Merging Embedded Topics with Optimal Transport for Online Topic Modeling on Data Streams

Federica Granese[1,3] (✉), Benjamin Navet[4],
Serena Villata[1], and Charles Bouveyron[2]

[1] Université Côte d'Azur, CNRS, Inria, I3S, Marianne, France
[2] Université Côte d'Azur, Inria, CNRS, LJAD, Maasai, France
[3] Inria, Defence & Security mission, France
[4] Université Côte d'Azur, 3IA TechPool, France
{firstname.lastname}@inria.fr

Abstract. Topic modeling is a key component in unsupervised learning, employed to identify topics within a corpus of textual data. The rapid growth of social media generates an ever-growing volume of textual data daily, making online topic modeling methods essential for managing these data streams that continuously arrive over time. This paper introduces a novel approach to online topic modeling named StreamETM. This approach builds on the Embedded Topic Model (ETM) to handle data streams by merging models learned on consecutive partial document batches using unbalanced optimal transport. Additionally, an online change point detection algorithm is employed to identify shifts in topics over time, enabling the identification of significant changes in the dynamics of text streams. Numerical experiments on simulated and real-world data show StreamETM outperforming competitors. We provide the code publicly available at `https://github.com/fgranese/StreamETM`.

Keywords: Topic modelling · Optimal transport · Data streams.

1 Introduction

With the rapid expansion of social media and digital communication, vast amounts of textual data are continuously generated and distributed across various platforms. This growing volume of information necessitates automated methods for efficient information retrieval. In this context, topic models are powerful statistical tools for uncovering the hidden semantic structure within a collection of documents [10]. Specifically, these models aim to identify latent topics based on word co-occurrence patterns. Each topic represents a coherent semantic concept and is characterized by a group of related words. For instance, a topic related to `sports` may include words such as `baseball`, `basketball`, and `football` [19]. Topic models have been widely applied to analyze various types of textual data, including fiction, non-fiction, scientific publications, and political texts [5]. However, most of the existing work on topic modeling focuses on *offline* settings, where the model is trained on a fixed dataset (batch) and remains static. However, with the continuous generation of new content, there is a growing need for models that can operate in an *online* setting. Typical examples are news agencies that release their clients' news in real time or social networks that continuously deliver their users' posts to the network. In these scenarios, topic modeling algorithms must continuously update as new documents arrive.

Recent solutions for online topic modeling are often built upon BERTopic [12], a model that generates topic representations in three main steps: document embeddings, dimensionality reduction, and clustering. Unfortunately, since these models rely on pre-trained language models, fine-tuning the parameters for each step as new data arrives is challenging. This can make maintaining an efficient and adaptive process difficult as the data evolves in real-time.

In online settings, another challenge is to automatically associate new topics with existing ones. Indeed, topics are not static, but they evolve over time, often shifting in meaning and representation. Nevertheless, most existing models rely on static clustering methods or fixed word distributions, making it difficult to track these changes effectively. For example, before 2022, discussions on `AI` were likely dominated by terms like `transformers` and `GAN`, whereas today, they focus more on `LLM`.

Finally, users of these online methods must be able to detect significant shifts in the model's dynamics. Indeed, analysts monitoring data flows of this type are particularly interested in being alerted when a sudden or significant change has occurred in the data flow. In this case, the user can analyze the changes between topics and take appropriate decisions and actions. To our knowledge, this feature is not currently offered as part of online topic modeling methods.

This work addresses these limitations through the following contributions:

1. **We explore the potential of optimal transport for topic association and discovery**, demonstrating its effectiveness in aligning evolving topics and ensuring coherent topic transitions over time, and its superiority over Euclidean or Cosine similarities for this task.
2. **We introduce StreamETM, an online version of the Embedded Topic Model (ETM)**. StreamETM combines a variational inference strategy for the ETM model, applied sequentially on consecutive time windows, with a merging approach based on unbalanced optimal transport.
3. **We complete StreamETM with a change point detection algorithm**, allowing the automatic determination of significant changes in the dynamics of the studied documents. To our knowledge, StreamETM is the only unsupervised and online approach proposed for the complex tasks of online topic modeling and change point detection on text data streams.

1.1 Related works

Offline setting. Topic modeling was initially developed using heuristic approaches and was later studied with a statistical perspective two decades ago.

The Latent Semantic Index [7](LSI) is considered the first work to provide statistical foundations for this task. Building on this, Probabilistic LSI (pLSI) [14] introduced a mixture model, where each component represents a specific topic and defines a corresponding vocabulary distribution. However, LSI lacks a generative model at the document level and is prone to overfitting. In 2003, Blei et al. proposed Latent Dirichlet Allocation (LDA) [4], which models topic proportions using a Dirichlet distribution. Extensions include deep generative models, such as [18], which introduced a variational distribution parametrized by a neural network and Wasserstein autoencoders [16]. A successive evolution of the LDA is with the Embedded Topic Model [10] (ETM), which allowed using deep embeddings to represent both the words and the topics in the same vector space. Specifically, these embeddings are part of the decoder and can be pre-trained on large datasets to incorporate semantic meaning. More recently, language models such as BERT [8] have been used for topic modeling. BERTopic [12] is a topic model that generates topic representations in three steps: first, each document is embedded using a pre-trained language model; second, UMAP reduces the embeddings' dimensionality for optimized clustering with HDBSCAN; finally, topic representations are extracted from the clusters using a custom class-based TF-IDF (c-TF-IDF) variation. Finally, recently, topic modeling has been explored by prompting large language models to generate a set of topics given an input dataset [17,11].

Online setting. On the one hand, LDA was first extended to an online version in [13] with a stochastic optimization algorithm using a natural gradient step to optimize the variational Bayes lower bound as data arrives. However, this approach is not suited for streams of documents that cannot be stored. The solution in [2] addresses this by extending LDA to document batches using copulas. On the other hand, two versions of BERTopic can be used in an online setting, namely MergeBERT[5] and OnlineBERT[6]. MergeBERT is a pseudo-online variant of the original BERTopic model [12]. Topic models are merged sequentially by comparing their topic embeddings. If topics from different models are sufficiently similar (w.r.t. cosine similarity), they are considered the same, and the topic from the first model in the sequence is retained. However, if topics are dissimilar, the topic from the latter model is added to the former set. Crucially, this process does not involve an actual merging of topic representations, meaning that the words associated with a topic do not evolve over time as new models (and, therefore, new documents) are introduced. OnlineBERT preserves the embedding transformation of the documents and the final c-TF-IDF approach used in BERTopic while introducing an online variant for the dimensionality reduction step (IncrementalPCA), a MiniBatchKMeans for clustering, and an online CountVectorizer for tokenization to update out-of-vocabulary words and to prevent the sparse bag-of-words matrix from growing excessively large. Compared to MergeBERT, OnlineBERT, while being truly online, loses some of the advan-

[5] https://maartengr.github.io/BERTopic/getting_started/merge/merge.html

[6] https://maartengr.github.io/BERTopic/getting_started/online/online.html

tages of the former model. For instance, UMAP is generally better at preserving complex, non-linear relationships, which can lead to more coherent topics. Furthermore, the combination of IncrementalPCA and MiniBatchKMeans may result in the over-proliferation of subtopics over time. A single topic could be split into multiple subtopics as new data arrives, often leading to unnecessary topics that are difficult to interpret. We emphasize that the online and dynamic settings for topic modeling are fundamentally distinct. In online topic models, data is processed incrementally, with topics being updated in real time as new documents arrive. In contrast, dynamic topic modeling works *a posteriori* and captures the temporal evolution of topics by analyzing them across fixed time intervals (e.g., weeks or years).

Optimal transport and topic modeling. Optimal transport has already been used in a few situations related to topic modeling. We refer to [9], which employs optimal transport for label name-supervised topic modeling, assigning documents to predefined topics based on semantic similarities computed from pre-trained LMs/LLMs. Similarly, in [20], documents are embedded into an H-dimensional semantic space using a pre-trained transformer model, such as BERT. In this approach, topics and words are randomly projected into the same semantic space, with their embeddings jointly optimized alongside the transport maps. Our work differs in several key aspects. We leverage optimal transport to merge topic embeddings rather than to establish document-topic associations. Consequently, the transport map is applied to objects within the same semantic space. Moreover, unlike prior work, our approach does not employ the transport map during training to optimize embeddings. Instead, as discussed in Sec. 5, it can also be utilized at evaluation time to align predicted and ground-truth topics. Lastly, our framework operates in an online setting, with data coming over time.

2 Preliminaries on the Embedded Topic Model

Let us consider for the moment a corpus $\mathcal{W} = \{\mathbf{W}^{(1)}, \ldots, \mathbf{W}^{(D)}\}$ of D documents, where the vocabulary consists of V distinct words. Each $\mathbf{W}^{(d)}$ contains N_d words and is represented as $\mathbf{W}^{(d)} = (\mathbf{w}_1^{(d)}, \ldots, \mathbf{w}_{N_d}^{(d)}) \in \{0,1\}^{N_d \times V}$ where each word $\mathbf{w}_j^{(d)}$ is a one-hot vector, meaning that $w_{ji}^{(d)} = 1$, if the j-th word in document d is the i-th word in the vocabulary, 0, otherwise.

A "topic" is represented as a full distribution over the vocabulary, and a document is assumed to come from a mixture of topics, where the topics are shared across the corpus, and the mixture proportions are unique to each document. Specifically, a *topic* k is represented by a vector $\beta_k \in \Delta_V$, where Δ_V is the V-dimensional simplex. We denote the *topic matrix* as $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_K) \in \mathbb{R}^{V \times K}$. In Embedded Topic Models [10] (ETM), both words and topics are represented using embeddings. ETM first embeds the vocabulary in an L-dimensional space and represents each document in terms of K *latent topics*. We call the embeddings of the words $\boldsymbol{\rho} = (\rho_1, \ldots, \rho_V) \in \mathbb{R}^{L \times V}$ providing the representation of the words in a L-dimensional space. Similarly, a *latent topic k* is represented by a vector $\alpha_k \in \mathbb{R}^L$ and we denote the *latent topic matrix* $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_K) \in \mathbb{R}^{L \times K}$. In this context, the topic distribution over the vocabulary is assumed to be $\beta_k = \text{softmax}(\boldsymbol{\rho}^T \alpha_k)$, where the ETM assigns a high probability to word j in topic k by measuring the agreement between the word embedding and the topic embedding. We refer to the seminal paper of Dieng et al. [10] for the full description of the generative process of the d-th document under the ETM. Overall, the ETM model assumes that each document d is sampled from a mixture of topics with its proportion denoted as $\theta_d = \text{softmax}(\delta_d)$ where $\delta_d \sim \mathcal{N}(0, I)$. For each word j in the document, a topic assignment $z_j^{(d)}$ is sampled from a categorical distribution parameterized by θ_d. The word $\mathbf{w}_j^{(d)}$ is then generated via a softmax transformation of the inner product between the word and topic embeddings.

ETM employs variational inference to approximate the intractable likelihood of observing $\mathcal{W}$ given $\boldsymbol{\alpha}$ and $\boldsymbol{\rho}$, using a mean-field assumption where the variational distribution q_ϕ factorizes over documents. A variational autoencoder (VAE) models this distribution as a Gaussian with parameters learned by a neural network. The Evidence Lower Bound (ELBO)

$$\mathcal{L}(\mathcal{W}, \boldsymbol{\alpha}, \boldsymbol{\rho}; q_\phi) = \mathbb{E}_{q_\phi} \left[\log p(\mathcal{W}, \delta \mid \boldsymbol{\alpha}, \boldsymbol{\rho}\right] - \mathbb{E}_{q_\phi} \left[\log q_\phi(\delta)\right],$$

is optimized via the reparameterization trick and stochastic gradient descent.

3 The Stream Embedded Topic Model

Let us consider a stream of documents arriving as batches at discrete time steps, $\mathcal{W}_{[1:T]} = \{\mathcal{W}^{(1)}, \ldots, \mathcal{W}^{(t-1)}, \mathcal{W}^{(t)}, \mathcal{W}^{(t+1)}, \ldots, \mathcal{W}^{(T)}\}$, where each $\mathcal{W}^{(i)}$ in $\mathcal{W}_{[1:T]}$ represents a corpus of documents as defined in Sec. 2.

3.1 Learning an ETM model on the current batch $\mathcal{W}^{(t)}$

At each time step, we aim to learn a new ETM model that, based only on the corpus of documents available at the current time step and the latent topic embeddings from the previous step, can accurately link past topics to present ones while also identifying new topics. This scenario differs from a dynamic system, where it is assumed that all information is available at the final time step T. In contrast, our setting operates online, where only the data observed up to the current time step can be used for learning. The model must continuously adapt without access to future observations, as in real-time applications.

We will refer with $M_{\mathcal{W}^{(t-1)}, \boldsymbol{\alpha}^{(t-1)}, \rho} \equiv M^{(t-1)}$ to the ETM model at time step $t-1$ where $\boldsymbol{\alpha}^{(t-1)}$ is the latent topic matrix at time $t-1$, we assume the embeddings of the words $\boldsymbol{\rho}$ to be constant over time. Similarly, we will denote the topic matrix at time $t-1$ as $\boldsymbol{\beta}^{(t-1)}$. At time t, our goal is first to maximize

$$\mathcal{L}(\mathcal{W}^{(t)}, \tilde{\boldsymbol{\alpha}}^{(t)}, \boldsymbol{\rho}; q_{\phi^{(t)}}) = \mathbb{E}_{q_{\phi^{(t)}}} \left[\log p(\mathcal{W}^{(t)}, \delta^{(t)} \mid \tilde{\boldsymbol{\alpha}}^{(t)}, \boldsymbol{\rho}\right] - \mathbb{E}_{q_{\phi^{(t)}}} \left[\log q_{\phi^{(t)}}(\delta^{(t)})\right], \tag{1}$$

following the classical offline ETM models described in Sec. 2. Therefore, we seek an appropriate merging strategy[7] g to map the previously learned topic embedding space and the current one into a new representation, and we impose $\boldsymbol{\alpha}^{(t)} = g(\tilde{\boldsymbol{\alpha}}^{(t)}, \boldsymbol{\alpha}^{(t-1)})$. Finally, $M^{(t)}$ is obtained by optimizing a second time Eq. (1) using stochastic gradient descent, with $\boldsymbol{\alpha}^{(t)}$ and $\boldsymbol{\rho}$ kept fixed.

3.2 Optimal transport for merging and discovering topics

We now analyze the problem of determining an *effective* strategy g for identifying the topics in $\boldsymbol{\alpha}^{(t-1)}$ that are most similar to those in $\tilde{\boldsymbol{\alpha}}^{(t)}$, enabling us to merge these topic embeddings while incorporating the new topics present in $\tilde{\boldsymbol{\alpha}}^{(t)}$.

Transport map computation. We recall that

$$\boldsymbol{\alpha}^{(t-1)} = (\alpha_1^{(t-1)}, \dots, \alpha_K^{(t-1)}) \in \mathcal{A}^{(t-1)} \subseteq \mathbb{R}^{L\times K}$$

and

$$\tilde{\boldsymbol{\alpha}}^{(t)} = (\tilde{\alpha}_1^{(t)}, \dots, \tilde{\alpha}_J^{(t)}) \in \mathcal{A}^{(t)} \subseteq \mathbb{R}^{L\times J},$$

where J can be either different from or equal to K. Let $a = \frac{1}{K}\sum_{i=1}^{K} \delta_{\alpha_i^{(t-1)}}$ and $\tilde{a} = \frac{1}{J}\sum_{i=1}^{J} \delta_{\tilde{\alpha}_i^{(t)}}$ be the two discrete distributions of mass on $\mathcal{A}^{(t-1)}$ and $\mathcal{A}^{(t)}$. We aim to find the least costly way to shift the mass (i.e., the topics) from the previous time step to the current one. To this end, we formulate the problem as Unbalanced Optimal Transport (UOT) [3], a relaxed version of OT where the total mass of each source (the topics of $\tilde{\boldsymbol{\alpha}}^{(t)}$) can be spread across multiple targets (the topics of $\boldsymbol{\alpha}^{(t-1)}$):

$$\mathbf{T}^\star = \underset{\mathbf{T}\in\mathbb{R}_+^{J\times K}}{\arg\min}\, \langle \mathbf{C}, \mathbf{T}\rangle + \lambda_{\tilde{a}} D_\psi\left(\mathbf{T}\mathbb{1}_J, \tilde{a}\right) + \lambda_a D_\psi\left(\mathbf{T}^\top \mathbb{1}_K, a\right), \tag{2}$$

where $\langle\cdot,\cdot\rangle$ is the Frobenius inner product, and $D_\psi(\cdot,\cdot)$ is the Bregman divergence that penalizes violations of the marginal constraints. Additionally, $\lambda_a \in \mathbb{R}_+$ (resp. $\lambda_{\tilde{a}} \in \mathbb{R}_+$) represents the penalty associated with a (resp. $\tilde{a}$). Moreover, $\mathbf{C} \in \mathbb{R}_+^{J\times K}$ is the cost-matrix in which the entries C_{jk} encode the cost of moving $\tilde{\alpha}_j^{(t)}$ towards $\alpha_k^{(t-1)}$. In this particular setting, as we deal with text, we chose the cosine similarity as the cost function. Finally, $\mathbb{1}_{(\cdot)}$ represents the vector of dimension $(\cdot)\times 1$, which is used to ensure that the sum per row and column does not diverge significantly from the original distributions a and $\tilde{a}$. In this way, only a portion of the total mass is transported, and the total mass can be unbalanced between the sources and targets due to the constraint relaxation. Intuitively, a sparse transport matrix indicates that mass is transferred only between semantically similar topics, while distant topics receive no transport.

Note that the UOT problem can be efficiently recast as a non-negative penalized linear regression problem. We refer to [6] for additional details.

[7] Note that, at $t = 1$, no merging strategy is applied.

Merging topics and discovery of new ones. For each $\tilde{\alpha}_j^{(t)}$, we determine the corresponding target topic by identifying the index where the transport plan assigns the highest mass. Specifically, we select the topic $k^\star$ that maximizes the transport matrix entry, given by: $k^\star = \arg\max_{k \in \{1,\dots,K\}} T^\star_{jk}$. Therefore,

$$\alpha_j^{(t)} = \omega \tilde{\alpha}_j^{(t)} + (1-\omega)\alpha_{k^\star}^{(t-1)}, \tag{3}$$

where $\omega \in [0,1]$ is a memory parameter. Otherwise, if no mass has been transported from j, meaning for all $k \in \{1,\dots,K\}$, $T^\star_{jk} = 0$, the jth topic is a new one and can be added to the set of topics: $\alpha_{J+1}^{(t)} = \tilde{\alpha}_j^{(t)}$.

3.3 Change point detection

To monitor the significant changes in the dynamic of the data stream we analyze, we propose to add a change point detection step to our approach. In addition to the detection of new topics (topics that are added in the merged model), we propose to make use of the online Bayesian changepoint detection (OCPD [1]) algorithm to monitor significant changes in the sequence of merged models $\{M^{(1)}, M^{(2)}, \dots, M^{(T)}\}$. We propose to apply the OCPD algorithm to time series of topic distributions over the documents at different time steps. It is worth highlighting that OCPD is, to this date, the most performant change point detection method able to work in a fully online framework and can issue alerts on the fly.

4 Experimental Setting

We describe the experimental setting for evaluating StreamETM. Designing this setting posed several challenges. First, since we consider an online and unsupervised setting, our evaluation goes beyond assessing model performance at individual time steps; we also analyze the overall interaction dynamics induced by merging topic embeddings. Second, as we are working with topics, relying solely on human judgment for evaluation is insufficient, requiring us to explore alternative quantitative metrics. Note that we compare with online (not dynamic) topic models, as dynamic approaches lack incremental learning, which is central to our study (cf. Sec. 1.1).

4.1 Datasets

We consider the `20kNewsGroup`[8] dataset as text corpora for our experiments, comprising around 18k newsgroup posts on 20 topics. We confine to 5 of the 20 topics and randomly draw 15 times approximately 5k samples from the total datasets[9]. We partition the 5k samples into 500 sample batches to simulate

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

[9] The same sample may appear in multiple datasets. However, each dataset would have been too small without repetition when partitioned across different time steps.

a ≈10 time steps scenario. For each time step, the corresponding dataset has been pre-processed by first lemmatizing the text, removing lowercase and punctuation, filtering out stopwords (cf. `nltk.corpus.stopwords`), and eliminating low-frequency words (words appearing only once) and those appearing in more than 70% of documents to reduce overly frequent terms. The topic distributions are computed considering practical use cases.

Our practical use-cases. We simulate the online setting by assuming that each time step $\boldsymbol{\tau}^{(i)}$, $i = 1, \ldots, T$, is represented as a distribution:

a) Custom: A designed setting where the topics are intentionally chosen to be sufficiently distinct. At each time step, at most four out of five topics are *active* (Fig. 3(a)). Topics: `autos`, `sport`, `medicine`, `space`, `religion`.

b) Dynamic: Text corpora with significant temporal shifts in topic relevance. At each time step i, the activity of each topic k is determined independently. A binary variable $z_k^{(i)}$ is drawn from a Bernoulli distribution $z_k^{(i)} \sim \text{Bernoulli}(p)$, where $p \in [0, 1]$ represents the probability that a topic remains *active*. For each time step, the unnormalized proportion of topics is: $\tau_k^{(i)} = z_k^{(i)} \cdot \text{Dir}_k(\alpha)$, $\alpha > 1$. Finally, the topic proportions $\boldsymbol{\tau}^{(i)}$ are normalized to ensure they sum to 1 (Fig. 6(a)). Topics are randomly chosen: `computer`, `sale`, `cryptography`, `religion`, `mideast`.

4.2 Architectures and training procedure

StreamETM. At each time step, we trained an ETM on English text using fixed GloVe embeddings from the `glove-wiki-gigaword-300` vocabulary, truncated to the first 15k words. The model was initialized with 3 topics: an 800-dimensional hidden layer for the encoder and 300-dimensional word embeddings. The topic embeddings were initialized using Xavier uniform initialization at time step 0, while in subsequent iterations, they were set to the values computed at the previous time step, following the strategy described in Sec. 3.2. The training was performed over 3k epochs with a batch size of 1000, a learning rate of 0.01, and a weight decay of 0.006 using the Adam optimizer. Regarding the UOT procedure, we use the Cosine Distance for the cost map. The transport map is computed using the Python function `ot.unbalanced.mm_unbalanced`, with KL divergence and marginal relaxation at 0.09.

MergeBERT. We used the `paraphrase-multilingual-mpnet-base-v2` model from SentenceTransformers to generate document embeddings. These embeddings were processed using UMAP for dimensionality reduction, with 10 components, a minimum distance of 0.1, and cosine similarity. HDBSCAN was applied for clustering with Euclidean Distance and a minimum cluster size of 3. We improved term weighting using the ClassTfidfTransformer with BM25. BERTopic was used for topic modeling, with the PartOfSpeech model for enhanced text representation. The cosine similarity threshold for merging topics was set to 0.7. A lower threshold would lead to over-proliferation of topics, while a higher value could cause the newly formed topics to collapse.

4.3 Evaluation metrics

Qualitative. We analyze the distribution of topics over time and visually compare the original distribution with the predicted ones. In addition, we examine the top five words associated with each topic at each time step. Since each setting involves 15 training runs, we manually align topic indices across executions. On average, MergeBERT identifies more than 20 topics at each time step. Therefore, we focus on topics that are more similar to the targeted ones for this metric.

Quantitative. We measure topic quality in terms of topic coherence (TC) [15] and topic diversity (TD) [10]. Topic coherence is the average pointwise mutual information of two words drawn randomly from the same document: the most likely words in a coherent topic should have high mutual information. In contrast, topic diversity is the percentage of unique words in the top 10 words of all topics. Intuitively, topic diversity measures how varied the overall topics are.

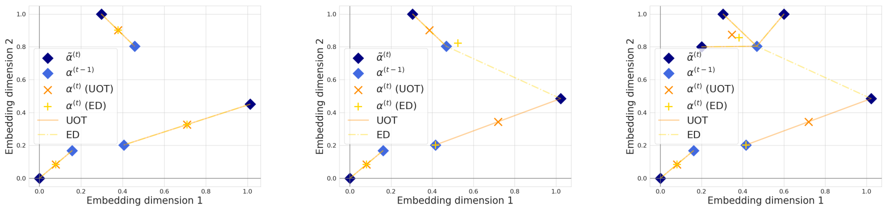
Online change point detection. We apply an online change point detection (OCPD) algorithm to the predicted topic distributions, using the R package `ocp` to analyze topic proportions over time and detect significant rupture points. If the predicted distributions closely resemble the original ones, the algorithm should identify rupture points at approximately the same time steps. To avoid the need for manual topic alignment across different training runs, we consider a rupture point to be correctly identified (true positive) if the algorithm detects *any* change at the same time step as in the original distribution or within one-time step before or after, regardless of the specific topic. We compute ROC curves based on different threshold values of the OCPD (between 0 and 1).

5 Numerical Experiments

This section examines StreamETM from multiple perspectives. We first highlight the advantages of optimal transport for topic merging and discovery, followed by a quantitative evaluation. Finally, we analyze the approach from qualitative, quantitative, and online change-point detection perspectives.

5.1 Impact of optimal transport for topic merging and discovery

Comparison with Euclidean Distance. In Fig. 1, we illustrate the role of unbalanced optimal transport (UOT) in topic merging, comparing it to the classical Euclidean Distance (ED) within a 2D Euclidean space. While ED evaluates pairwise topic distance, disregarding the overall distribution, UOT accounts for

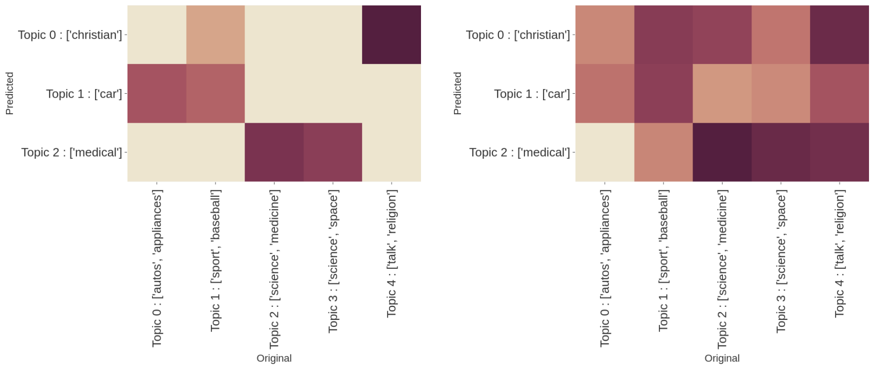


(a) No perturbation applied (b) Perturbation on a topic (c) New topics

Fig. 1. Topic embeddings in a Euclidean space. On the left, the setting is without perturbation, and on the center and the right, a perturbation is added to the dark blue diamond at the position $(1.01, 0.45)$. Dark blue diamonds represent topic embeddings at time $t-1$, while light blue markers indicate topic embeddings at time t before merging. The merged embeddings obtained via UOT are shown as '$\times$', whereas those obtained using ED are shown as '$+$'. Dashed lines connect topics matched by UOT, while dot-dashed lines indicate associations based on ED.

the global structure. As a result, *(i)* ED may introduce spurious correlations; *(ii)* The merge based on ED could be more sensitive to small perturbations of the input. For simplicity, we model the topics at time $t-1$ (in light blue) as samples from a normal distribution. Similarly, the topics at time t (in dark blue) are generated by perturbing each topic at time t with an additional value drawn from the same normal distribution. To compute the transport map, we use the procedure previously described but consider the Euclidean cost matrix. Initially, ED and UOT are equally mapping the topics at time t to the ones $t-1$, resulting in generating the same new topics, represented as a '$+$' for ED and a '$\times$' for UOT (cf. Fig. 1(a)). However, when introducing a small perturbation to the input—specifically shifting the dark blue point initially at coordinates $(1.01, 0.45)$ to $(1.02, 0.49)$ in Fig. 1(b)—we observe that while UOT remains stable, ED yields unintended new associations, leading to spurious topics. Finally, as new topics emerge, as shown in Fig. 1(c), we observe that the newly merged topic is moving closer to the one created with UOT. However, the topic at $(1.02, 0.49)$ is almost completely lost in ED. We can imagine that while these behaviors can be easily crafted in a lower-dimensional space, more complex reactions could arise in a higher-dimensional space, especially when considering the issue of topic overproliferation. This is particularly relevant since most metrics do not account for the global structure of the distributions.

Comparison with Cosine Distance. Similar to our approach with Euclidean Distance, we now demonstrate the role of UOT compared to Cosine Distance (CD). We first evaluate UOT and CD to merge the topic embeddings and check the discovery performance in subsequent steps. Specifically, we consider 7 of the 20 discussion topics in the `20kNewsGroup`; we randomly draw 1k documents from these topics for two subsequent time steps whose topic distributions are fixed and obtained from a Dirichlet distribution of parameter 1. We expect our approach to merge 3 common topics and to detect that 2 new ones should not be merged. Tab. 1 reports the topic merging and discovery accuracies (the closer to 1, the better) averaged on 50 simulated document sets. As can be seen, the UOT approach is globally more efficient than the other approaches.



(a) Optimal Transport

(b) Cosine Distance

Fig. 2. The left figure shows the transport map, while the right one depicts the cosine similarity map. In both cases, darker cells indicate regions of higher transported mass, Fig. 2(a), or shorter cosine distance, Fig. 2(b).

Table 1. Topic merging and discovery accuracies on 50 simulated datasets. MA stands for Merging Accuracy, and DA for Discovery Accuracy.

Method	MA	DA	$H_{(\mathrm{MA,DA})}$
UOT Cosine	0.79 ± 0.24	**0.93** ± 0.18	**0.85**
UOT Euclidean	0.75 ± 0.23	0.85 ± 0.28	0.79
UOT Minkowsky	0.75 ± 0.23	0.85 ± 0.28	0.79
CD	**0.84** ± 0.21	0.72 ± 0.26	0.77
ED	0.74 ± 0.24	0.84 ± 0.25	0.76

Finally, we evaluate UOT and CD to align the predicted topics with the true topics provided by the dataset labels of `20kNewsGroup`. Specifically, we focus on the Custom setting shown in Fig. 3. Since ETMs treat documents as bag-of-words, we construct topic embeddings for what we refer to as 'pure documents'—documents whose topics are directly derived from the dataset labels. Typically, these labels contain only two or three words, so we augment them by adding the most semantically similar words based on the GloVe model. Therefore, given a topic k, we can compute β_k by considering the words in our 'pure documents', and we can approximate the corresponding topic embeddings as

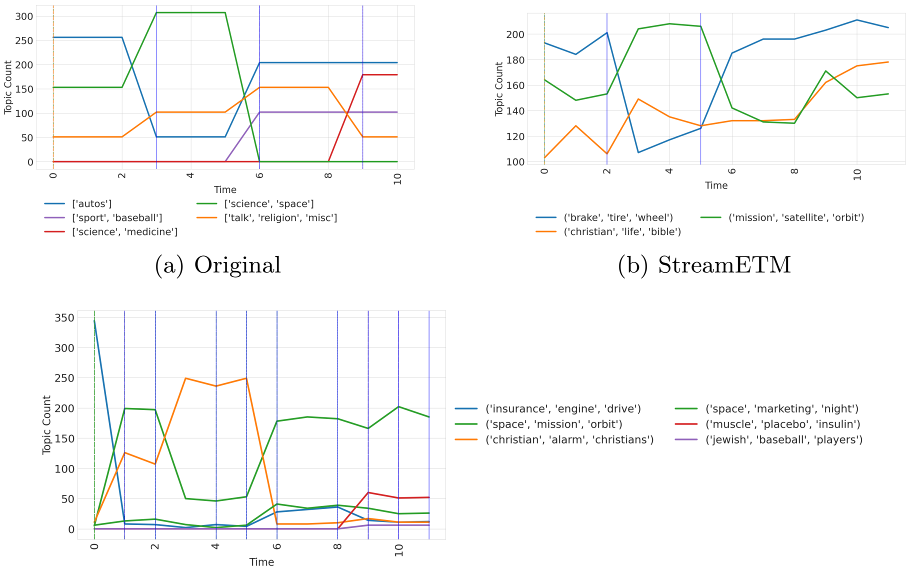
$$\alpha_k \approx (\boldsymbol{\rho}^\top)^+ \ln(\beta_k), \tag{4}$$

where $(\boldsymbol{\rho}^\top)^+$ is the Moore-Penrose pseudoinverse of $\boldsymbol{\rho}^\top$. The logarithm accounts for the inversion of the softmax transformation. Let us denote as $\widehat{\boldsymbol{\alpha}}$ the latent topic matrix extracted from the model at time T and $\boldsymbol{\alpha}_{pure}$ the matrix computed from the 'pure documents'. Our goal is to visualize how optimal transport and cosine similarity align $\widehat{\boldsymbol{\alpha}}$ with $\boldsymbol{\alpha}_{pure}$. To compute the transport map, we follow the same strategy used during training (Sec. 3).

The results are shown in Fig. 2, where darker cells in the matrices indicate regions of higher transported mass (Fig. 2(a)) or shorter Cosine Distance (Fig. 2(b)). As observed, the associations in the transport map appear reasonable, as the matrix is not particularly dense, and transportation occurs primarily between similar topics. An interesting case is the predicted topic 2, which pertains to medicine and is mapped between the two original topics containing the word `science`. Conversely, the Cosine Distance matrix appears denser, leading to less immediate associations. For example, `christianity` could be associated with any original topic, even though the distance is slightly smaller from the topic `talk`, `religion`. This can lead to greater instability, as the minimum distance may not necessarily correspond to the correct topic from a human perspective.

5.2 Qualitative analysis of the recovered topic dynamics

In Figs. 3 and 6, we plot the topic evolution over time for a randomly selected training run in the Custom and Dynamic settings, respectively. Even if the proposed StreamETM model cannot identify all five topics, it can mimic the original topics' evolutions. In Fig. 3(b), after time 6, the model likely merges the topics `science`, `medicine`, `sport`, and `baseball` with other topics. Specifically, the `space` topic, instead of disappearing, likely absorbs the `medicine` topic, as both could contain similar terms. In Fig. 3(c), we observe a similar shape as in Fig. 3(a), but the topics are swapped (cf. `space` and `religion`, in MergeBERT, vs. `autos`, `space`, in the original distribution). As can be observed from the plots, compared to StreamETM, MergeBERT generates an excessive number of topics, leading to an overproliferation of topics. For example, in Fig. 3(c) the topic `insurance`, `engine`, `drive` could be regarded as the same topic as `radar`, `tire`, `detector`. More evidently, in Fig. 6(c), `church`, `faith`, `christian` and `atheist`, `religion`, `atheism`. MergeBERT appears to capture more subtopics, while StreamETM focuses on larger concepts. Several factors contribute to this difference: (i) MergeBERT is based on a SentenceTransformer model, which captures more granular, context-dependent information, whereas StreamETM treats documents as a bag of words; (ii) MergeBERT's performance is highly dependent on a large number of parameters, making it challenging and impractical to tune for each document; (iii) MergeBERT does not explicitly merge topics, instead selecting top words from the first model if cosine similarity is sufficiently high;

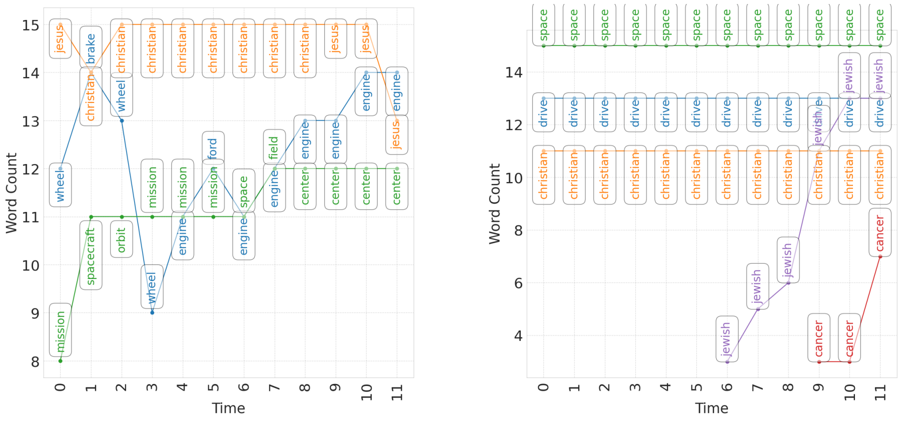


(a) Original

(b) StreamETM

(c) MergeBERT

Fig. 3. Qualitative assessment of topic evolution over time in the Custom setting. Blue vertical lines indicate the change points detected by the algorithm.

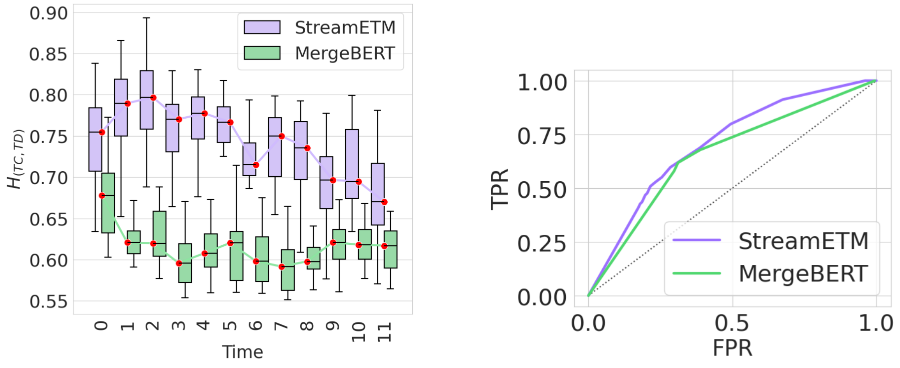


(a) StreamETM

(b) MergeBERT

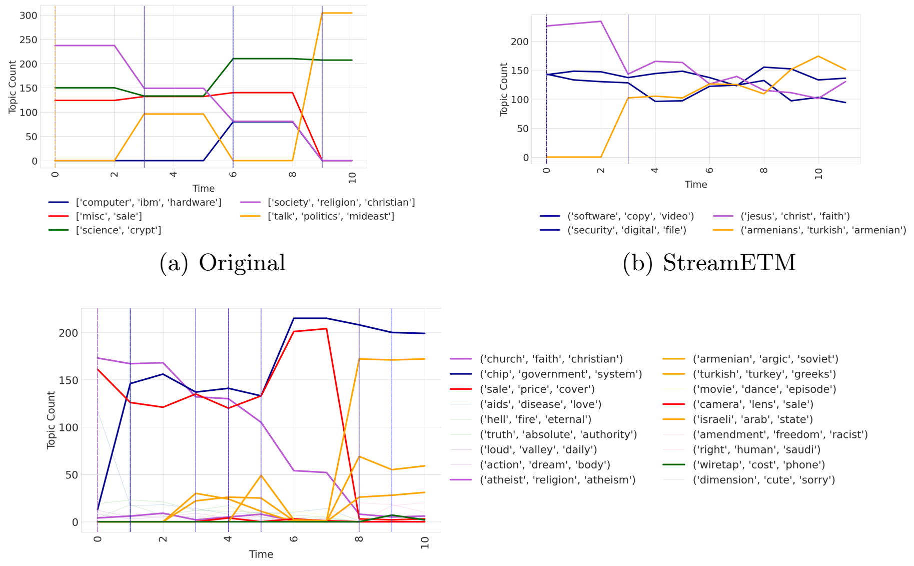
Fig. 4. Custom setting. The most frequent word for topic across the 15 training runs.

(iv) in StreamETM, the UOT mechanism enables topic embeddings to merge more effectively, an adjustable threshold on the transported mass would allow for fewer merges and the creation of more distinct topics.



(a) $\mathrm{H}_{(TC,TD)}$ (b) OCPD

Fig. 5. Custom setting. In (a), the harmonic mean between TC and TD, and in (b), the ROC curves. Results were computed across the 15 training runs.

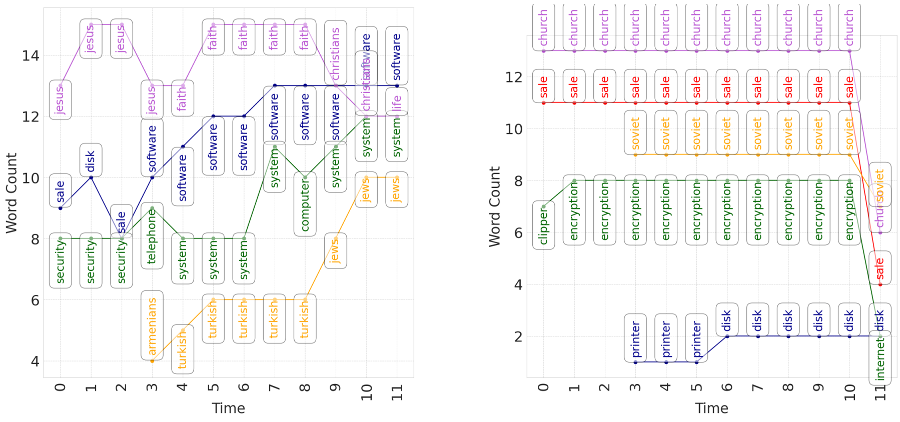


(a) Original (b) StreamETM

(c) MergeBert

Fig. 6. Qualitative assessment ot topic evolution over time in the Dynamic setting. Blue vertical lines indicate the change points detected by the algorithm.

In Figs. 4(a) and 4(b), in Figs. 7(a) and 7(b), we display the most frequent word for each topic across the 15 training runs and manually align the topic indices across the different executions. Since MergeBERT identifies more than 20 topics, we focus on the 5 topics of particular interest to us. In the Custom setting, we observe that for both models, the most frequent word is `christian`,



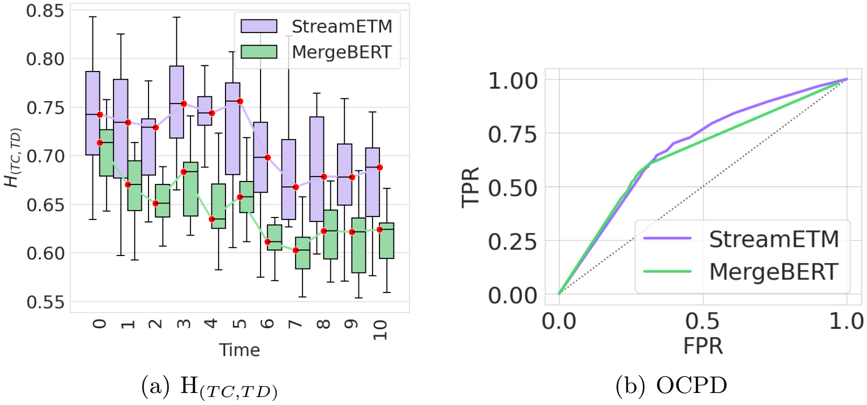
(a) StreamETM (b) MergeBERT

Fig. 7. Dynamic setting. The most frequent word for topic across the 15 training runs.

and both models show high stability around this topic, with that word appearing in almost all 15 training runs. However, we see much more variability in the other two topics in Fig. 4(a), reflecting the evolution of topics over time. For instance, the frequency of the word `wheel` associated with the `autos` topic diminishes at time 3. A similar trend is observed in the Dynamic setting. As mentioned earlier, one of the strengths of StreamETM is its ability to adapt topics over time as the text corpus evolves. The 5 top words for each topic over time across the training executions are provided in Tabs. 4 and 5 (cf. Appendix A.3).

5.3 Quantitative analysis of the recovered topic dynamics

Figs. 5(a) and 8(a) illustrate the harmonic mean between TC and TD, denoted as $H_{(TC,TD)}$, across the 15 training executions. The numerical results are provided in Tabs. 2 and 3 (cf. Appendix A.2). From the TD perspective, both models achieve satisfactory results. However, more significant differences are observed from the TC perspective. Notably, when comparing the box plot in Fig. 5(a) with the topic evolution in Fig. 3, we observe a decrease in the metric at time 6, which corresponds to the moment when there is an inversion in the distribution between the blue and green topics, alongside the emergence of the new violet topic. Following this, the metric stabilizes until time 9, when the new red topic is introduced. A similar pattern is evident in Fig. 8(a). This behavior is expected, as the model does not identify the new topic at the time step, leading to decreased topic coherence. Finally, MergeBERT shows the same behavior as StreamETM in Fig. 8(a), but at a lower TC, instead in Fig. 5(a), the behavior is slightly inverted, and this could be given to the fact that from time 0 to 6 the model associates the original blue topic evolution to the green one and the green with the orange one.



(a) $H_{(TC,TD)}$ (b) OCPD

Fig. 8. DYNAMIC setting. In (a) harmonic mean between TC and TD and in (b) ROC curves. Results were computed across the 15 training runs.

5.4 Online change point detection

Figs. 5(b) and 8(b) present the performance of the OCPD algorithm in terms of ROC curves across the 15 training runs. If the topic of evolution is correctly predicted, the algorithm should detect rupture points at approximately the same time steps. The results show that this final task is extremely difficult, and both methods exhibit certain drawbacks. With StreamETM, fewer topics are detected, which increases the likelihood that the OCPD algorithm will identify fewer rupture points than expected. In contrast, MergeBERT experiences an explosion in the number of topics, likely leading to more false positives.

6 Conclusions

We considered the extremely challenging problem of online topic modeling on document streams, with online change point detection. In order to address the limitations of existing online topic modeling approaches, we introduced StreamETM, an online extension of the Embedded Topic Model (ETM) for streams of text documents. Our method leverages variational inference to update topic distributions sequentially while incorporating optimal transport to monitor, merge, and discover evolving topics over time. This approach ensures that topics remain coherent despite the continuous influx of new documents. Beyond model development, we complemented StreamETM with a change point detection algorithm to automatically identify shifts in topic dynamics. This enables the proposed approach to provide a synthetic summary of the document stream through intelligible topics and to issue alerts to the analysts when significant changes in the dynamics of text streams are detected. Our experiments demonstrate that StreamETM effectively adapts to streaming textual data. Optimal transport provides a principled way to associate topics across time windows, addressing the shortcomings of static clustering methods. Extensive numerical experiments in simulated and real-world scenarios showed that StreamETM outperforms its competitors in various scenarios. Future work includes modeling the extension of the vocabulary that may evolve over time, and proposing a model selection strategy to determine on the fly the appropriate number of topics.

7 Acknowledgment

This work has been supported by the French government through the 3IA Côte d'Azur Investments, which are managed by the National Research Agency (ANR) with the reference number ANR-23-IACL-0001.

References

1. Adams, R.P., MacKay, D.J.: Bayesian online changepoint detection. arXiv preprint arXiv:0710.3742 (2007)
2. Amoualian, H., Clausel, M., Gaussier, E., Amini, M.R.: Streaming-lda: A copula-based approach to modeling topic dependencies in document streams. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining. pp. 695–704 (2016)
3. Benamou, J.D.: Numerical resolution of an "unbalanced" mass transport problem. ESAIM: Mathematical Modelling and Numerical Analysis **37**(5), 851–868 (2003)
4. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. Journal of machine Learning research **3**(Jan), 993–1022 (2003)
5. Boyd-Graber, J., Hu, Y., Mimno, D., et al.: Applications of topic models. Foundations and Trends® in Information Retrieval **11**(2-3), 143–296 (2017)
6. Chapel, L., Flamary, R., Wu, H., Févotte, C., Gasso, G.: Unbalanced optimal transport through non-negative penalized linear regression. Advances in Neural Information Processing Systems **34**, 23270–23282 (2021)
7. Deerwester, S., Dumais, S.T., Furnas, G.W., Landauer, T.K., Harshman, R.: Indexing by latent semantic analysis. Journal of the American society for information science **41**(6), 391–407 (1990)
8. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers). pp. 4171–4186 (2019)
9. Dhanania, G., Mysore, S., Pham, C.M., Iyyer, M., Zamani, H., McCallum, A.: Interactive topic models with optimal transport. arXiv preprint arXiv:2406.19928 (2024)
10. Dieng, A.B., Ruiz, F.J., Blei, D.M.: Topic modeling in embedding spaces. Transactions of the Association for Computational Linguistics **8**, 439–453 (2020)
11. Doi, T., Isonuma, M., Yanaka, H.: Topic modeling for short texts with large language models. In: Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 4: Student Research Workshop). pp. 21–33 (2024)
12. Grootendorst, M.: Bertopic: Neural topic modeling with a class-based tf-idf procedure. arXiv preprint arXiv:2203.05794 (2022)

13. Hoffman, M., Bach, F., Blei, D.: Online learning for latent dirichlet allocation. advances in neural information processing systems **23** (2010)
14. Hofmann, T.: Probabilistic latent semantic indexing. In: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval. pp. 50–57 (1999)
15. Mimno, D., Wallach, H., Talley, E., Leenders, M., McCallum, A.: Optimizing semantic coherence in topic models. In: Proceedings of the 2011 conference on empirical methods in natural language processing. pp. 262–272 (2011)
16. Nan, F., Ding, R., Nallapati, R., Xiang, B.: Topic Modeling with Wasserstein Autoencoders. In: Korhonen, A., Traum, D., Màrquez, L. (eds.) Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 6345–6381. Association for Computational Linguistics, Florence, Italy (July 2019).
17. Pham, C.M., Hoyle, A., Sun, S., Resnik, P., Iyyer, M.: Topicgpt: A prompt-based topic modeling framework. arXiv preprint arXiv:2311.01449 (2023)
18. Srivastava, A., Sutton, C.: Autoencoding variational inference for topic models. arXiv preprint arXiv:1703.01488 (2017)
19. Wu, X., Nguyen, T., Luu, A.T.: A survey on neural topic models: methods, applications, and challenges. Artificial Intelligence Review **57**(2), 18 (2024)
20. Wu, X., Nguyen, T., Zhang, D., Wang, W.Y., Luu, A.T.: Fastopic: Pretrained transformer is a fast, adaptive, stable, and transferable topic model. Advances in Neural Information Processing Systems **37**, 84447–84481 (2024)