynamic topic modeling is different than the LDA approach. BERTopic [12] uses a topic modeling technique that leverages transformers and c-TF-IDF to create dense clusters allowing for easily interpretable topics whilst keeping important words in the topic descriptions.

BERTopic performs topic modeling as follows :

- the model starts by converting documents to numerical representations. Although there are many methods for doing so the default in BERTopic is sentence-transformers [23]. These models are often optimized for semantic similarity which helps tremendously in the clustering task. Moreover, they are great for creating either document or sentence-embeddings.
- After having created numerical representations of the documents, BERTopic reduces the dimensionality of these representations. This is because cluster models typically have difficulty handling high dimensional data due to the curse of dimensionality. There are great approaches that can reduce dimensionality, such as PCA [14], but as a default UMAP [20] is selected in BERTopic.
- Once the embeddings' dimension reduced, BERTopic performs clustering. For that, the model leverages a density-based clustering technique, HDBSCAN. It can find clusters of different shapes and has the nice feature of identifying outliers where possible. As a result, BERTopic do not force documents in a cluster where they might not belong. This improves the resulting topic representation as there is less noise to draw from.
- Afterwards, BERTopic retrieves topic representations of the clusters of documents. To do this, BERTopic uses a cluster-level TF-IDF [12], called class-based TF-IDF. The latter enables BERTopic to retrieve importance scores for each word within a cluster. The more important words are within a cluster, the more it is representative of that topic. In other words, by extracting the most important words per cluster, BERTopic gets the descriptions of topics.

BERTopic contains an extension for dynamic topic modeling. The model allows for dynamic topic modeling by calculating the topic representation at each time stamp without the need to run the entire model several times. To do this, the model first needs to be fitted as if there were no temporal aspect in the data, that is the model performs topic modeling on the entire sets of documents without taking into account the timestamps of the data. Thus, a general topic model is created.

BERTopic uses the topics detected on the global representation of embedded documents as to the main topics that can be found at, most likely, different timestamps. For each topic and timestamp, the model calculates the c-TF-IDF representation. This results in a specific topic representation at each timestamp without the need to create clusters from embeddings as they were already created.

BERTopic has achieve state-of-the-art results in topic modeling and dynamic topic modeling tasks. However, the dynamic extension of BERTopic is based on the assumption that future documents are known in the present. Indeed, when fitting the model as if there were no temporal aspect in the data, BERTopic assumes that the documents coming in the future are already known in the present. In many cases, this hypothesis is not realistic and not adapted to real-life applications. There exists many applications where we do not know the future sets of documents beforehand. Of course, there are cases where BERTopic's approach can work, these are cases where one aims at analysing the dynamic of topics a posteriori, that is where all timestamps have already existed and all documents are known. In Criteo's case, the text data the company deals with everyday cannot be known beforehand. Hence, even though BERTopic's approach for dynamic topic modeling is interesting, the company cannot use BERTopic's extension for dynamic topic modeling as it is not suited for Criteo's use cases.

4.2 Start point of our work

Even though LDA and D-LDA are well designed topic models and dynamic topic models, BERTopic's methodology has some advantages over LDA for our applications of dynamic topic modeling. Firstly, BERTopic leverages contextual embeddings and therefore captures the contextual nature of the text. Moreover, the structure of BERTopic allows for a very flexible algorithm that can easily adapt to new advancements in language models, clustering techniques, dimensionality reduction techniques, etc... This enables the amelioration of the model every time improvements are made. Lastly, BERTopic is fast to run, which is practical for real-life applications. It is for these reasons we decided to extend BERTopic's method to dynamic topic modeling in an efficient and realistic way, adapted to Criteo's applications.

4.3 Normalized Pointwise Mutual Information

In dynamic topic modeling, people use a coherence score to measure how interpretable the topics are to humans. In this case, topics are represented as the top n words with the highest probability of belonging to that particular topic. Topic Coherence (TC) [25] [5] measures how much the top-n words representing each topic are related and therefore the coherence of topics. A well-known metric for topic coherence is Normalized Pointwise Mutual Information (NPMI) [5], which is computed using Normalized Pointwise Mutual Information of each pair of words (w_i, w_j) in the top-n words of each topic. The NPMI score of two words w_i and w_j is given by :

$$NPMI(w_i, w_j) = \frac{log(\frac{P(w_i,w_j)}{P(w_i)P(w_j)})}{-log\left(P(w_i, w_j)\right)}$$

Where $P(w_i, w_j)$ is the probability of words w_i and w_j to occur together. $P(w_i)$ and $P(w_j)$ are the probabilities of words w_i and w_j. Some orientation values of NPMI are as follows : when two words only occur together $NPMI(w_i, w_j) = 1$; when two words occur separately but not together, we define $NPMI(w_i, w_j)$ to be 1, as it approaches this value when $P(w_i, w_j)$ approaches 0 and $P(w_i)$

and $P(w_j)$ are fixed.

The evaluation of the topic coherence of a model is as follows : for each topic detected by the model we compute the NPMI score between each pair of words in the top-n words, and average the results. The TC score of a model is given by the average NPMI score over all topics detected.

5 Proposed model

In this section, we describe our dynamic topic model[1]. Unlike BERTopic [12], our model is based on the assumption that documents coming in the future are unknown in the present. As a consequence, we have to deal with new documents at each timestamp, that is at each timestamp the model deals with documents it has never encountered before. The following subsections describe the steps of the proposed model to perform dynamic topic modeling. Our approach iterates the steps of the model at each timestamp so that we can track topics over time.

5.1 Document embeddings

Given a set of documents, the first step in our approach consists in embedding every document. The embedded documents need to respect certain properties in order to be able to extract topics. The main assumptions in our approach are that documents belonging to the same topic are semantically similar and, on the other hand, documents coming from different topics are semantically dissimilar. For that reason, we embed documents to create representations in vector space that can be compared semantically, i.e. we use an embedding where the distance between document vectors reflects semantic association. Semantically similar documents will be close to each other in the embedding space, whereas dissimilar documents should be placed far from each other. This spatial representation of documents is called a semantic space. We argue that a semantic space with the outlined properties is a continuous representation of topics.

There exists several document embedding model such as Doc2Vec[18], inter alia. In our model, we use Sentence-BERT [23], a python framework for state-of-the-art text embeddings. Sentence-BERT converts sentences and paragraphs to vector representations using pre-trained language models. The framework offers a large collection of pre-trained models tuned for various tasks, among them document embedding. Given the large number of available Sentence-BERT models, it seemed relevant to test the different Sentence-BERT models in our model on a dynamic topic modeling task and compare the results to choose which pre-trained model to incorporate in our model. We tested all pre-trained Sentence-BERT models[2] in our model and performed dynamic topic modeling on 3 different datasets for each Sentence-BERT. At each run, we computed the Topic Diversity (TD) and Topic Coherence (TC) (see section 4.3) , two widely-used metrics for evaluating the quality of the topics detected by dynamic topic models. TC takes values in $[-1, 1]$, where a TC score of 1 indicates a perfect semantic association between words composing a topic, whereas a TC score of -1 indicates poor coherence in the detected topics. TD takes values in $[0, 1]$ and indicates how varied are the topics discovered by the models. The higher the TD score the better. We executed 5 runs for each Sentence-BERT model (incorporated in our model as our document embedding model) and we report the average scores over the 5 runs. This experiment gives us a general overview of the quality of topics detected by our model depending on the document embedding model used. Table 1 displays the results of our test.

[1] Our code is available at https://github.com/sotodavid/Dynamic-Topic-Modeling---MVA or at https://gitlab.crto.in/a.rakotomamonjy/dynamictopicmodelling/-/blob/main/Experiments/DynamicTopicModeling

[2] The Sentence-BERT pre-trained models are available at https://www.sbert.net/docs/pretrained_models.html

Pre-trained Sentence-BERT	Trump dataset		20NewsGroup		United Nations	
	TD	TC	TD	TC	TD	TC
all-mpnet-base-v2	0.830	-0.036	0.756	0.131	0.852	0.171
all-MiniLM-L6-v2	0.826	0.112	0.838	0.161	0.872	0.227
USE	**0.871**	**0.137**	0.854	0.154	**0.891**	**0.242**
multi-qa-mpnet-base-dot-v1	0.793	0.097	0.826	0.152	0.797	0.102
all-distilroberta-v1	0.850	0.105	**0.877**	0.165	0.867	0.156
all-MiniLM-L12-v2	0.843	0.125	0.845	**0.166**	0.861	0.209

Table 1: **Tests of different Sentence-BERT pre-trained models on our model.** The different document embedding models were tested on 3 datasets : the Trump dataset, the 20NewsGroup dataset and the United Nations dataset. We evaluate the models using the Topic Diversity (TD) and Topic Coherence (TC) metrics.

As we can see in Table 1, there is no model that outstands on the three datasets. Depending on the datasets, some Sentence-BERT models obtain better results than the others. Overall, *USE*[3] seem to be a reliable pre-trained document embedding model in the sense that it obtains pretty good results on the three datasets. *USE* returns document embeddings of dimension 768. Hence, our model uses *USE* as its document embedding model.

5.2 Dimension reduction

Once the document embeddings created, the next step consists in reducing the document vectors' dimensions. Among other benefits, low dimensional document embedding allows for an efficient and accurate clustering process. Indeed, it has been proven by Aggarwal et al.[1] and by Beyer et al.[15] that when working in high dimensions we loose the aspect of closeness between data points. In particular, as data increases in dimensionality, distance to the nearest data point has been shown to approach the distance to the farthest data point. Given these facts, dimension reduction seems therefore an essential step in our method. Reducing document vectors' dimension will allow to find dense clusters of documents more efficiently and accurately. There exists several techniques for reducing dimensionality such as Principal Component Analysis (PCA)[14] or t-Distributed Stochastic Neighbor Embedding (t-SNE)[27], and although these latter are well-known methods for reducing dimensionality, we use in our model a manifold learning technique for dimension reduction known as Uniform Manifold Approximation and Projection (UMAP)[20]. There are several benefits in using UMAP such as its speed (due to do fact that UMAP handles large datasets and high dimensional data without too much difficulty) or the fact that UMAP scales well in embedding dimension. Moreover, UMAP has shown to preserve more of the local and global features of high-dimensional data in lower projected dimensions. The main hyper-parameter that needs be chosen for UMAP is the *number of nearest neighbours*; this parameter is at the core of how the algorithm performs dimension reduction [20]. The *number of nearest neighbours* parameter is significant because it controls the balance between preserving global structure versus local structure in the low dimensional embedding. Augmenting the *number of nearest neighbours* puts the emphasis on global over local structure preservation and vice versa, decreasing the parameter puts emphasis on local structure preservation. We find that a *number of nearest neighbours* of 15-20 gives the best results in our

[3]USE stands for Universal Sentence Encoder

experiments, as larger values have a higher chance of neglecting the local features of the data, and on the other hand smaller values have a higher chance of ignoring the data's global structure.

5.3 Clustering of documents

The next step in our model consists in clustering the documents. There exists different types of clustering such as centroid-based clustering or distribution-based clustering, inter alia. Density-based clustering seemed an appropriate method for our model given the semantic-based document embedding method we employed in our model. Indeed, the goal of density-based clustering is to find areas of highly similar documents in the semantic space, which indicate an underlying topic. Our approach for the clustering process is based on the assumption that each cluster present in the semantic space is actually a topic. This is because, in our approach, clusters are dense areas of semantically similar documents, which is the idea of how topics are represented in a semantic space of embedded documents. Consequently, we consider each cluster detected by the clustering algorithm as topics. The challenge for the clustering process is that the document vectors will have varying density throughout the semantic space. Additionally there will be sparse areas where documents are highly dissimilar. This can be seen as noise, as there is no prominent underlying topic. The clustering of document embedding vectors in the semantic space is done using HDBSCAN, a density-based clustering algorithm developed by Campello et al.[7] that has obtained state-of-the-art results on clustering tasks. HDBSCAN is used to find the dense areas of document vectors, as it was designed to handle both noise and variable density clusters. Precisely, HDBSCAN assigns a label to each cluster of document vectors and assigns a noise label to all document vectors that are not in a dense cluster. Documents that are classified as noise can be seen as documents that do not contain a prominent topic. The rest of the documents belong to one topic. Unlike other clustering algorithms, HDBSCAN finds itself the number of topics, which corresponds to the number of dense areas of similar documents detected by the algorithm. As mentioned earlier, one main assumption behind our approach is that the number of dense areas of document vectors equals the number of prominent topics to be detected. This approach to find the number of clusters is practical in real-life applications since the number of topics in a set of corpus is unknown most of the time.

HDBSCAN has several hyper-parameters that determine how it performs clustering. Perhaps the most important parameter is *minimum cluster size* since this parameter is at the core of how the algorithm finds cluster varying density. As the name indicates, this parameter represents the smallest size of a cluster considered by the algorithm. We find that setting the number of *minimum cluster size* to 15 gives the best results, as larger values have a higher chance of merging unrelated document clusters.

For dynamic topic modeling, our model retrieves the topics at each timestamp using HDBSCAN : the latter performs clustering on the UMAP reduced document vectors. Figure 2 displays examples of clusters detected by our model using HDBSCAN.

5.4 Topic representation - First part

Topic representations are modeled based on the documents of each cluster where each cluster will be assigned to one topic. Our model uses class-based TF-IDF, a.k.a. c-TF-IDF, a variant of TF-IDF adapted to clusters instead of documents. We adopted the approach of Maarten Grootendorst [12] for the first representation of topics in our model, but adapted the c-TF-IDF calculation to our approach and main assumption which is that future documents are unknown in the present.

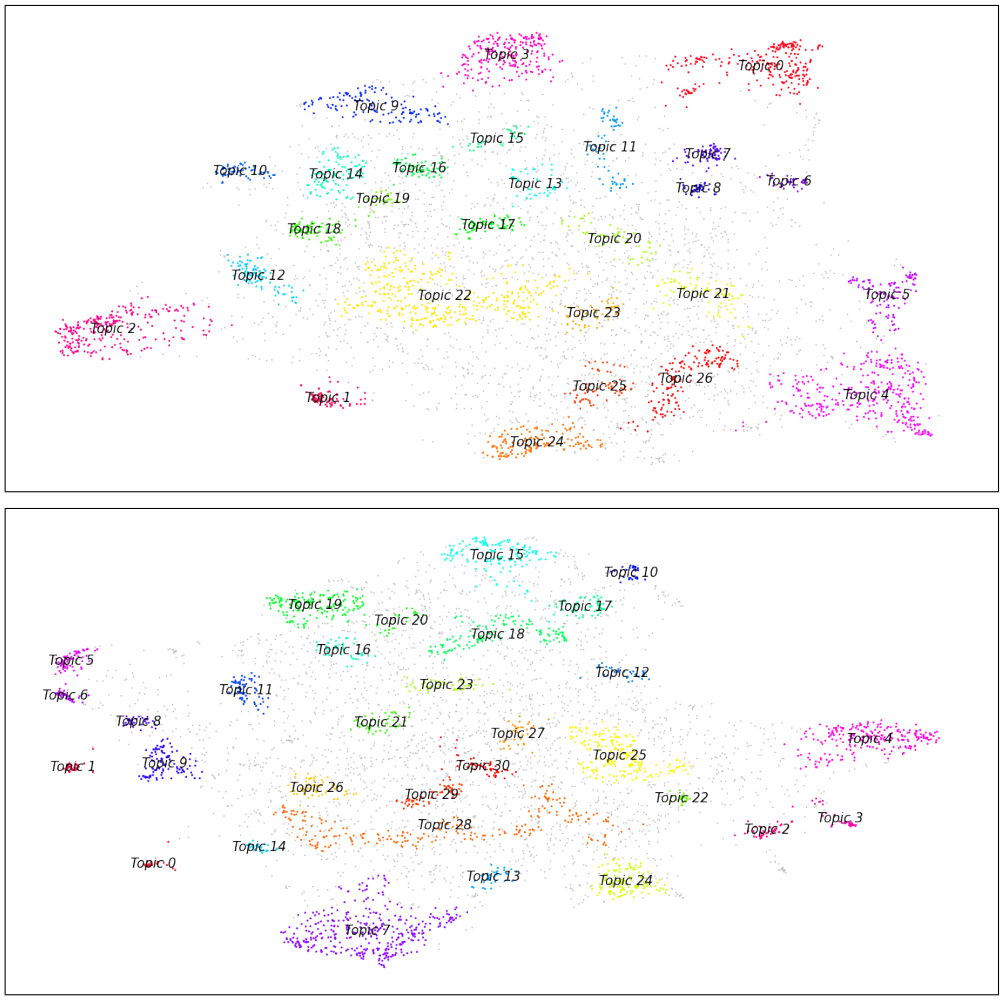

Figure 2: **Visualization of topics detected by our model using HDBSCAN**. Colored points represent detected topics and grey points are documents HDBSCAN has labeled as noise. The upper figure displays the 27 clusters detected by our model at timestamp $t = 0$. The lower figure displays the 31 clusters detected by our model at timestamp $t = 1$. As we can see, the number of topics can differ from a timestamp to another which implies that, aside from topics who evolve over time, there are topics that emerge and other topics that disappear over time.

Regular TF-IDF [13] for a word w in a document d is calculated by multiplying term frequency and inverse document frequency :

$$TFIDF(w, d) = TF(w, d) \cdot log\left(\frac{N}{count(d \in D : w \in d)}\right)$$

Where the term frequency models the frequency of a word w in document d. The inverse document frequency measures how much information word w provides to document d and is calculated by taking the logarithm of the number of documents in the corpus N divided by the total number of documents that contain w. Multiplying these two terms results in the TF-IDF score of a word in a document. The higher the score, the more relevant that word is in that particular document.

Following the approach of BERTopic [12] for topic representation, we use an adapted version of TF-IDF to clusters of documents. Our method for topic representation is analogous to that of BERTopic because we are interested in the importance of words in classes/topics and not documents. Topic words are defined as the most relevant and representative words of a particular topic. Hence, it makes sense to use c-TF-IDF for topic representation.

In the c-TF-IDF procedure, we start by concatenating all documents in a cluster as a single document. The reason for doing so is that we want to quantify the relevance of a word in a cluster instead of a document. Transforming classes into single documents facilitates the modification of TF-IDF to classes. Indeed, TF-IDF is adjusted to account for this representation by translating documents to clusters :

$$c - TFIDF(w, c) = TF(w, c) \cdot log\left(1 + \frac{A}{count(w \in c : c \in C)}\right)$$

Where the term frequency $TF(w, c)$ models the frequency of a word w in a class c. We recall that each class c is a collection of documents concatenated into a single document. The inverse document frequency is here replaced by a inverse class frequency, which measures how much information word w provides to a class c. Inverse class frequency is calculated by taking the logarithm of the average number of words per class A divided by the frequency of word w across all classes, and we add one to this fraction to ensure the positivity of the inverse class frequency.

Our model leverages c-TF-IDF to retrieve topic representation in dynamic topic modeling. At each timestamp, the model iteratively computes the c-TF-IDF score for each word of each cluster detected. Given a timestamp t, the set of clusters C_t corresponds to the clusters detected by the model through HDBSCAN at time t. Consequently, the average number of words per class A_t is calculated at each time t and is likely to differ from a time to another. This approach ensures the congruence of all c-TF-IDF terms with each timestamp. Our adapted version of c-TF-IDF to dynamic topic modeling results in a time-aware c-TF-IDF, given by :

$$c - TFIDF(w, c, t) = TF(w, c) \cdot log\left(1 + \frac{A_t}{count(w \in c : c \in C_t)}\right)$$

This procedure gives us relevant topic representations at all time. Moreover, this procedure is pragmatic and adapted to many real-life application.

5.5 Topic representation - Second part

The next step of our model consists in embedding topic words in order to obtain a new representation of topics. Specifically, we need a word embedding that respects the semantic of words where the distance between word vectors represents semantic association. Semantically similar words should be placed close together in the embedding space, and dissimilar words should be placed further from each other. This spatial representation of words is called a semantic space [11]. To learn embedded word vectors we use GloVe [22], a state-of-the-art word embedding model. GloVe is an unsupervised learning algorithm developed by Pennington et al. [22] for obtaining vector representations of words based on their semantic. The resulting topic representations are sets of points in a vector space. Our model will use these new representation of topics to quantify the similarity between topics, and therefore to analyze the evolution of topics. Figure 3 illustrates the process to obtain the new representation of topics.

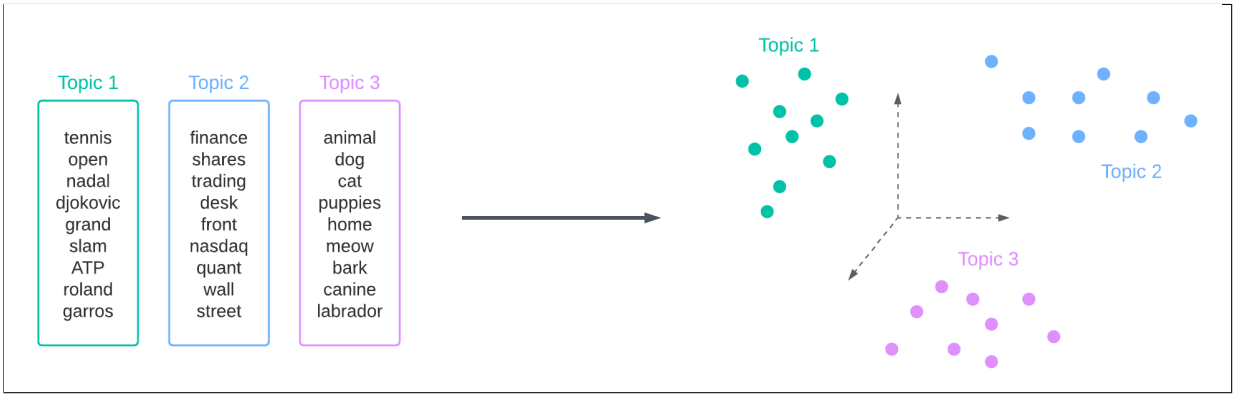

Figure 3: **Depiction of the process for obtaining new topic representation.** From the top words obtained with c-TF-IDF (*left side*), we apply a word embedding model on the top words to obtain a new representation of topics (*right side*) : topics are now represented by sets of points in a vector space.

5.6 Topic evolution

The last step of our model consists in analyzing the evolution of topics. In our model, we tackle the problem of topic evolution by focusing on the similarity between topics. Our approach is based on the idea that detected topics regarded as belonging to the same topic are necessarily similar either in content or representation, and vice versa topics that are different are dissimilar in content and representation. Given the new representation of topics retrieved in the previous step, the idea is to use a measure adapted to our new representation of topics to quantify the similarity between topics over time.

5.6.1 Chamfer distance

There exists several measures to compute the distances between sets of points such as [16][24]. In our model, we use the Chamfer distance (CD) [26] to quantify the similarity between topics. Chamfer Distance is a broadly adopted metric for measuring the similarity between point sets. The Chamfer Distance between two sets of points S_1 and S_2 is defined as :

$$CD(S_1, S_2) = \frac{1}{|S_1|} \sum_{x \in S_1} \min_{y \in S_2} \|x - y\|_2^2 + \frac{1}{|S_2|} \sum_{y \in S_2} \min_{x \in S_1} \|x - y\|_2^2$$

CD takes the distance of each points into account. Each point $x \in S_1$ finds its nearest neighbour in S_2 and vice versa. All the point-level pair-wise distances are averaged to produce the Chamfer distance.

5.6.2 CD threshold

The tracking of topics over time is one of the principal challenges in dynamic topic modeling. Several methods have been proposed for topic evolution as in [12][9][8]. However, even though there has been some great improvements in the analysis of topics over time, topic evolution remains a promising research area. In our research, we tried to find an efficient and pragmatic way to track topics across

time. When tackling the problem of topic evolution, we encounter one big challenge which is not to confound alteration of topics with evolution of topics. This is a convoluted problem because determining whether a topic has evolved or changed completely is not always intelligible, and is regarded as a human-level judging problem. In fact, there is no patent definition of topic evolution, hence there is always an ambiguity in what some would consider an evolution of topic. As one can imagine, this is a tricky decision for a machine to make since even humans would sometimes disagree on whether a topic has evolved or not.

As said earlier, we use the Chamfer distance (CD) to quantify the similarity between topics. Therefore, we want to use the CD to track topics over time. The idea is to set a threshold for determining whether two topics can be considered as belonging to the same topic or not. To set a threshold, we need to take into account certain facts. First of all, our model returns 22 topic words per topic which implicates that each topic is represented by 22 points in a vector space. Since CD relies on the word embedding vectors in our case, the number of words in common between two topics has an impact on their Chamfer distance. Two identical topics, i.e. topics containing the exact same words will have a CD close to 0. On the contrary, two topics totally different, that is topics that do not share any word in common, will have a high CD value. Setting a CD threshold to decide whether a topic has evolved or not is obviously a convoluted task. The smaller the threshold, the lesser the number of topics detected as similar and as a consequence our model will track fewer topics over time and could miss ground-truth topic evolution. Conversely, the higher the threshold, the higher the number of topics detected as similar. The risk in setting a high CD threshold is to detect too many fictitious similar topics, which would distort the model's performance for the tracking of topics. Hence, the threshold needs to be chosen carefully.

We conduct an experiment[4] to observe the distribution of CD values with respect to the number of topic words in common between two topics. The goal of this experiment is to see if the distribution of CD values indicate a certain value to consider for the threshold. In particular, we want to determine, using Chamfer distance, if the preservation of at least half of topic words is reasonable for identifying the evolution of topics. Since the CD value depends on the word embedding vectors and implicitly on the exact matching of words between topics, we conduct the following experiment : we create 2 sets of 22 words chosen randomly from the GloVe vocabulary[5] such that they share N words in common, that is the topics have N words in common (chosen randomly) and the rest of the words are different. We compute the Chamfer distance between the topics and iterate the experience 50,000 times. Therefore, we obtain 50,000 CD values of topics sharing N words in common and observe the distribution of CD values obtained. We iterate the experiment 22 times for each N between 0 and 22 (the total number of topic words in our model). Figure 5 displays some results of the experiment. The rest of the results can be observed in the appendix section C. In particular, we observe the CD distributions obtained with $N \pm 11$, which represents half of the topic words returned by our model. When observing the distributions of the cases where N is equal to 9, 10, 11 and 12 in figures 5 and 13, we can notice that a great proportion of CD values ranges between 300 and 500. In particular, we observe that topics that share $N = 9, 10$ words in common have CD values ranging mostly between 400 and 550. On the other hand, we observe that topics that have $N = 11, 12$ topics in common have CD values ranging mostly between 300 and 500. Moreover, we observe in figure 13 (from the appendices) that when N is greater than 11, most CD values are smaller than 500. Even in the cases where $N = 9, 10$, we see that an important amount of CD values are smaller than 500. Logically, as

[4]The notebook of the experiment is available at : https://github.com/sotodavid/Dynamic-Topic-Modeling---MVA/blob/main/CD_values_distribution.ipynb

[5]The GloVe vocabulary is available here : https://nlp.stanford.edu/data/glove.6B.zip

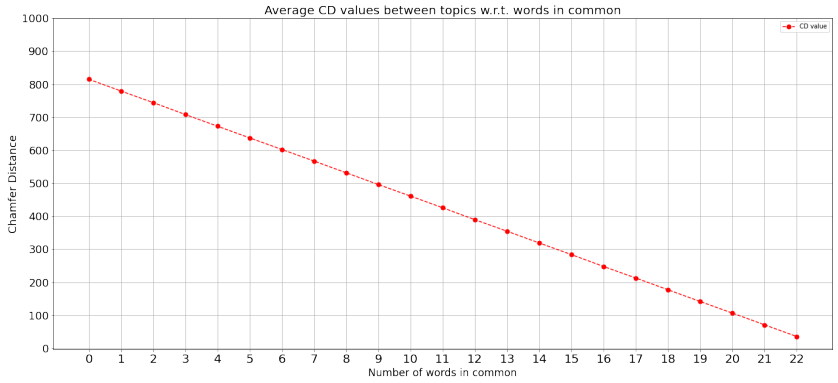

Figure 4: Average Chamfer Distance values between topics with respect to the number of words in common. For each number of words in common N, we randomly created topics sharing N words in common, we computed the CD between topics and report the average CD for each N.

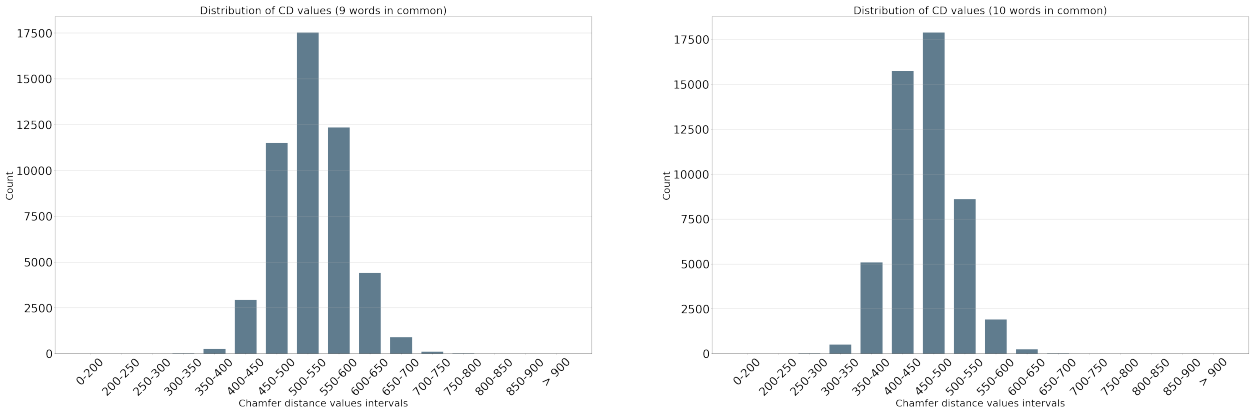

Figure 5: **Distribution of Chamber distance values.** The *left figure* is the distribution of CD between topics that have 9 words in common. The *right figure* is the distribution of CD between topics that share 10 words in common. As we can see in the *left figure*, a great proportion of topics have 9 words in common have a CD value greater than 500. In the *right figure*, we can see that a great proportion of topics that have 10 words in common have a CD value smaller than 500.

the number of words in common N between topics increases, the CD values decrease (we can observe this in the results in figure 13). In addition to the experiment, we display in figure 4 the average CD values depending on the number of words in common between topics. For each number of common words N, ranging from 0 to 22, we report the average CD values over 50,000 values obtained. The average CD curve shows that for $N = 9, 10, 11$, the average CD value ranges between 400 and 500. Hence, any threshsold between 400 and 500 could have an important impact on the model's capacity to detect evolution of topics. With a threshold of 400, the model would tend not to detect topics containing 9 or 10 words in common as similar. On the other hand, with a threshold of 500 our model would have a higher chance of detecting ground-truth evolution of topics that have 9 or 10 words in common (as they can exist), but it would also augment the risk of detecting bogus evolution of topics.

Finally, based on the obtained results, we set a Chamfer distance threshold of 500 in our model. This is because we want to our model to uncover the most of ground-truth evolution of topics, even the challenging ones. Hence, in our model, two topics with a CD smaller than 500 will be considered

as belonging to the same topic. On the other hand, topics having a CD value greater than 500 will be considered by our model as different topics. This will allow our model to follow topics over time and at the same time to identify new topics as well as topics disappearing.

5.6.3 The tracking of topics

To track topics over time, our model uses the Chamfer distance. For each topic at each timestamp t, our model computes the Chamfer distance with all topics of the following timestamp $t+1$, and identifies the nearest topic thanks to the CD values. Our model will consider the topic (from time $t+1$) with the smallest CD value as being the topic with the highest chance of being an evolution of the topic of reference. Our model identifies the closest topic as being the topic that has the smallest CD with the topic of reference. Figure 6 depicts the process of tracking topics using the Chamfer distance. Our model finds the closest topic to the topic of reference and decides whether or not these topics are similar based on their distance.

This procedure enables our model to track topics over time. Storing the distance values enables our model to follow and observe a topic over time. Moreover, the tracking of topics over time performed by our model allows to observe the evolution of topic representations. Figure 11 from the appendices (section B.2) displays examples of topic evolution.

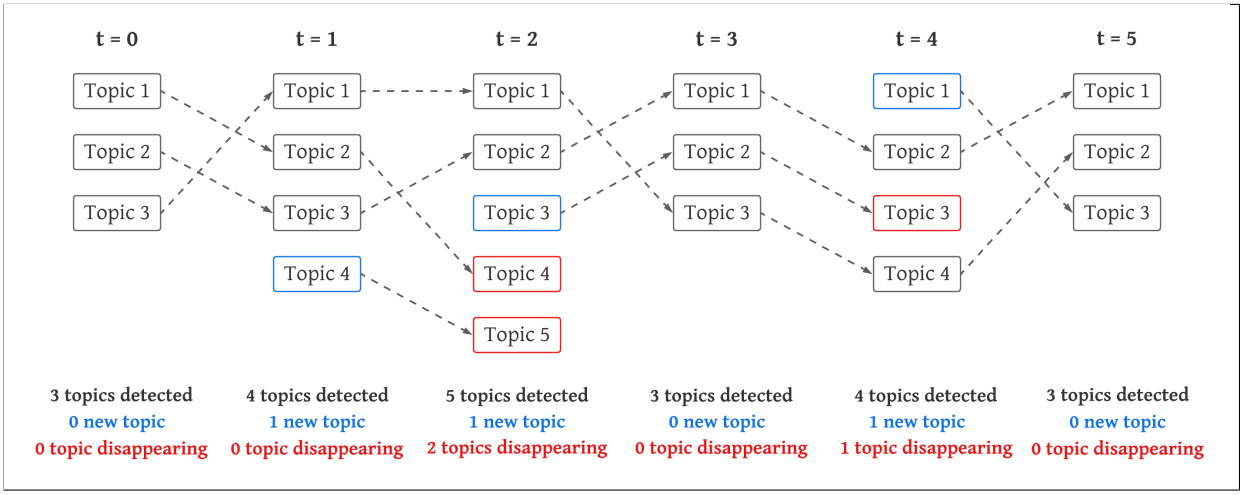

Figure 6: **Depiction of the tracking of topics using Chamfer distance.** At each timestamp t, our model detects a certain number of topics. Between two timestamp t and $t+1$, after having retrieved the new representation of topics, the model computes the Chamfer distance between each topic of the two timestamps. The established threshold of 500 enables us to track topics over time. In the figure, each arrow represent the evolution of a particular topic, an evolution that has been detected by the model thanks to the CD threshold. The threshold allows also to detect new topics (*blue topics*) as well as topics disappearing (*red topics*). For instance, we are able to follow *topic 1* from timestamp $t = 0$ over time. The latter corresponds to *topic 2* at time $t = 1$ and to *topic 4* at time $t = 2$, time where it appears for the last time before disappearing.

6 Experiments

We conduct extensive experiments on several benchmark text datasets to evaluate the performance of our model against BERTopic, a state-of-the-art dynamic topic model that inspired our model, to show that our approach is at least as good as BERTopic's approach for dynamic topic modeling (if not better).

6.1 Overview

We conduct three experiments to evaluate different aspects of dynamic topic modeling. The first experiment consists in evaluating the quality of topics detected by the dynamic topic models. The second experiment aims at evaluating the models' capacity to detect the ground-truth number of topics over time. The third experiment aims to evaluate dynamic topic models' ability to track topics over time. In particular, we are interested in the models' ability to detect emerging topic and vanishing topics over time.

6.2 Datasets

Our experiments are conducted on three datasets. Two of them are widely-used benchmark text datasets for dynamic topic modeling, namely Trump's tweets dataset (**Trump**)[6] and United Nations general debates dataset (**UN**)[7]. The Trump dataset is composed of thousands of tweets written by Donald Trump between the years 2009 and 2021, year where his account was suspended by Twitter. The UN dataset, as its name indicates, is composed of scripts of the general debates that took place at the United Nations sessions between the years 2006 and 2015. The last dataset we use in our experiments is the 20 News Groups dataset (**20NG**)[8], a widely-used dataset for topic modeling (and not dynamic topic modeling). Unlike Trump dataset, UN dataset and most dynamic topic modeling datasets, 20NG contains ground-truth topic labels (20 topics in total) but is originally used for static topic modeling. For the second and third experiments, we will need a dataset with ground-truth topic labels to evaluate dynamic topic models. For these reasons, we adapt the 20NG dataset for dynamic topic modeling by adding timestamps to the data. When adding the timestamps to the 20NG dataset, we simulate evolution of topics over time because we are interested in evaluating the performances of dynamic topic models on tracking topics over time. In particular, we simulate the emergence of topics and the disappearance of topics over time. This way, we obtain a dynamic topic modeling dataset containing ground-truth evolution of topics that we can observe and use to evaluate the performance of dynamic topic models. The statistics of the datasets used in the experiments are shown in Table 2.

	Number of docs	Number of timestamps	Number of labels
Trump	45355	13	unknown
UN	50422	10	unknown
20NG (adapted for dtm)	18846	15	20

Table 2: Statistics of the datasets

[6] https://www.thetrumparchive.com/faq
[7] https://runestone.academy/runestone/books/published/httlads/_static/un-general-debates.csv
[8] http://qwone.com/~jason/20Newsgroups/

6.3 Notations

For the second and third experiments we distinguish four types of topics. *Remaining topics (RT)* refer to topics still existing from a timestamp t to the next timestamp $t+1$. We define *emerging topics (ET)* as topics appearing progressively over time until becoming prominent topics. As mentioned before, the detection of *emerging topics* is of particular interest to us. *Vanishing topics (VT)* are defined as topics disappearing from a timestamp t to the next timestamp $t+1$. We will refer to all topics detected at a timestamp as *overall topics (OT)*. Hence, overall topics contain remaining topics, emerging topics and vanishing topics.

Moreover, in this section we refer to *top k words* of a topic as the k words most representative of a topic, among the topic words of that topic. In our model we retrieve the topic words using c-TF-IDF. The k words with the highest c-TF-IDF scores, among the topic words of a topic, are considered as the *top k words* of that particular topic.

6.4 Evaluation metrics

In the first experiment, we report Topic Coherence (TC) and Topic Diversity (TD) as performance metrics for topic quality. Topic Coherence measures the semantic coherence in the most significant words (top words) of a topic, given a reference corpus. We apply the widely-used Normalized Pointwise Mutual Information (NPMI) [5] metric (see section 4.3), computed over the top 22 words (which correspond to the topic words) of each topic. To evaluate the topic quality of each model, we compute the NPMI score of each topic detected at each timestamp on several runs and report the average score. TC takes values in $[-1, 1]$, where -1 indicates incoherent topic (semantically-wise); 1 indicates perfect semantic association between top words, hence coherent topics. Topic Diversity, as its name implies, measures how diverse the discovered topics are. We define topic diversity to be the percentage of unique words (Dieng et al., 2020 [10]) in the top 22 words of a topic. To evaluate the topic diversity of a model, we compute the percentage of unique words (PUW) [10] at each timestamp on several runs and report the average PUW score. TD close to 0 indicates redundant topics while TD close to 1 indicates more varied topics. For both metrics, higher values indicate better performance.

For the second experiment, we are interested in evaluating the number of topics detected by the models. For this purpose, we use a criteria of difference as the metric for the evaluation. The criteria measures the difference between the number of topics detected and the ground-truth number of topics. In particular, the criteria indicates the average number of topics non-detected by a model or excedent in the detction of topics for a model, compared to the ground-truth number of topics. The criteria is calculated at each timestamp and we report the average values obtained for each model. The criteria is calculated for the overall number of topics detected by each model ($criteria_{OT}$), but also for the remaining topics ($criteria_{RT}$), emerging topics ($criteria_{ET}$) and vanishing topics detected ($criteria_{VT}$), and are defined as follows :

$$Criteria_{OT} = |\ \#\ topics\ detected\ - \#\ Groundtruth\ topics\ |$$

$$Criteria_{RT} = |\ \#\ RT\ detected\ -\ \#\ Groundtruth\ RT\ |$$

$$Criteria_{ET} = |\ \#\ ET\ detected\ -\ \#\ Groundtruth\ ET\ |$$

$$Criteria_{OT} = |\ \#\ OT\ detected\ -\ \#\ Groundtruth\ OT\ |$$

In the third experiment, we evaluate the models' capacity to detect ground-truth topics using Precision and Recall metrics. Indeed, in this experiment we are interested in evaluating the relevance of the topics discovered by the model based on ground-truth collections of topics. In this experiment, precision indicates the proportion of veracious topics detected among the topics detected by a model. On the other hand, Recall indicates the percentage of veracious topics discovered by a model among all veracious topics to be discovered. The precision and recall formulas, adapted to our experiments, are given by :

$$Precision = \frac{|\{Groundtruth\ topics\} \cap \{Detected\ topics\}|}{|\{Detected\ topics\}|}$$

$$Recall = \frac{|\{Groundtruth\ topics\} \cap \{Detected\ topics\}|}{|\{Groundtruth\ topics\}|}$$

6.5 Implementation details

The parameters of our model for the experiments are as follows. We use the Universal Sentence Encoder as our document embedding model. The dimensionality reduction is done with the UMAP algorithm with parameters *number of neighbors :* 15, *number of components :* 5 and *metric*: cosine. The clustering process is done with HDBSCAN, set with the following parameters : *mininimum cluster size*: 15, *metric*: euclidean, *cluster selection method*: eom. The first topic representation, done with c-TF-IDF, retrieves 22 top words per topic detected.

6.6 Results

6.6.1 Topic quality

The first experiment consists in evaluating the quality of topics detected by dynamic topic models. We want to show that the our model detects topics whose quality are as good (or at least comparable) as the quality of topics detected by BERTopic, a state-of-the-art dynamic topic model. As mentioned earlier, Topic Coherence (TC) and Topic Diversity (TD) are widely-used metrics to evaluate the quality of topics. In our experiment, TC is evaluated using NPMI. The latter emulates human judgment with respect to the "coherence" of a topic. It focuses on words' contexts to give a coherence score to a topic. The coherence score measures how similar words in a topic are to each other, i.e. how "coherent" topics are. The measure ranges between -1 and 1, where 1 indicates a perfect association. TD is measured by the percentage of unique words for all topics, the measure ranges between 0 and 1 where 0 indicates redundant topics and 1 indicates varied topics. For each model, for each dataset, the TC and TD scores were calculated at each timestamp and averaged. We report the average values obtained on 4 runs for each model on each dataset. Hence, each score represents the average of 40 values on UN dataset, 52 values on Trump dataset and 60 values on 20NG dataset. Table 3 displays the TC and TD results obtained on the three datasets.

Model	Trump dataset		United Nations		20NewsGroup	
	TD	TC	TD	TC	TD	TC
BERTopic	**0.913**	**0.147**	0.823	-0.037	0.837	0.021
Our model	0.871	0.137	**0.919**	**0.268**	**0.862**	**0.124**

Table 3: Topic Quality results

As we can see from table 3, there is no model that outstands on the three datasets. BERToppic obtains slighlty better results on Trump dataset. However, on UN and 20NG datasets our model outperforms BERTopic. Based on the results of this experiment, we can conclude that the topics detected by our model are as coherent and diverse as the topics detected by BERTopic.

6.6.2 Number of detected topics

The second experiment focuses on the number of topics detected by the models. We want to show that our model detects a number of topics similar (if not identical) to the ground-truth number of topics to be discovered. This is because in our evaluation of dynamic topic models we are interested in observing the models' capacity to detect ground-truth topics in a dataset, and not just topics "invented" by the models. We want to see if our model tends to discover too much topics at each timestamp or inversely if our model misses the detection of some topics when performing dynamic topic modeling. This aspect of dynamic topic modeling is important to us. Indeed, we would consider a model tending to detect too much topics with respect to the ground-truth number of topics not to be efficient, even if some of the discovered topics are ground-truth topics because it would mean that most of the discovered topics are "invented" by the models and therefore unnecessary to detect.

For this experiment we need a dynamic topic modeling dataset containing ground-truth topic labels. As far as we know, such dataset does not exist. For that reason, for this experiment we use 20NG, a well-known topic modeling dataset containing ground-truth topic labels, and adapt it to dynamic topic modeling by adding timestamps. Since we are interested in evaluating the models' capacity to detect emerging topics as well as topics disappearing over time, we add timestamps to the 20NG dataset and simulate evolution of topics such as emerging topics and vanishing topics.

As mentioned before, in this experiment we focus on the number of detected topics. We want to see which model approximates better the ground-truth (GT) number of topics over time. At each timestamp, we compare the number of detected topics with the GT number of topics using the criteria defined in section 6.4, which calculates the difference between the number of topics detected and the ground-truth value. For each model, we compute the criteria of difference at each timestamp. We compare the number of remaining topics detected, the number of emerging topics detected and the number of vanishing topics detected with the GT values, at each timestamp. We average the values over all timestamps and report the average over several runs. The results are displayed in table 4.

Model	# OT	# RT	# ET	# VT
BERTopic	27.933	21.857	5.214	4.357
Our model	**5.067**	**6.350**	**2.785**	**2.285**

Table 4: Criteria results of the difference between the number of detected topics and the GT number of topics. We calculated the criteria for OT, RT, ET and VT topics.

As we can see in Table 4, in average our model approximates better the GT number of topics compared to BERTopic. For instance, at each timestamp our model detects 5.067 more/less topics than the GT number of topics whereas BERTopic detects 27.933 more/less topics than the ground-truth number of topics, which is not so good compared to our model. With our approach, the number of detected topics is more accurate. We observe the same result on the remaining (RT), emerging

(ET) and vanishing (VT) topics. Our model approximates better the GT number of remaining, emerging and vanishing topics compared to BERTopic. A plausible explanation of the results could be the difference in BERTopic's approach and our approach. As mentioned earlier, our model performs dynamic topic modeling by repeating a certain process at each timestamp. This is because we analyze each new set of documents at each timestamp independently of the future documents since they are unknown to us. Hence, our model performs a clustering at each timestamp, as new documents arrive, independently of the clusters found in the previous timestamps. This enables our model to find unbiased clusters, and therefore the number of topics detected is likely to approximate the GT number of topics to be detected. On the other hand, BERTopic's approach for dynamic topic modeling is special as it supposes that documents coming in the future are known from the beginning of the dynamic process, and therefore in this approach the global representation of documents prevails against the local representation of documents (that is the representation of documents at each timestamp). BERTopic performs dynamic topic modeling by applying BERTopic's topic modeling approach once on the global representation of documents to find the global topics, and then updates the representation of topics at each timestamp. Therefore BERTopic performs clustering only once, on the global representation of documents. The consequence of this approach is that BERTopic is likely to detect an excessive number of topics as the global representation of topics is a lot different than the local representations of topics and contains a lot more document embeddings. Figure 8 is a visualization of the global representation of 20NG documents. As we can see, the global representation is a lot different than the local representations (figure 9), which implies that BERTopic is likely not to get the structure of the document embeddings at each timestamp. These facts can explain the striking difference between the number of topics detected by BERTopic and the number of topics detected by our model, and hence the criteria values obtained by the models.

Figure 7 displays two examples of evolution of the number of topics detected by BERTopic and our model, versus the ground-truth evolution of topics, on 20NG. The green curves correspond to the evolution of the number of topics detected by BERTopic. The blue curves corresponds to the evolution of the number of topics detected by our model. The red curves correspond to the ground-truth values. We can observe that, even though the number of topics detected by our model is not perfect, visually there is a patent amelioration in the approximation of the GT number of topics with our model compared to BERTopic. Based on the results from table 4 and figure 7, we can conclude that our approach allows a better approximation of the GT number of topics at each timestamp. More plots are available in the appendix section 10.

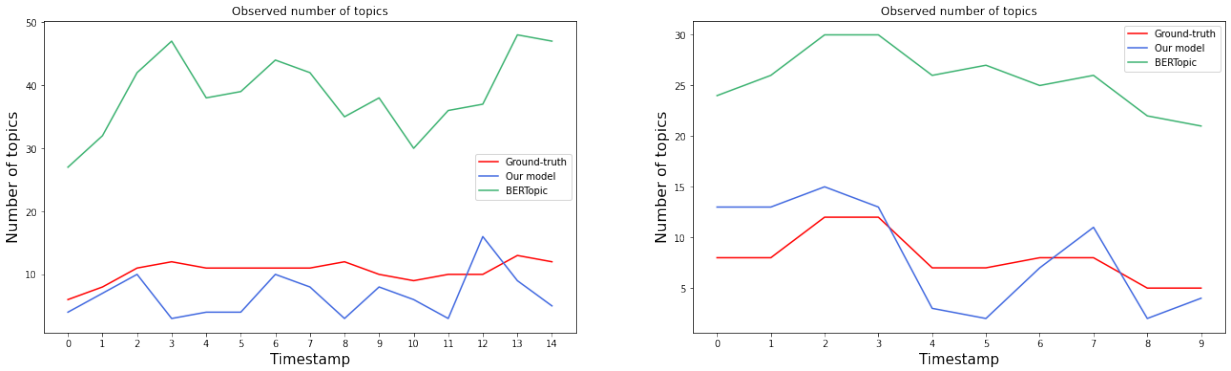

Figure 7: **Examples of the evolution of number of topics over time**. The *green* and *blue* curves correspond respectively to the results obtain with BERTopic and our model. The red curve corresponds to the GT values.

6.6.3 Detection of ground-truth topics

The last experiment consists in analysing the relevance of detected topics. In this part we focus on the veracity of the topics detected. This experiment aims at evaluating the models' ability to detect ground-truth topics over time. As mentioned earlier, this aspect of dynamic topic modeling is important to us because it reveals the models' capacity to discover meaningful topics. For this experiment we use the 20NG dataset adapted to dynamic topic modeling because this dataset contains ground-truth topic labels we can use for the evaluation.

In this experiment, we use the precision and recall metrics to evaluate the models' capacity to detect ground-truth topics. Precision, as defined in section 6.4, gives the percentage of detected topics that are also veracious topics, that is ground-truth topics. The higher the precision, the more efficient is the model. Indeed, a precision of 1 indicates that all discovered topics are ground-truth topics, whereas a precision of 0 implies that the model has not detected any veracious topics. On the other hand, recall gives the percentage of veracious topics detected by a model. A recall of 1 indicates that all ground-truth topics are discovered by the model. A recall of 0 means that no ground-truth topics are discovered. In the experiment, the procedure to decide whether a detected topic is a veracious topic or not is as follows : since we know the ground-truth topics and their top 20 words, we use the matching of top words and their embeddings to decide whether a detected topic is a veracious topic or not. A detected topic is said to be a veracious topic if it has at least 10 words (among the top 20 topic words) in common with a GT topic and if the Chamfer distance between this GT topic and the topic of reference is smaller than 500. Since we know the ground-truth topic words, we can apply this rules in our experiment. For this experiment, our model returns 20 topic words per topic discovered.

For each model we compute the precision and recall metrics at each timestamp and average the results. We report the average precision and recall values obtained for each model on the 20NG dataset on 4 runs. Beside the overall topics, we calculate the models' precision and recall on the detection of remaining topics, emerging topics and vanishing topics. The results are displayed in table 5.

Model	**Overall topics**		**Remaining topics**		**Emerging topics**		**Vanishing topics**	
	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall
BERTopic	0.168	**0.590**	0.195	**0.620**	0.032	0.312	0.023	0.255
Our model	**0.624**	0.361	**0.862**	0.397	**0.114**	**0.513**	**0.145**	**0.375**

Table 5: Precision and recall results on the detection of veracious topics. We distinguish four types of detected topics : overall topics, remaining topics, emerging topics and vanishing topics.

The results from table 5 are ambiguous. There is no model that outstands for the detection of GT topics in general and especially for the detection of GT remaining topics. The precision results on the detection of remaining topics and overall topics indicate that our model is more precise when detecting these type of topics. For instance, in average, more than 86% of topics detected by our model as remaining topics are actual ground-truth remaining topics, whereas only 19% of BERTopic's detection of remaining topics are veracious. However, the recall results seem to indicate that BERTopic detects a greater porportion of ground-truth topics, in particular a greater proportion of ground-truth remaining topics. These results can be explained by the fact that BERTopic tends

to detect way more topics than the GT number of topics at each timestamp, and therefore many topics detected by BERTopic turn out to be bogus topics. As explained earlier, due to its approach for dynamic topic modeling BERTopic tends to discover an excessive amount of topics at each timestamp. On the other hand, our model tends to detect a number of topic that remains close to the GT number of topics. However, the recall metric shows that our model "only" detects 36% of the ground-truth topics and 39% of GT remaining topics, whereas BERTopic's recall is around 60% meaning that BERTopic detects around 60% of ground-truth topics and in particular 60% of GT remaining topics. By detecting an excessive amount of topics, BERTopic diminishes its precision but augments its capacity to detecting ground-truth topics in general.

For the detection of GT emerging topics and GT vanishing topics, we can see from table 5 that our model outperforms BERTopic. In average, our model obtains a better precision and recall for the detection of GT emerging topics and GT vanishing topics, meaning that our model discovers more GT emerging and vanishing topics than BERTopic, and is more accurate in the detection. As we can observe in table 5, BERTopic obtains really poor precision for the detection of emerging topics and vanishing topics. Its recall is not great either. These results are the aftermath of BERTopic's approach where global representation of documents prevails over local representation. This causes the model to miss the alterations in the representation of documents at each timestamp. By looking at the global representation of 20NG in figure 8 and the local representation in figure 9, it is patent any model that focuses on the global representation is likely to miss the changes of topics. Moreover, the global representation could be misleading for analysing the evolution of topics. Indeed, when looking at figure 8, we can observe hodgepodges of topics in the representation, whereas in the local representations from figure 9 the topics are more spaced between them. Hence, any model that relies only on the global representation like BERTopic is going to miss the emergence of topics and disappearance of topics.

This experiment has shown that our model out performs BERTopic in the detection of ground-truth emerging topics and vanishing, and even though our model do not surpass BERTopic in the detection of remaining topics and topic in general, the obtained results are promising since. This experiment has also enabled us to assess the models' capability to discover relevant topics and at the same time to assess the models' ability not to "invente" topics. The results have shown that our model detects less bogus topics than BERTopic.

7 Conclusion

In this project, we present an unsupervised learning algorithm that performs dynamic topic modeling using a semantic space of embedded documents and a semantic space of word vectors. We have shown that the semantic space of embedded documents is a continuous representation of documents that allows for the discovery of topics from dense areas of highly similar documents. Our model also allows for comparing similarity between topics based on distance in the semantic space of topic words. The proposed model is an amelioration of BERTopic's extension to dynamic topic modeling. Indeed, our model uses part of BERTopic's approach such as document embedding or topic representation through c-TF-IDF, but unlike BERTopic, our apporach is based on the assumption that we do not know the sets of documents that will come in the future. This assumption changes everything in our approach compared to BERTopic. Due to this practical assumption, the model we propose iterates a topic modeling process at each timestamp but adds two crucial steps that enable the tacking of topics over time in a practical and efficient way. Firstly, our model extracts a new representation of topics using the topic words retrieved with c-TF-IDF. Afterwards, the model uses the new topic representations to measure the similarity between topics using Chamfer distance. This method allows the tracking of topics over time and, in particular, allows the discovery of new topics as well as the detection of topics disappearing over time.

Moreover, we have proposed a novel method for evaluating dynamic topic models that involves using datasets with ground-truth topic labels to assess the models' ability to detect ground-truth topics. The evaluation we propose enables to measure the real amount of information gained when using the dynamic topic models. This evaluation measures the models' capacity to detect ground-truth topics but also the models' ability not to detect bogus topics, that is topics that are "made-up" by the models and therefore unnecessary to detect. Our results show that our model detects veracious topics over time more efficiently than BERTopic. Specifically, the results have shown that our model outperforms BERTopic in the detection of emerging topics and vanishing topics.

There are several advantages in using our model over traditional dynamic topic modeling methods like D-LDA. The primary advantages are that our model automatically finds the number of topics and finds topics that are more informative and representative of the corpus. Moreover, thanks to the c-TF-IDF procedure, stop-word lists are not required to find informative topic words, making it easy to use on a corpus of any domain or language. Furthermore, our approach is well suited for many real-life applications of dynamic topic modeling. Inter alia, our model has no problem in performing dynamic topic modeling as new documents appear over time, unlike other models such as BERTopic.

8 Acknowledgement

I would like to acknowledge my indebtedness and render my warmest thanks to my supervisors Dr. Alain Rakotomamonjy and Dr. Alberto Lumbreras who made this work possible. Their friendly guidance and expert advice have been invaluable throughout all stages of the work. I also would like to thank them for their patience and assistance during this internship. All the discussions I had with Alain and Alberto have contributed greatly to my scientific improvement.

I want to thank my academic advisor, Dr. Michael Arbel, for accepting to be my referee for this internship and for having taken the time for monthly meetings during my internship, where mr. Arbel gave me valuable advices for my research work.

Bibliography

[1] Charu C. Aggarwal, Alexander Hinneburg, and Daniel A. Keim. On the surprising behavior of distance metrics in high dimensional space. In *Lecture Notes in Computer Science*, pages 420–434. Springer, 2001.

[2] David M. Blei. Introduction to probabilistic topic models. 2010.

[3] David M. Blei. Probabilistic topic models. *Communications of the ACM*, 55(77-84):91–105, 2012.

[4] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. In *Journal of Machine Learning Research 3(Jan):601-608.*

[5] G. Bouma. Normalized (pointwise) mutual information in collocation extraction. In *From Form to Meaning: Processing Texts Automatically, Proceedings of the Biennial GSCL Conference 2009*, volume Normalized, pages 31–40, Tübingen, 2009.

[6] Jordan Boyd-Graber, Yuening Hu, and David Mimno. Applications of topic models. *Foundations and Trends¬Æ in Information Retrieval*, 11(2-3):143–296, 2017.

[7] Ricardo J. G. B. Campello, Davoud Moulavi, and Joerg Sander. Density-based clustering based on hierarchical density estimates. In Jian Pei, Vincent S. Tseng, Longbing Cao, Hiroshi Motoda, and Guandong Xu, editors, *Advances in Knowledge Discovery and Data Mining*, pages 160–172, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[8] Clément Christophe, Julien Velcin, Jairo Cugliari, Manel Boumghar, and Philippe Suignard. Monitoring geometrical properties of word embeddings for detecting the emergence of new topics. *hal*, hal-03699173, 2021.

[9] Adji B. Dieng, Francisco J. R. Ruiz, and David M. Blei. The dynamic embedded topic model. *CoRR*, arXiv:1907.05545, 2019.

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

[11] Thomas L. Griffiths, Mark Steyvers, and Joshua B. Tenenbaum. Topics in semantic representation. *Psychological Review*, 114(2):211–244, 2007.

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

[13] Thorsten Joachims. A probabilistic analysis of the rocchio algorithm with tfidf for text categorization. technical report, carnegie-mellon univ pitts- burgh pa dept of computer science. 1996.

[14] I.T. Jolliffe. *Principal Component Analysis*. Springer Verlag, 1986.

[15] Raghu Ramakrish-nan Kevin Beyer, Jonathan Goldstein and Uri Shaft. When is "nearest neighbor" meaningful? In *International conference on database theory*, pages 217–235. Springer, 1999.

[16] Matt J. Kusner, Yu Sun, Nicholas I. Kolkin, and Kilian Q. Weinberger. From word embeddings to document distances. *ICML*, WCP volume 37, 2015.

[17] John Lafferty and David Blei. Correlated topic models. In Y. Weiss, B. Schölkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems*, volume 18. MIT Press, 2005.

[18] Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In Eric P. Xing and Tony Jebara, editors, *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 1188–1196, Bejing, China, 22–24 Jun 2014. PMLR.

[19] Jon Mcauliffe and David Blei. Supervised topic models. In J. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems*, volume 20. Curran Associates, Inc., 2007.

[20] Leland McInnes, John Healy, Nathaniel Saul, and Lukas Grossberger. Umap: Uniform manifold approximation and projection. *The Journal of Open Source Software*, 3(29):861, 2018.

[21] David Mimno, Hanna M. Wallach, Jason Naradowsky, David A. Smith, and Andrew McCallum. Polylingual topic models. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 880–889, Singapore, August 2009. Association for Computational Linguistics.

[22] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014.

[23] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019.

[24] Yossi Rubner, Carlo Tomasi, and Leonidas Guibas. The earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision*, 40:99–121, 11 2000.

[25] Silvia Terragni, Ismail Harrando, Pasquale Lisena, Raphael Troncy, and Elisabetta Fersini. One configuration to rule them all? towards hyperparameter transfer in topic models using multi-objective bayesian optimization. *cs.CL*, abs/2022.10825, 2022.

[26] Junzhe Zhang-Tai WANG Ziwei Liu Dahua Lin Tong Wu, Liang Pan. Density-aware chamfer distance as a comprehensive metric for point cloud completion. In *In Advances in Neural Information Processing Systems (NeurIPS), 2021*, 2021.

[27] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *The Journal of Machine Learning*, 3, 2008.

[28] Chong Wang, David M. Blei, and David Heckerman. Continuous time dynamic topic models. *CoRR*, abs/1206.3298, 2012.

A 20NewsGroups dataset

A.1 20NewsGroups dataset details

The following table 6 contains the 20NG ground-truth topics' details. We recall that for our experiments in section 6, we adapted the 20NG dataset to dynamic topic modeling by adding timestamps to the data. The goal was to simulate evolution of topics over time to assess the model's efficiency in detecting ground-truth topics.

Topic ID	Topic name	Top 5 words
0	alt.atheism	atheism, atheists, god, keith, islam
1	comp.graphics	graphics, image, jpeg, images, gif
2	comp.os.ms-windows.misc	max, os,ms, misc, windows
3	comp.sys.ibm.pc.hardware	scsi, drive, ide, pc, hard
4	comp.sys.mac.hardware	mac, apple, quadra, nubus, lc
5	comp.windows.x	window, motif, widget, server, mit
6	misc.forsale	sale, offer, shipping, condition, price
7	rec.autos	cars, autos, engine, oil, ford
8	rec.motorcycles	bike, dod, motorcycle, riding, helmet
9	rec.sport.baseball	baseball, season, team, players, games
10	rec.sport.hockey	hockey, team, game, play, nhl
11	sci.crypto	key, encryption, clipper, db, chip
12	sci.electronics	circuit, voltage, amp, power, electronics
13	sci.medical	medical, health, disease, cancer, patients
14	sci.space	space, nasa, launch, shuttle, orbit
15	society - religion.christian	god, jesus, church, christians, christ
16	talk.politics.guns	gun, firearms, weapons, batf, fbi
17	talk.politics.middle east	israel, armenians, jews, turkey, arab
18	talk.politics.misc	president, cramer, government, jobs, clinton
19	talk.religion.misc	christian, jesus, god, morality, objective

Table 6: 20NewsGroups ground-truth topics details

A.2 Visualization of 20NG embedded documents - Global representation

Figure 8 displays the global representation of 20NG's embedded documents, that is the representation of all documents composing the 20NG dataset without taking into account the timestamps. The embedded documents are displayed along with their ground-truth topics. Each color represents one ground-truth topic. The global representation of 20NG has been retrieved using Sentence-BERT, the same way BERTopic retrieves the 20NG global representation.

In figure 8, we can observe in many parts of the embedding space hogdepodges of topics. For instance, topics 1, 2, 3, 4, 5, 6, and 12 (displayed in the upper right part of the figure) are literally mixed in the space. The embedded documents composing these topics are not far enough from the other topics' embedded documents to distinguish clearly these topics. Topics 19, 0 and 15 also form a hodgepodge.

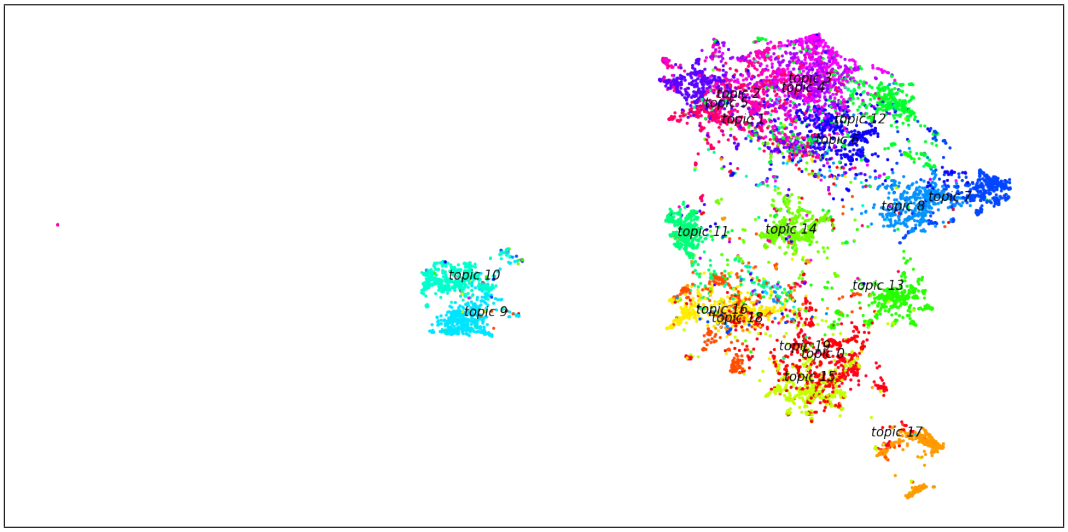

Figure 8: **Visualization of 20NG's ground-truth topics in the semantic space of embedded documents.** the representation was retrieved using Sentence-BERT. Each color is assigned to one ground-truth topic. Each document belongs to one topic. This is the global representation of 20NG documents in the semantic space.

Performing clustering on the global representation of embedded documents is very challenging. Moreover, when looking at figure 9, there is a huge difference between the global representation of 20NG and local representations of 20NG. It is obvious that the clusters detected on the 20NG global representation are going to differ from the clusters detected on 20NG local representations in terms of content, size, forms and number of detected topics.

By looking at figure 8 and figure 9, we understand why BERTopic [12] has a hard uncovering the 20NG ground-truth topics and especially why BERTopic detects way too much topics.

A.3 Visualization of 20NG embedded documents - Local representation

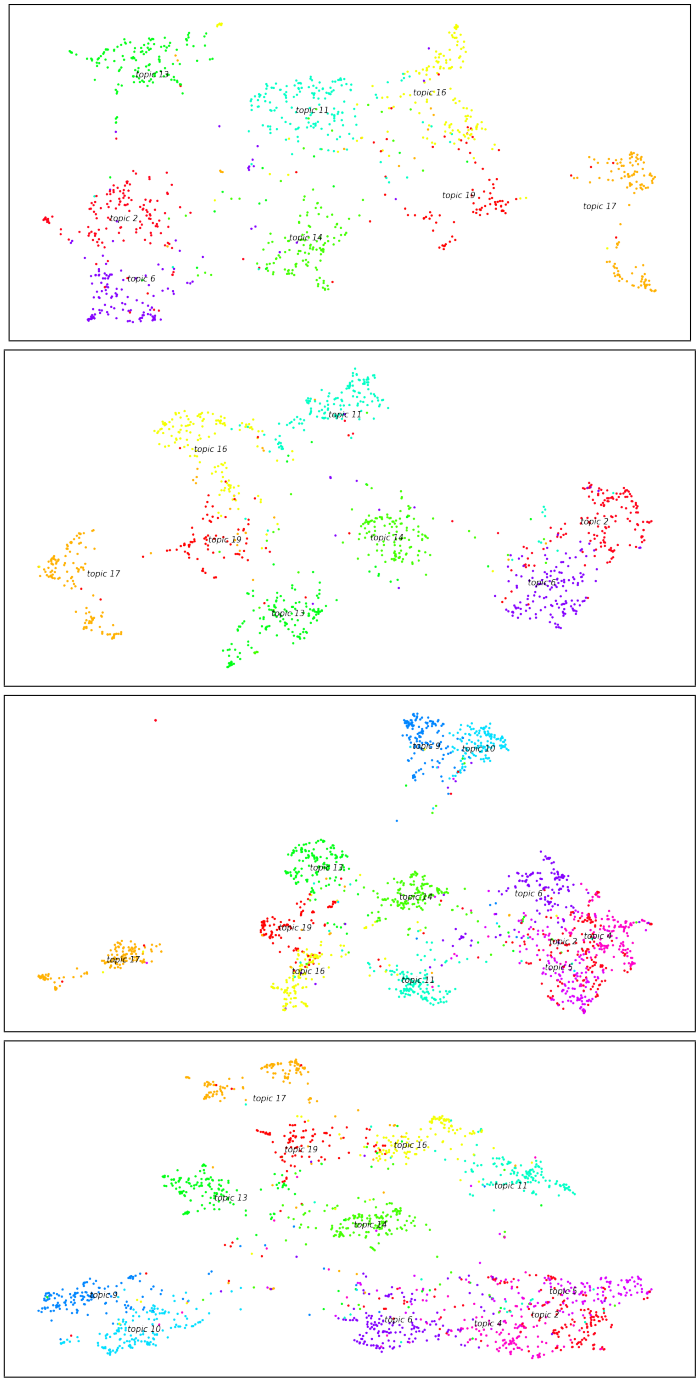

Figure 9: **Visualization of dynamic 20NG - Part 1**, our adapted version of 20NG for dynamic topic modeling. We display the representation of embedded documents along with their ground-truth topics for timestamps $t = 0$ (*upper figure*), $t = 1$ (*2nd figure*), $t = 2$ (*3rd figure*), and $t = 3$ (*lower figure*). Each color corresponds to a ground-truth topic.

As we can see, in the 20NG local representations the topics are more sparsed than compared to 20NG global representation. The fact that there are less documents and topics in each local representation

and that topics are less mixed facilitates the clustering and therefore the detection of ground-truth topics. Unlike BERTopic, our model performs clustering at each timestamp on the local representations of 20NG. This can explain why our model approximates better the ground-truth number of topics, compared to BERTopic.

B Topic evolution

B.1 Visualization of the evolution of the number of detected topics

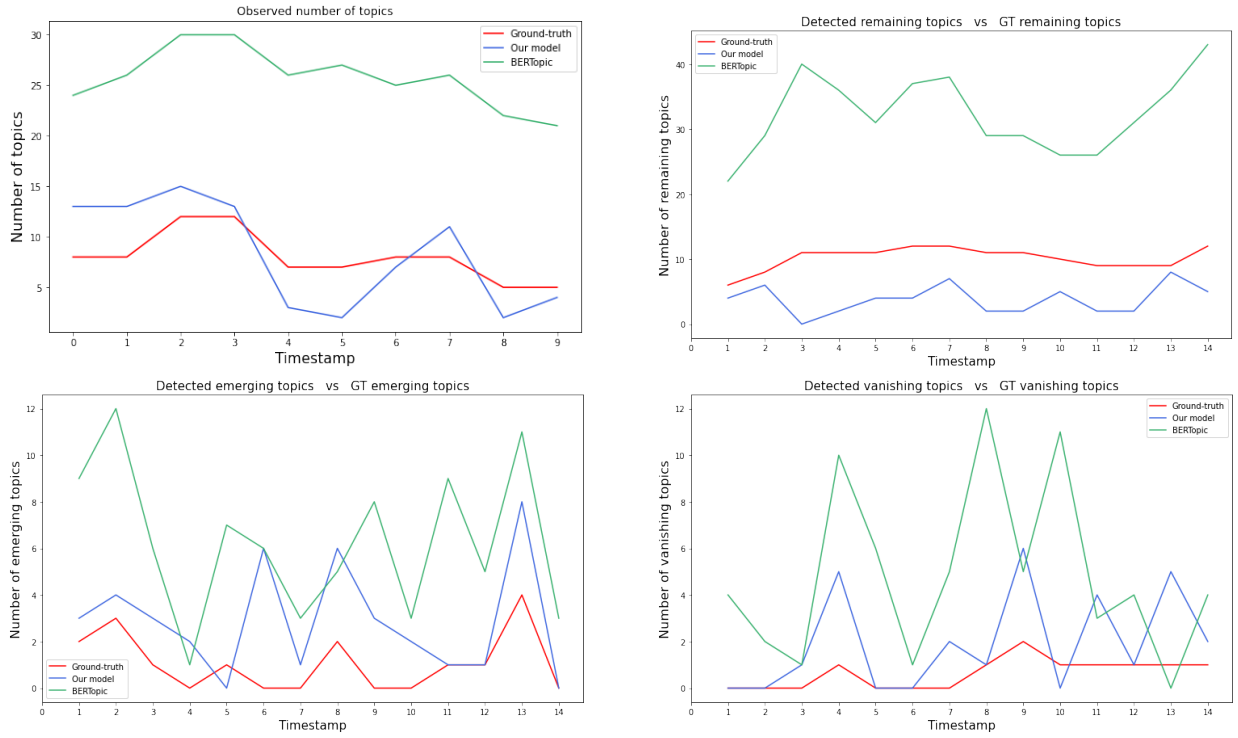

Figure 10: **Examples of the evolution of number of topics over time**. The *green* and *blue* curves correspond respectively to the results obtain with BERTopic and our model. The *red* curves corresponds to the GT values. The *upper left* figure corresponds the to the evolution of number of overall topics detected at each timestamp. The *upper right* figure displays the evolution of the number of topics detected as remaining topics. The *lower left* figure corresponds to the evolution the number of topics detected by the models as emerging topics. The *lower right figure* is the evolution of the number of topics detected as vanishing topics by the models.

Figure 10 is part of the results of the experiment explained in section 6.6.2. Visually, we can see that our model approximates better the GT number of topics over time compared to BERTopic. Figure 10 corroborates the quantitative results obtained on the evaluation of the model's capacity to detect the same number of topics over time as the ground-truth values. Overall, the number of topics detected by our model is not perfect, but the results are way better than ther number of topics detected by BERTopic.

B.2 Evolution of word probabilities

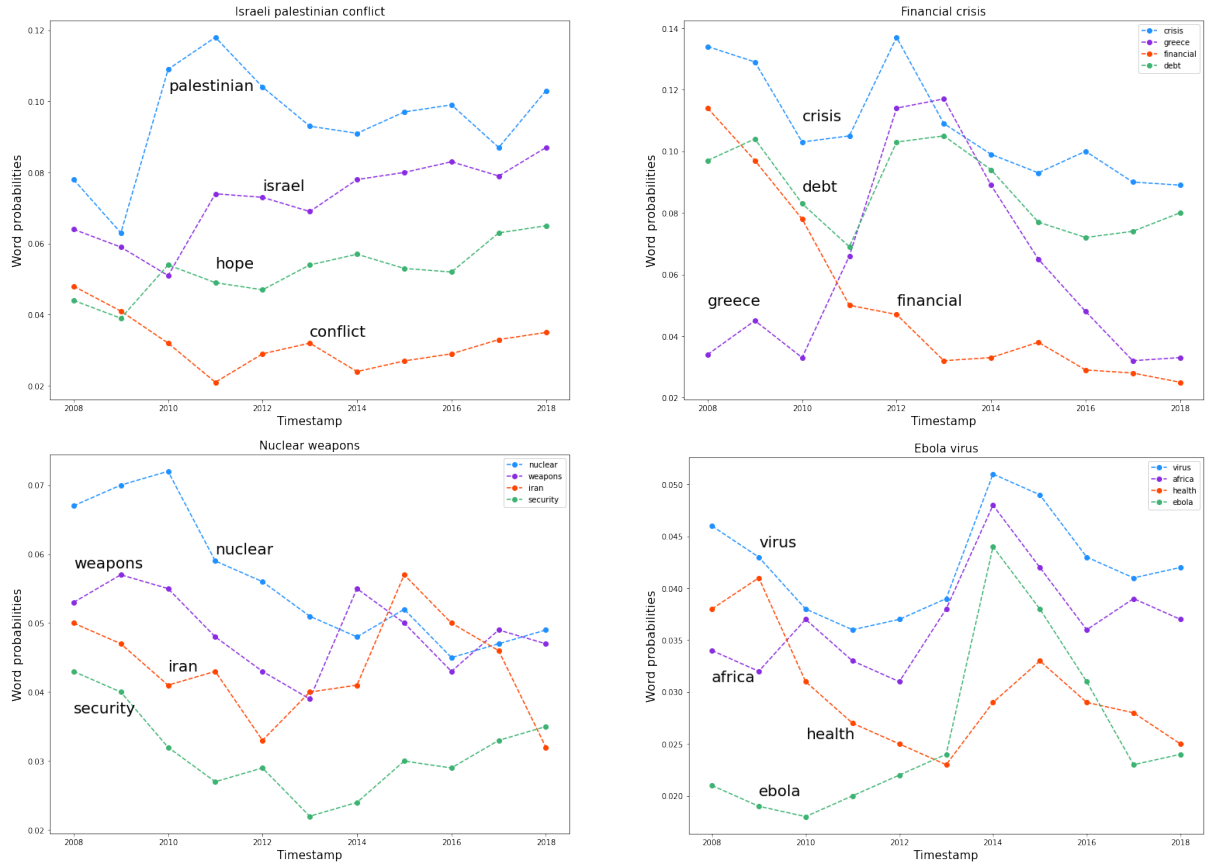

Figure 11: **Evolution of topic words' probabilities over time**. We display examples of evolution of topics detected by our model on the UN dataset. Precisely, we display the word probabilities evolution of the top 4 words of topics *Israeli palestinian conflict* (upper left figure), *Financial crisis* (upper right figure), *nuclear weapons* (lower left figure), *ebola virus* (lower right figure). These topics were retrieved from the UN dataset.

As explained in section 5.6.3, our model allows the tracking of topics over time. This tracking enables us to analyze the evolution of topics representation, in particular the evolution of topic words. Figure 11 is an example of topic evolution observed thanks to our model.

C Distribution of Chamfer distance values

In this section we display the results of the experiment explained in section 5.6.3, experiment that was conducted for choosing a Chamfer Distance threshold for the tracking of topics over time. As mentioned earlier, we had a particular interest in observing the distribution of CD values obtained for topics that have around half of their topic words in common. Since our model retrieves 22 topic words, we look at the CD distribution for number of words common close to 11. This is because the CD decreases with the number of words in common between topics. That is, the greater the number of words in common between topics, the smaller the CD value betwee these topics. In figures 12, 13 and 14, we can see that the distribution shift slowly toward the value 0 as the number of words in common increases.

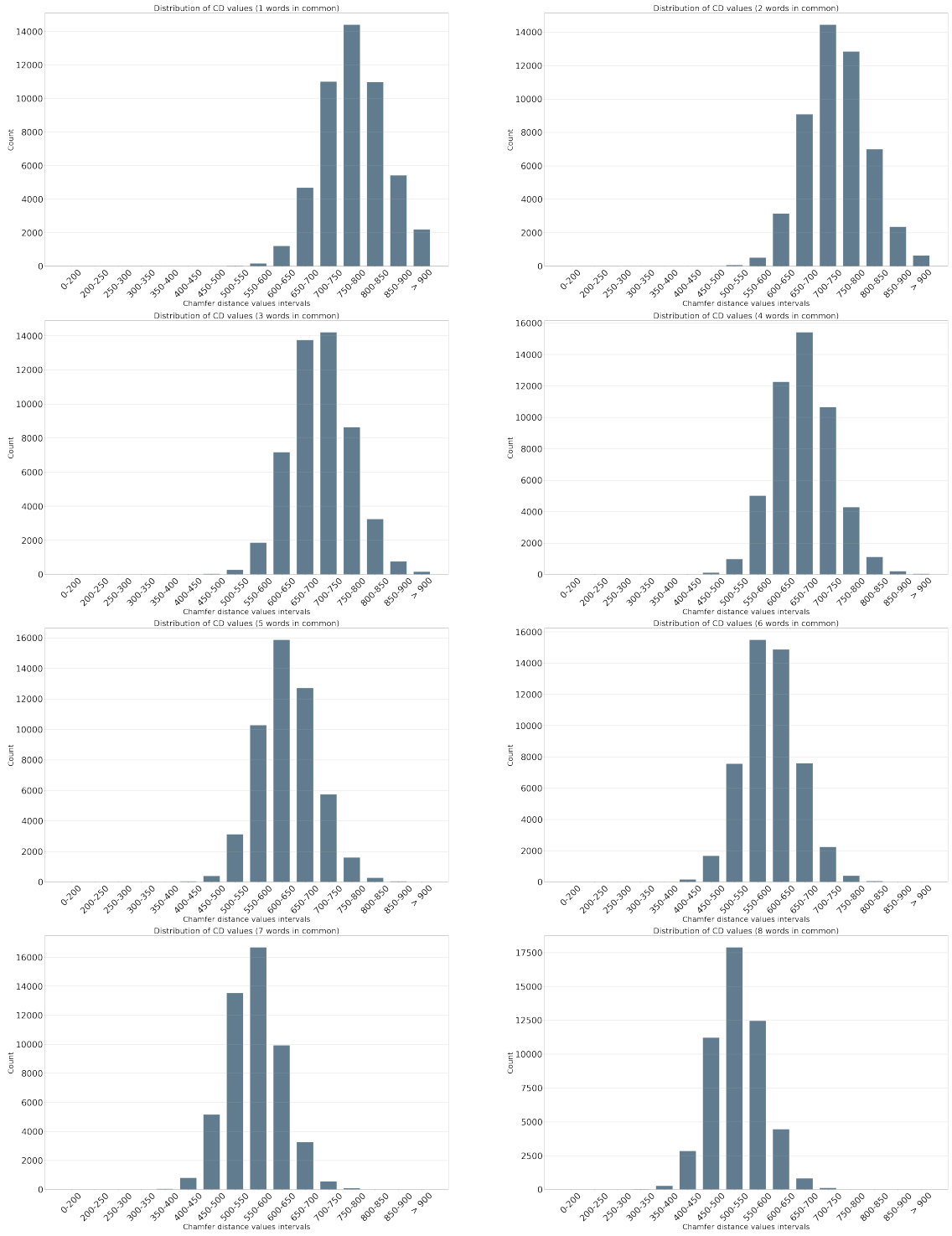

Figure 12: **Chamfer Distance distributions - First part**. We display the CD distributions obtained for each N number of words in common between topics, where N takes value between 0 and 8, in increasing order. The first distribution (*upper left figure*) displays the distributions obtained for $N = 1$. The last distribution (*lower right figure*) displays the dictributions obtained for $N = 8$.

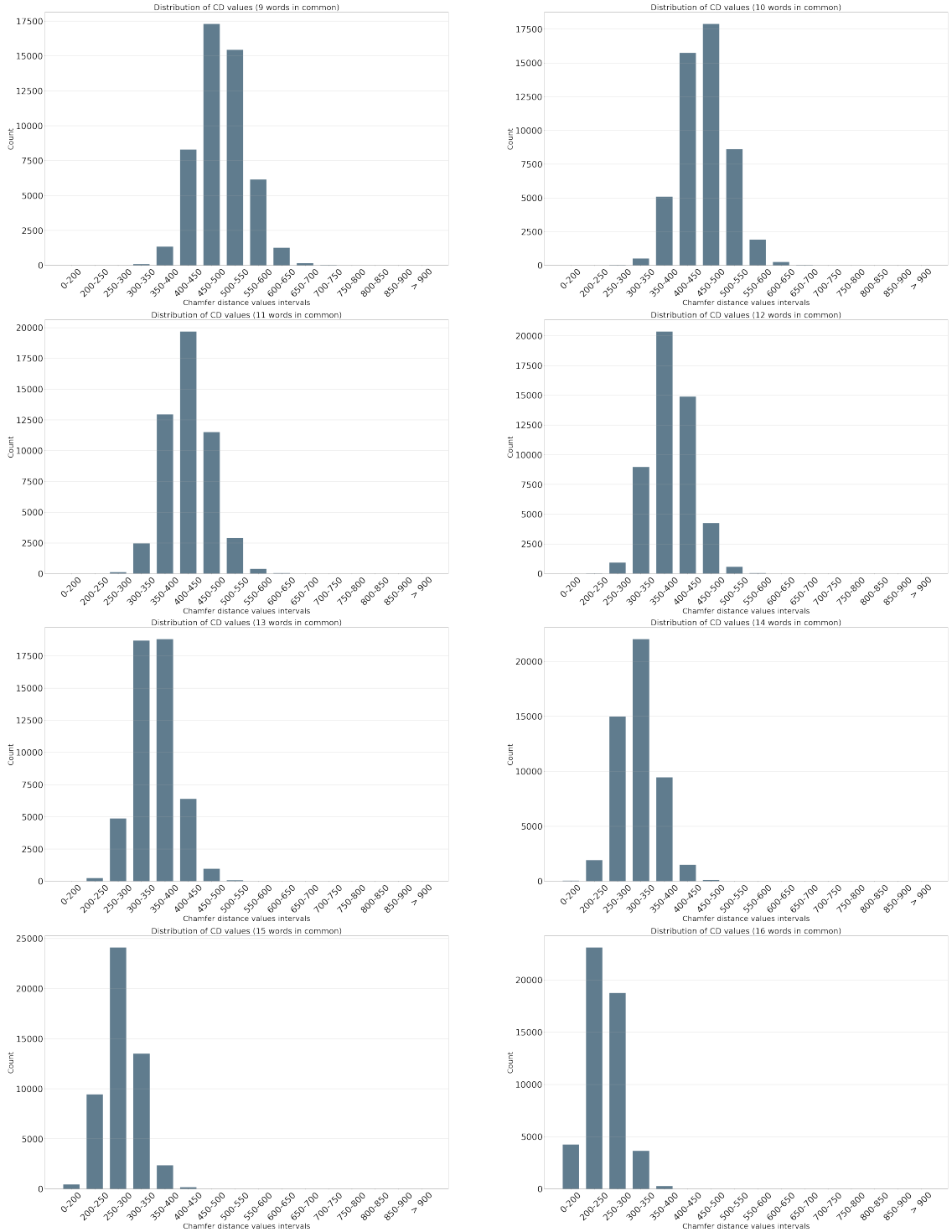

Figure 13: **Chamfer Distance distributions - Second part**. We display the CD distributions obtained for each N number of words in common between topics, where N takes value between 9 and 16, in increasing order. The first distribution (*upper left figure*) displays the distributions obtained for $N = 9$. The last distribution (*lower right figure*) displays the dictributions obtained for $N = 16$.

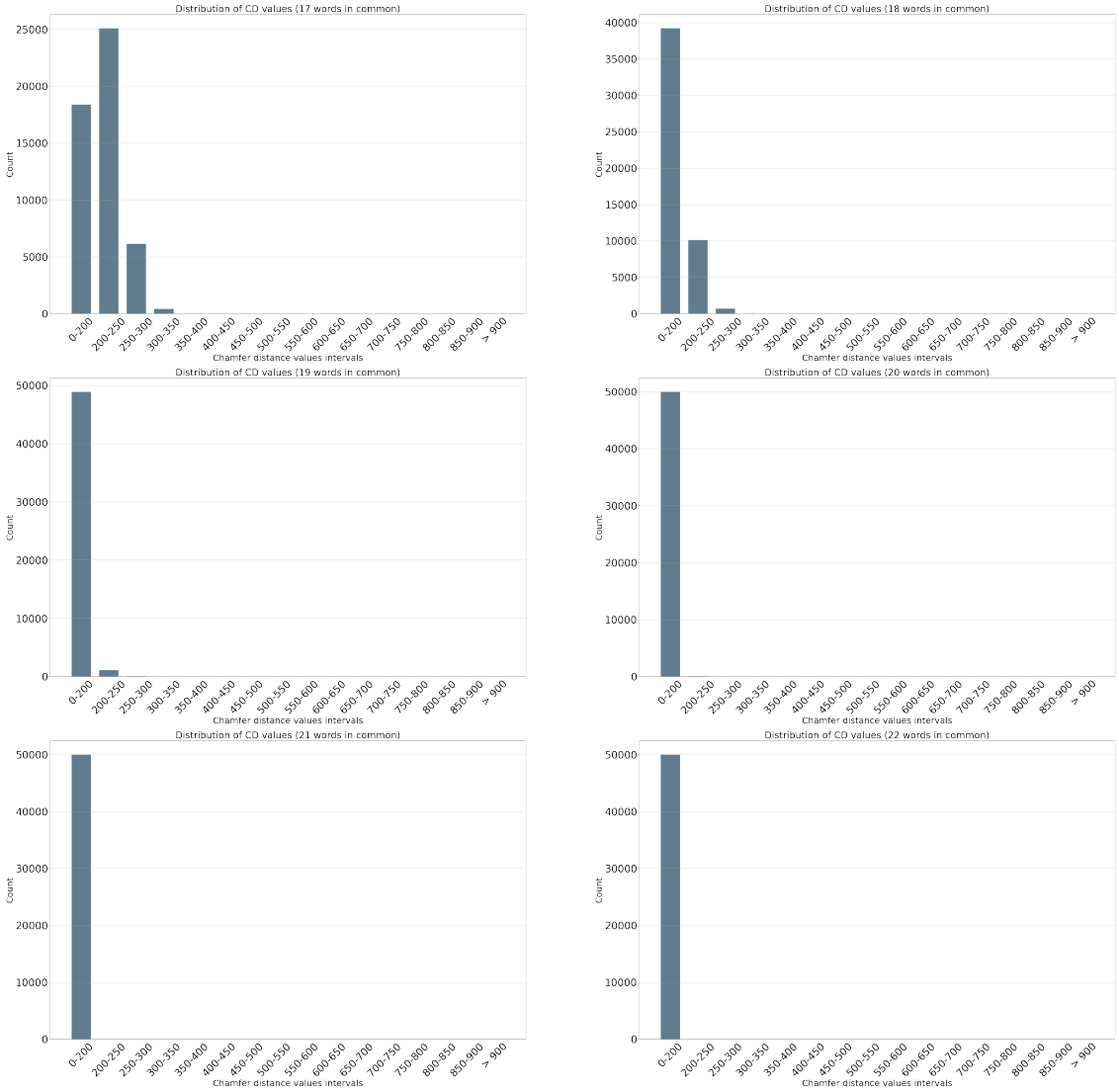

Figure 14: **Chamfer Distance distributions - Third part**. We display the CD distributions obtained for each N number of words in common between topics, where N takes value between 17 and 22, in increasing order. The first distribution (*upper left figure*) displays the distributions obtained for $N = 17$. The last distribution (*lower right figure*) displays the dictributions obtained for $N = 22$.