DynaMiTE: Discovering Explosive Topic Evolutions with User Guidance

Nishant Balepur[‡*] **Shivam Agarwal**[‡*] **Karthik Ramanan**[‡]

Susik Yoon[‡] **Jiawei Han**[‡] **Diyi Yang**[★]

[‡]University of Illinois at Urbana-Champaign, [★]Stanford University
{balepur2,shivama2,kv16,susik,hanj}@illinois.edu, diyiy@stanford.edu

Abstract

Dynamic topic models (DTMs) analyze text streams to capture the evolution of topics. Despite their popularity, existing DTMs are either fully supervised, requiring expensive human annotations, or fully unsupervised, generating topic evolutions that often do not cater to a user's needs. Further, the topic evolutions produced by DTMs tend to contain generic terms that are not indicative of their designated time steps. To address these issues, we propose the task of discriminative dynamic topic discovery. This task aims to discover topic evolutions from temporal corpora that distinctly align with a set of user-provided category names and uniquely capture topics at each time step. We solve this task by developing DynaMiTE, a framework that ensembles semantic similarity, category indicative, and time indicative scores to produce informative topic evolutions. Through experiments on three diverse datasets, including the use of a newly-designed human evaluation experiment, we demonstrate that DynaMiTE is a practical and efficient framework for helping users discover high-quality topic evolutions suited to their interests[1].

1 Introduction

Dynamic topic models (DTMs) seek to capture the evolution of topics in time-stamped documents (Blei and Lafferty, 2006). These models can be applied to many downstream tasks, including studying breakthroughs in scientific research (Uban et al., 2021), discovering global issues in parliamentary debates (Müller-Hansen et al., 2021; Guldi, 2019), and tracking evolving news stories (Li et al., 2020; Vaca et al., 2014; Yoon et al., 2023b). As information and language continuously evolve, DTMs are important tools for communicating these changes to users (Vosecky et al., 2013; Dieng et al., 2019).

Evolution	2013	2017	2021
DNLDA *NLP*	language multilingual sentence	language english chinese	models tasks language
DNLDA *NNs*	results full connection	cnn filters learn	architecture cnn accuracy
Ours *NLP*	fsl speech rec. translation	stance detection nli sts	plm xlm-roberta mbert
Ours *NNs*	tnn neuron mult. noise	gru overparameterize pointnet	ntk infinite-width qnn

Table 1: Evolution from unsupervised DTM DNLDA (Churchill and Singh, 2022) for topics *natural language processing* (NLP) and *neural networks* (NNs) on Arxiv machine learning papers, compared to our output.

Existing DTMs are either fully supervised or fully unsupervised, both of which have their own limitations. To uncover topic evolutions in document collections, supervised DTMs (Park et al., 2015; Jiang, 2015) require each document to have a topic label. However, obtaining such topic labels requires annotating the document collection, which can be expensive and time-consuming. Hence, unsupervised DTMs (Blei and Lafferty, 2006; Wei et al., 2007; Zhang and Lauw, 2022; Grootendorst, 2022) are a more practical and popular approach, as they can be applied to unlabeled document collections. Despite their widespread usage, we observe two drawbacks of unsupervised DTMs that limit their effectiveness in downstream applications.

First, unsupervised DTMs fail to consider their users' needs, such as specific *topics* or *categories* of interest[2]. Hence, the discovered topics may not

*Equal contribution.

[1]We release our code at https://github.com/nbalepur/DynaMiTE

[2]We use *topics* and *categories* interchangeably.

be completely interpretable or relevant to the user (Chang et al., 2009). For example in Table 1 (red), the unsupervised DTM retrieves generic terms like "learn" and "results" which are not distinctly related to the desired topic of *NNs*. These terms also overlap with *NLP*, another topic of the user's interests. As shown in Table 1 (blue), it would be more informative to return specific models ("tnn") and techniques ("ntk") discussed primarily in the context of *NNs*. These *category indicative terms* promote a deeper understanding of the topics of interest, increase the likelihood that the retrieved outputs satisfy a user's needs, and enhance downstream tasks such as content discovery and corpus summarization (Wang et al., 2009; Boyd-Graber et al., 2017; Yoon et al., 2023a).

Second, unsupervised DTMs fail to distinguish between terms that are generic and terms that are distinct to each time step. For example in Table 1 (red), the unsupervised DTM retrieves "languages" for *NLP* at each time step, which is redundant and does not capture the field's evolution from 2013 to 2021 (Sun et al., 2022). As shown in Table 1 (blue), a user would be more informed by terms that uniquely characterize *NLP* in each year, such as "stance detection" in 2017 and "mbert" in 2021. Such *time indicative terms* provide clearer insights into how a topic has changed and they can aid users in downstream tasks, such as associating concepts with specific time steps (§5.4) and identifying key shifts in successive years (§6.4).

To address the above shortcomings, we introduce a new task, *discriminative dynamic topic discovery*, which aims to create informative topic evolutions suited to a user's needs. We minimally represent a user's interests as a set of provided category names or seeds, i.e., terms present in the input corpus. A discriminative dynamic topic discovery framework must produce evolving topics for each seed that are distinctly relevant to the category and time step.

For this task, we develop **DynaMiTE**, an iterative framework to **Dyna**mically **Mi**ne **T**opics with Category S**e**eds. Avoiding the pitfalls of existing DTMs, DynaMiTE combines three scores to ensure that candidate terms are (1) semantically similar to a user's interests, (2) popular in documents indicative of the user-specified category, and (3) indicative of the corresponding time step. We briefly describe these scores as follows:

(1) Semantic Similarity Score: Combining the strengths of category-guided and temporal embedding spaces, we propose a *discriminative dynamic word embedding model* to compare the semantics of candidate terms and user-provided seeds (§4.1).

(2) Category Indicative Score: We assume that high-quality candidate terms related to a user-provided category name are likely to be found in documents that discuss the category name. Thus, we calculate a term's distinct popularity in a set of retrieved *category indicative documents* (§4.2).

(3) Time Indicative Score: To discover candidate terms that uniquely capture time steps, we introduce a time indicative score based on *topic burstiness*. We seek candidate terms whose popularity rapidly explodes and defuses (§4.3).

DynaMiTE ensembles these three scores after every training iteration to mine a single term for each time step and each category (§4.4). These terms are used to refine the discriminative dynamic word embeddings and category indicative document retrieval, resulting in informative topic evolutions. We present DynaMiTE as a fast, simple, and effective tool for aiding trend and evolution exploration.

Our contributions can be summarized as follows:

- We propose a new task, discriminative dynamic topic discovery, which produces informative topic evolutions relevant to a set of user-provided seeds.
- We develop DynaMiTE, which iteratively learns from discriminative dynamic embeddings, document retrieval, and topic burstiness to discover high-quality topic evolutions suited to a user's needs.
- We design a new human evaluation experiment to evaluate discriminative dynamic topic discovery. We find that users prefer DynaMiTE due to its retrieval of category and time indicative terms.
- Through experiments on three diverse datasets, we observe that DynaMiTE outperforms state-of-the-art DTMs in terms of topic quality and speed.

2 Related Work

We outline two variations on topic mining which incorporate time and user guidance, respectively.

2.1 Dynamic Topic Modeling

Many popular unsupervised DTMs (Blei and Lafferty, 2006; Churchill and Singh, 2022) build upon

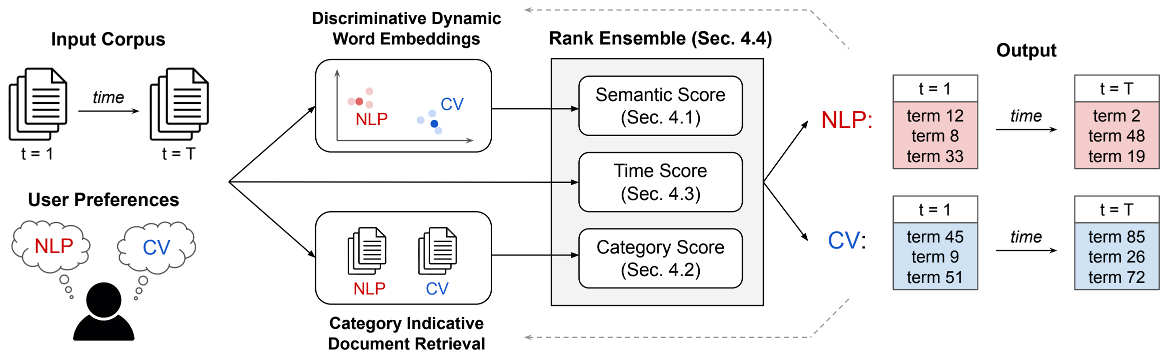

Figure 1: Overview of DynaMiTE. Given a temporal collection of documents and user-provided seeds, DynaMiTE first calculates semantic similarity scores with discriminative dynamic word embeddings, category indicative scores with document retrieval, and time indicative scores based on topic burstiness. Ensembling these scores, DynaMiTE iteratively mines topic evolutions and uses this information to further enrich its outputs.

LDA (Blei et al., 2003), where each document in a corpus is drawn from a generative process. Typically, inference on this process is performed through variational approximation (Wei et al., 2007; Jähnichen et al., 2018) or Gibbs Sampling (Iwata et al., 2009; Bhadury et al., 2016). Subsequent DTMs incorporate continuous timestamps (Wang and McCallum, 2006; Wang et al., 2008) and multiple timescales (Iwata et al., 2010; Nallapati et al., 2007; Chen et al., 2018). Recent embedding-based DTMs (Dieng et al., 2019) aim to address the limitations of LDA-based models, such as the inability to model the semantics of words. Leveraging transformers, BERTopic (Grootendorst, 2022) represents dynamic topics as evolving clusters. Dynamic word embeddings (Rudolph and Blei, 2018; Yao et al., 2018), which capture the evolution of language, can use semantic similarity to retrieve evolving topics.

A drawback common to all aforementioned approaches is the inability to incorporate user guidance. We address this limitation by enabling users to specify seeds for each topic evolution. Further, there does exist a small family of supervised DTMs (Park et al., 2015; Jiang, 2015), but these models can only be used on labeled document corpora, and thus cannot be directly applied for our setting.

2.2 User-guided Topic Discovery

Varying forms of guidance have been integrated into non-dynamic topic models. SeededLDA (Jagarlamudi et al., 2012) generates topics with user-given “seed topics”. Later methods allow users to specify whether pairs of words should be generated by the same topics (Andrzejewski and Zhu, 2009) and anchor specific words to topics (Gallagher et al., 2017). Recently, user queries have been used to guide topic models (Fang et al., 2021).

More relevant to our task are models that iteratively expand upon a set of user-provided seeds. GTM (Churchill et al., 2022) uses Generalized Polya Urn sampling (Mimno et al., 2011) to learn topics based on user-given seeds. Embedding-based approaches such as CatE (Meng et al., 2020) learn discriminative embeddings for user-provided categories. Recent seed-guided topic mining works (Zhang et al., 2022a,b) use language model representations and topical sentences to improve CatE.

These works assume a non-dynamic corpus and thus cannot discover topic evolutions from temporal corpora, which is the main focus of this paper.

3 Problem Definition

We define *discriminative dynamic topic discovery* as follows: Given a corpus of time-stamped document collections $\mathcal{D} = \{\mathcal{D}_1, \mathcal{D}_2, ..., \mathcal{D}_T\}$ and a set of user-provided seeds $\mathcal{C} = \{c_1, c_2, ..., c_n\}$, discriminative dynamic topic discovery aims to retrieve topic evolutions $\{\mathcal{S}_{tj}\}_{t=1}^{T}$ for each category c_j. The topic $\mathcal{S}_{tj}$ contains a list of terms $\{w_1, w_2, ..., w_m\}$ that are discriminatively relevant to time t and category c_j. The time steps $\mathcal{T} = \{1, ..., T\}$ are any ordinal measure of time and can vary depending on the granularity required.

4 Methodology

To solve discriminative dynamic topic mining, we propose **DynaMiTE**, which iteratively populates each topic $\mathcal{S}_{tj}$. Each topic $\mathcal{S}_{tj}$ initially contains just the category name c_j, and after every training iteration of DynaMiTE, we expand each $\mathcal{S}_{tj}$ with a single term w. For a term w to be added to $\mathcal{S}_{tj}$, we require three conditions to be satisfied: (1) w must be semantically similar to $\mathcal{S}_{tj}$; (2) w must be prevalent in documents which discuss $\mathcal{S}_{tj}$; (3) w must be a time indicative word of time t.

We achieve these three goals by calculating three respective scores for candidate terms, namely **semantic similarity scores** with discriminative dynamic word embeddings (§4.1), **category indicative scores** from retrieved category indicative documents (§4.2), and **time indicative scores** based on topic burstiness (§4.3). Combining these scores (§4.4), we can iteratively mine terms and use this information to further enrich our framework, illustrated in Figure 1 and detailed in Algorithm 1.

4.1 Semantic Similarity Score

Static word embeddings (Mikolov et al., 2013; Pennington et al., 2014) are one option to compute the semantic similarity between candidate terms and user-provided categories. However, static embeddings do not consider the category and time dimensions, thus losing the ability to model category distinctive information (Meng et al., 2020) and capture evolving semantics (Bamler and Mandt, 2017). Hence, we combine the category and time dimensions into a single discriminative dynamic word embedding model based on Yao et al. (2018).

Given a temporal corpus $\mathcal{D}$, we seek to model the semantics of every word $w \in \mathcal{D}$ at every time step t. To do so, we wish to find a word embedding matrix $U(t) \in \mathbb{R}^{V \times d}$ for each time t, where V is the vocabulary size and d is the word embedding dimension. We assume that $U(t)$ is affected by *local contexts*, *temporal contexts*, and *user guidance*.

Local Contexts: To learn accurate word semantics for topic discovery, it is essential to go beyond the bag-of-words assumption of LDA (Meng et al., 2020). Thus, we follow skip-gram (Mikolov et al., 2013) and assume that the semantics of surrounding words w_j in a local context window of size h (i.e., $[i - \frac{h}{2}, i + \frac{h}{2}]$) are influenced by the semantics of the center word w_i. To learn semantics from local contexts for matrix $U(t)$, we leverage the fact that skip-gram word embeddings can be obtained by factoring the $V \times V$ pointwise mutual information (PMI) matrix of $\mathcal{D}_t$ (Levy and Goldberg, 2014), i.e.

$$\mathrm{PMI}(x, y) = \log \frac{p(x, y)}{p(x)p(y)} \approx U(t)U(t)^T. \quad (1)$$

$p(x)$ is the proportion of words in $\mathcal{D}_t$ that are the word x. $p(x, y)$ is the number of co-occurrences of words x and y within windows of size h, divided by total number of possible window-pairs. We extend this idea and find that the *positive normalized* PMI (PNPMI) matrix is just as effective, defined as:

$$\mathrm{PNPMI}(x, y) = \max\left\{\frac{\mathrm{PMI}(x, y)}{\log(p(x, y))}, 0\right\}. \quad (2)$$

We learn local contexts by minimizing the distance between $U(t)U(t)^T$ and PNPMI matrix $Y(t)$:

$$\lambda_{local}(t) = \left\|Y(t) - U(t)U(t)^T\right\|_F^2. \quad (3)$$

We choose PNPMI over PMI because it is bounded between 0 and 1, allowing us to easily modify the similarity of specific word embeddings when we later add user guidance. Specifically, manually setting $\mathrm{PNPMI}(x, y) = 0$ (or 1) implies that x and y have independent (or complete) co-occurrences in local context windows of size h, in turn causing x and y to have dissimilar (or similar) embeddings.

Temporal Contexts: As words change meaning over time, so should their embedding space representations (Bamler and Mandt, 2017). Hence, we follow the assumption that semantics drift slightly between successive time steps and control the distance between neighboring embeddings:

$$\lambda_{temp}(t) = \|U(t+1) - U(t)\|_F^2. \quad (4)$$

With temporally aligned embeddings, DynaMiTE can address issues of data sparsity by borrowing semantics from neighboring time steps. This process also allows us to identify significant shifts in category semantics between successive time steps, which we explore in our experiments section (§6.4).

User Guidance: Separating categories in the embedding space will enforce a stronger understanding of category names, as categories will become clusters surrounded by category distinct terms (Meng et al., 2020). For example, representing the categories *NLP* and *NNs* as separated clusters in the embedding space will cause overlapping, generic terms like “results” to fall between these clusters. Thus, overlapping terms will no longer be semantically similar to either category. To form these clusters at each time t, we adjust the embedding space so words in the same topic have similar embeddings and words in different topics have dissimilar embeddings. As discussed in §4.1, we can do this by forming a category discriminative matrix $Z(t) \in \mathbb{R}^{V \times V}$ to modify specific PNPMI values:

$$Z(t)_{x,y} = \begin{cases} 1, & x, y \in \mathcal{S}_{ti} \\ 0, & x \in \mathcal{S}_{ti}, y \in \mathcal{S}_{tj}, i \neq j \\ \mathrm{PNPMI}(x, y), & x \text{ or } y \text{ in no topics at } t \end{cases} \quad (5)$$

By minimizing the distance between $U(t)U(t)^T$ and $Z(t)$, we form category distinct clusters which become more refined as every topic $\mathcal{S}_{tj}$ grows:

$$\lambda_{user}(t) = \left\| Z(t) - U(t)U(t)^T \right\|_F^2. \quad (6)$$

Discriminative Dynamic Word Embeddings: By combining the loss terms of local contexts (Eq. 3), temporal contexts (Eq. 4), and user guidance (Eq. 6), we can jointly capture a category discriminative and temporal embedding space for $\mathcal{D}$:

$$\lambda = \alpha \sum_{t=1}^{T} \lambda_{local}(t) + \tau \sum_{t=1}^{T-1} \lambda_{temp}(t) + \kappa \sum_{t=1}^{T} \lambda_{user}(t). \quad (7)$$

We also add a loss term $\gamma \sum_{t=1}^{T} \|U(t)\|_F^2$ to encourage low-rank data fidelity. $\alpha, \tau, \kappa, \gamma$ are hyperparameters. We efficiently minimize λ with Block Coordinate Descent (Tseng, 2001) in Appendix A.

We calculate the **semantic similarity score** between candidate term w and topic $\mathcal{S}_{tj}$ by computing the cosine similarity of their embeddings. We obtain u_{tw}, the embedding of w, directly from the matrix $U(t)$. To obtain u_{ts}, the embedding of topic $\mathcal{S}_{tj}$, we average the embeddings of the terms that have been assigned to the topic, i.e., $w' \in \mathcal{S}_{tj}$:

$$\text{score}_S(w|\mathcal{S}_{tj}) = \frac{u_{tw} \cdot u_{ts}}{\|u_{tw}\| \, \|u_{ts}\|}. \quad (8)$$

4.2 Category Indicative Score

Skip-gram embeddings treat local contexts equally, regardless of whether the context is indicative of the category. However, a topic evolution that is distinctly relevant to its respective category should prioritize terms discussed in category indicative contexts. For example, "Chernobyl," a high-quality term for the category of *disaster*, is more likely to be discussed when the focus of the discourse is on *disasters*. To achieve this outcome, we follow previous works (Tao et al., 2016; Zhang et al., 2022b) and leverage the current topic evolution output to iteratively retrieve and quantify a candidate term's distinct popularity in category indicative contexts.

We assume that the category indicative contexts of time step t and category c_j can be represented as a set of documents $\Theta_{tj} \subseteq \mathcal{D}_t$. To obtain Θ_{tj}, we search $\mathcal{D}_t$ and select documents which contain any of the terms in $\mathcal{S}_{tj}$. Thus, Θ_{tj} is updated iteratively as $\mathcal{S}_{tj}$ grows. We calculate the relevance of candidate term w to Θ_{tj} through *popularity* (how often does term w appear in Θ_{tj}) and *distinctiveness* (how unique is term w to Θ_{tj} compared to other category indicative documents). Popularity deprioritizes hyper-specific terms, such as models uniquely introduced in an abstract, while distinctiveness deprioritizes generic terms. For popularity, we choose the logarithm of term frequency (TF) and for distinctiveness, we choose the softmax of BM-25 (Robertson et al., 1995) relevance:

$$\text{pop}(w, \Theta_{tj}) = \log(\text{TF}(w, \Theta_{tj}) + 1) \quad (9)$$

$$\text{dist}(w, \Theta_{tj}) = \frac{e^{\text{BM-25}(w, \Theta_{tj})}}{\sum_{i=1}^{n} e^{\text{BM-25}(w, \Theta_{ti})}}. \quad (10)$$

We also experimented with TF-IDF (Ramos, 2003) and Dense Passage Retrieval (Karpukhin et al., 2020) instead of BM-25, but selected BM-25 due to its balance of efficiency and performance. Combining popularity and distinctiveness, we can form a **category indicative score** for candidate term w:

$$\text{score}_C(w|\mathcal{S}_{tj}) = \text{pop}(w, \Theta_{tj})^{\beta} \, \text{dist}(w, \Theta_{tj})^{1-\beta}, \quad (11)$$

where $0 \leq \beta \leq 1$ is a hyperparameter.

4.3 Time Indicative Score

Previous works have demonstrated that topic evolutions can uniquely capture time steps when they contain a strong temporal ordering of burst topics (Kleinberg, 2002; Leskovec et al., 2009). For example, "ELMo" is a high-quality term that uniquely captures *NLP* in 2018, since it abruptly spiked in popularity when it was released that year. Thus, to improve the informativeness of our retrieved terms at each time t, we focus on terms that explode in popularity at t but are not popular before and after t. Motivated by the success of modifying TF-IDF for the temporal setting (Lee et al., 2011; Alsaedi et al., 2016; Zhang et al., 2022c), we develop a burst TF-IDF metric to obtain a time indicative score. We define the popularity of term w at time t by term frequency (TF), normalized by the number of documents in $\mathcal{D}_t$. To model if w is popular at time steps outside of t, we develop a burst inverse time frequency (BITF) metric, calculated as the logarithm of the inverse proportion of time steps, within a temporal window of size r (i.e., $[t - \frac{r}{2}, t + \frac{r}{2}]$), in which w appeared. We combine these metrics to calculate a **time indicative score** as follows:

$$\text{BITF}(t, w) = \frac{r}{\sum_{i=t-r/2}^{t+r/2} I(w \in \mathcal{D}_i)} \quad (12)$$

$$\text{score}_B(w|t) = \frac{\text{TF}(w)}{|\mathcal{D}_t|} \log(\text{BITF}(t, w)), \quad (13)$$

where I is the indicator function.

Algorithm 1 DynaMiTE

procedure DYNAMITE($\mathcal{D}, \mathcal{C}, \mathcal{T}, N$)
  Calculate score$_B(w, t), \forall w \in \mathcal{D}$
  Initialize dynamic embeddings
  Initialize each $\mathcal{S}_{tj}$ with c_j
  for iter $\leftarrow$ 1 to N **do**
    Update embeddings with Eq. (7)
    Retrieve $\Theta_{tj} \subseteq \mathcal{D}_t, \forall c_j \in \mathcal{C}, t \in \mathcal{T}$
    for $c_j \in \mathcal{C}$ **do**
      for $t \in \mathcal{T}$ **do**
        Calculate score$_S(w, \mathcal{S}_{tj}), \forall w \in \mathcal{D}_t$
        Calculate score$_C(w, \mathcal{S}_{tj}), \forall w \in \mathcal{D}_t$
        Ensemble scores into MR
        Sort all $w \in \mathcal{D}_t$ by MR
        Update $\mathcal{S}_{tj}$ with best w
  return $\{\mathcal{S}_{tj} | t \in \mathcal{T}, c_j \in \mathcal{C}\}$

4.4 The Iterative DynaMiTE Framework

We summarize DynaMiTE in Algorithm 1. Before training, we calculate every time indicative score, as it does not depend on the iterative topic evolutions. During each training iteration of DynaMiTE, we update the discriminative dynamic word embeddings according to Eq. 7 and retrieve all category indicative documents Θ_{tj}. Then, for every category $c_j \in \mathcal{C}$ and time $t \in \mathcal{T}$, we rank candidate terms in descending order by semantic similarity, category indicative, and time indicative scores, as follows:

$$\mathrm{r}_S(w|\mathcal{S}_{tj}) = \mathrm{argsort}(\{-\mathrm{score}_S(w, \mathcal{S}_{tj}) | w \in \mathcal{D}_t\}). \quad (14)$$

$\mathrm{r}_C(w|\mathcal{S}_{tj})$ and $\mathrm{r}_B(w|t)$ are similarly defined. To ensemble the ranks, we obtain the mean rank (MR):

$$\mathrm{MR}(w|\mathcal{S}_{tj}) = \frac{1}{3}(\mathrm{r}_S(w|\mathcal{S}_{tj}) + \mathrm{r}_C(w|\mathcal{S}_{tj}) + \mathrm{r}_B(w|t)). \quad (15)$$

The term with the lowest mean rank that does not exist in any topics at time t is added to each topic $\mathcal{S}_{tj}$. To obtain N unique terms for each topic $\mathcal{S}_{tj}$, we repeat the process of semantic modeling, document retrieval, and term ranking for N iterations.

5 Experimental Setup

We present a detailed setup in Appendix B.

5.1 Datasets

We conduct experiments on three datasets from different domains. **(1) Arxiv** (arXiv.org submitters, 2023) is a corpus of titles and abstracts of 214k machine learning papers from 2012 to 2022. We group them by year (11 time steps) and use *neural network*, *natural language processing*, and *computer vision* as seeds. **(2) UN** (Baturo et al., 2017) contains 250k speeches from the United Nations Debate Corpus, discussing global issues from 1970 to 2017. We group them into spans of four years (12 time steps) and choose *disaster* and *leader* as seeds. **(3) Newspop** (Moniz and Torgo, 2018) is a dataset of 93k headlines shared by major news outlets on social media from Oct. 2015 to Jul. 2016. We group posts by month (10 time steps) and choose *politics, obama* and *technology, microsoft* as seeds.

5.2 Baselines

We compare DynaMiTE with the following baselines: **DNLDA** (Churchill and Singh, 2022) is an unsupervised DTM based on LDA which jointly models topics and noise. **BERTopic** (Grootendorst, 2022) is an unsupervised DTM that clusters terms into dynamic topics. For the unsupervised DTMs, we manually select the best topic evolution for each category. **Bernoulli** (Rudolph and Blei, 2018) are dynamic word embeddings based on exponential family embeddings. **DW2V** (Yao et al., 2018) learns time-aware word embeddings based on skip-grams. For the embedding-based methods, we use cosine similarity to retrieve topic evolutions. **CatE** (Meng et al., 2020) is a seed-guided topic mining framework that learns discriminative category embeddings. We run CatE recursively on each corpus $\mathcal{D}_t$ to obtain topic evolutions.

5.3 Quantitative Metrics

We evaluate all models quantitatively using normalized pointwise mutual information (NPMI), a standard measure of topic coherence (Lau et al., 2014). We calculate the NPMI of 5 terms in each time t with respect to $\mathcal{D}_t$ and report their mean as a percentage (mean of 25 runs).

5.4 Human Experiments

Previous works have shown that topic coherence metrics like NPMI do not always align with topic quality (Hoyle et al., 2021; Lau et al., 2014). Thus, we conduct two human experiments to qualitatively evaluate topic evolutions. For both experiments, we design an interface using PrairieLearn (West et al., 2015) and invite three graduate students with knowledge of the three domains to annotate. We encourage them to use Google or any other resources to aid them. We provide a detailed human evaluation setup and screenshots in Appendix B.6.

(1) Term Accuracy: Term accuracy measures whether users are satisfied by the discovered topics of DTMs. We evaluate term accuracy by asking annotators if each term in the topic evolution uniquely “belongs” to its category and does not “belong” to

Method	Arxiv				UN				Newspop			
	NPMI	MACC	Rank	Conf	NPMI	MACC	Rank	Conf	NPMI	MACC	Rank	Conf
DynaMiTE (ours)	**7.80***	**0.802***	**0.878***	**4.00***	**8.28***	**0.871***	**0.934***	**4.83***	4.04	**0.770***	**0.892***	**4.33***
DNLDA (2022)	3.54	0.303	0.267	1.67	4.66	0.133	-0.063	1.00	3.10	0.210	0.218	1.00
BERTopic (2022)	7.53	0.403	-0.056	2.11	7.58	0.208	0.164	1.50	5.09	0.300	-0.191	2.00
Bernoulli (2018)	6.82	0.236	-0.180	1.11	7.60	0.108	0.247	1.17	3.65	0.640*	-0.206	1.17
DW2V (2018)	4.71	0.148	0.015	1.00	7.68	0.225	-0.142	1.50	2.67	0.345	0.115	1.17
CatE (2020)	6.38	0.394	0.164	1.67	6.83	0.088	-0.107	1.33	**5.37***	0.415	0.197	2.17

Table 2: Topic coherence (NPMI), term accuracy (MACC), and temporal quality (Rank and Conf) comparison. Models with metrics marked with * significantly outperform all non-marked baselines ($p < 0.05$ approximate randomization test (Noreen, 1989) for NPMI, $p < 0.005$ Wilcoxon signed-rank test (Woolson, 2007) for MACC and Conf, $p < 0.005$ permutation test (Dietz, 1983) for Rank). We follow Dror et al. (2018) to pick statistical tests.

Method	*Disaster* 1986 - 1989	*Disaster* 1990 - 1993	*Disaster* 1994 - 1997	*Leader* 1986 - 1989	*Leader* 1990 - 1993	*Leader* 1994 - 1997
DynaMiTE (ours)	chernobyl locusts hurricane hugo	chernobyl devastating earthquake iraqi invasion of kuwait	montserrat hurricane luis igadd	mr gorbachev shultz president reagan	npfl mr nelson mandela klerk	mahmoud npfl ulimo
DNLDA (2022)	lebanon lebanese (×) appeal (×)	bosnia herzegovina republic (×)	clear (×) strong (×) failure (×)	political (×) developments (×) continue (×)	president government (×) de (×)	road (×) theme (×) ahead (×)
BERTopic (2022)	natural disasters recent experiences (×) natural disaster	chernobyl chernobyl disaster coordinator (×)	natural disasters natural disaster disasters (×)	word leaders (×) virtuous (×) leadership (×)	word leaders (×) leadership (×) leaders (×)	word leaders (×) leadership (×) leaders (×)
Bernoulli (2018)	pushed (×) brink (×) worried (×)	pushed (×) nuclear conflagration worried (×)	pushed (×) nuclear conflagration worried (×)	demise (×) grief (×) excellency president	demise (×) grief (×) excellency president	demise (×) excellency president grief (×)
DW2V (2018)	catastrophe (×) earthquakes disasters (×)	catastrophe (×) earthquakes disasters (×)	catastrophe (×) disasters (×) earthquakes	great leader (×) hero (×) immortal (×)	great leader (×) hero (×) immortal (×)	great leader (×) hero (×) kim jong il
CatE (2020)	distorting (×) east-west atmosphere	international climate sustained development (×) atmosphere	exacerbation (×) international climate sustained development	fundamental freedoms (×) human rights (×) protection (×)	trampled (×) fundamental human rights (×) elementary (×)	international covenants (×) civil rights (×) fundamental freedoms (×)

Table 3: Qualitative assessment of 3-term topic evolution on UN dataset, using a random sample of consecutive time steps for brevity. Terms marked with (×) were determined not to belong to their category by over half of annotators.

other categories. We define "belongs" as any non-synonym relation (to avoid low-quality terms such as "tragedy" for *disaster*) between the term and the category. For reference, we provide annotators with relations from ConceptNet (Speer et al., 2017). We average the labeling of annotators and report the final results as mean accuracy (**MACC**). We find high inter-annotator agreement for MACC, with Fleiss' kappa (Fleiss, 1971) scores of 88, 86, 84 for Arxiv, UN, and, Newspop, respectively.

(2) Temporal Quality: NPMI and MACC do not evaluate if topic evolutions capture interpretable trends. Thus, motivated by the definitions of interpretability for non-dynamic topic models proposed by Doogan and Buntine (2021), we propose that *an interpretable topic evolution is one that can be ordered chronologically*. To evaluate this property, we remove the label that indicates which time step each set of terms belongs to, as well as terms that reveal the time step of the set. We shuffle these sets and ask annotators to order them chronologically.

We use Spearman's rank correlation coefficient (**Rank**) (Zar, 2005) to measure how similar the annotator's order is to the true order of the topic evolution and ask annotators to rate their confidence (**Conf**) on a scale from 1 to 5 using Mean Opinion Score (Streijl et al., 2016), where 5 indicates total confidence. We report **Rank** and **Conf** averaged over seeds and annotators. To our knowledge, this is the first work with human experiments to evaluate the temporal quality of topic evolutions.

6 Results

6.1 Performance Comparison

Quantitative Results: In Table 2, we find that DynaMiTE produces high-quality topic evolutions, almost always achieving superior quantitative results. The only exception is NPMI on the Newspop dataset, where CatE and BERTopic obtain higher scores than DynaMiTE. The Newspop dataset contains short headlines, where category names do not co-occur frequently with the high-quality terms mined by DynaMiTE, reducing NPMI. We contend that DynaMiTE still mines more informative terms,

Table 4: MACC performance comparison of model ablations. -Temp and -Discr remove the loss terms from Eqs. 4 and 6, respectively. -Semantic, -Category, and -Time remove the respective scores (Eqs. 8, 11, 13). Darker shades of red (↓) indicate worse performance.

	Method	Arxiv	UN	Newspop
	DynaMiTE	**0.802**	**0.871**	**0.770**
Loss Terms	- Temp	0.745	0.638	0.690
	- Discr	0.700	0.621	0.705
Ranked Scores	- Semantic	0.555	0.488	0.655
	- Category	0.742	**0.871**	0.715
	- Time	0.667	0.238	0.380

as demonstrated by the human evaluation metrics in Table 2. Overall, our strong quantitative results suggest that DynaMiTE (1) directly addresses a user's search needs (MACC, NPMI) and (2) captures interpretable trends (Rank, Conf), making it a preferred choice for exploring temporal corpora.

Qualitative Results: In Table 3, we observe two desirable properties of the topic evolutions produced by DynaMiTE: (1) While other models retrieve generic terms weakly related to *disaster* and *leader* (e.g. "demise" and "coordinator"), DynaMiTE mines terms which are distinctly and directly related to each category name. We believe that the use of category discriminative embeddings and category indicative document retrieval helps DynaMiTE avoid this pitfall and achieve higher MACC scores. (2) While other models contain similar sets of terms over time, DynaMiTE uses topic burstiness to find terms that uniquely capture each time step. This explains why annotators performed the best and were most confident when ordering the shuffled outputs of DynaMiTE. For example, a quick Google search will show that Hurricane Hugo occurred in 1989, Iraq invaded Kuwait in 1990, and Hurricane Luis was recorded in 1995 (Wikipedia contributors, 2023a,b). We show all qualitative results of our model in Appendix C.1.

6.2 Ablation Study

We perform an ablation study (Table 4) to observe how users perceive the outputs of DynaMiTE when its different components are removed. To directly measure user preferences, we use MACC. We observe the following: **(1)** DynaMiTE outperforms all ablations in most cases, implying that all components of the model complement each other. **(2)** It is interesting to note that removing the time indicative score causes on average, a 46.7% drop in MACC.

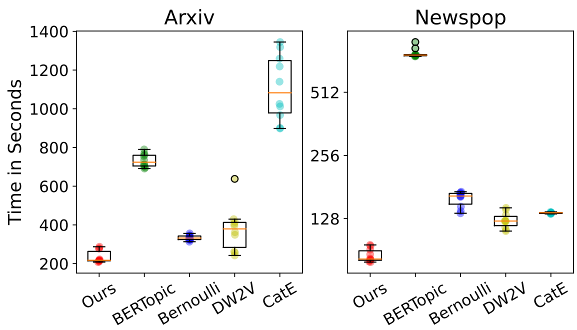

Figure 2: Runtime comparison (in seconds) for 5-term topic evolution retrieval on Arxiv and Newspop over ten runs. The right plot has a logarithmic y-axis scale. We omit DNLDA due to its poor performance (e.g. an average runtime of 5,117 seconds on Newspop).

This observation suggests a strong association between a term's distinct popularity within a temporal window and its perceived relevance to a category name. **(3)** After the time indicative score, removing the semantic similarity score leads to the next largest drop in MACC, being on average, 29.9%. Combining this observation with (2), we can infer that users prefer the full version of DynaMiTE due to its retrieval of terms both directly relevant to their interests and unique to each time step.

6.3 Runtime Comparison

DTMs are most often applied to rapidly changing domains, such as news and research, and thus benefit from running in real time. Further, efficient NLP frameworks greatly improve user experience (Telner, 2021). Hence, we study the runtime of DynaMiTE in Figure 2. We find that due to the combination of matrix factorization and Block Coordinate Descent to learn the embedding space, DynaMiTE achieves the fastest runtime on Arxiv and Newspop (UN follows the same trend). In addition, DynaMiTE operates entirely on CPUs, while BERTopic and Dynamic Bernoulli Embeddings require GPUs, making DynaMiTE a highly practical and resource-efficient solution for users.

6.4 Category Shift Analysis

We employ a discriminative dynamic embedding space with smoothness constraints over successive time steps to capture semantic shifts (Eq. 4). To study this property, we analyze the largest semantic shifts of our user-provided category names. First, we find the adjacent time steps t and $t-1$ where the embeddings of the category name are the most dissimilar. To pinpoint one contributor to this large

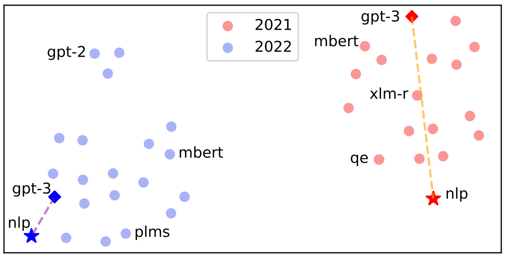

Figure 3: Discriminative dynamic embedding space of nearest neighbors to NLP in 2021 (red) and 2022 (blue) using t-SNE (van der Maaten and Hinton, 2008).

semantic shift, we identify the term whose embedding distance to the category name changed the most between t and $t-1$ using cosine similarity.

For the category of *natural language processing* on Arxiv, the largest semantic shift occurred between 2021 and 2022, with the main cause being "GPT-3." Our findings align with recent studies (Bommasani et al., 2021; Sun et al., 2022; Goyal et al., 2022) which suggest that GPT-3 has led to a paradigm shift in NLP, in turn changing the semantics of the category *NLP*. This phenomenon is visualized in Figure 3. We present more category shift experiments in the Appendix (Table 9).

7 Conclusion

We propose the new task of discriminative dynamic topic discovery and develop DynaMiTE to solve the task. Through experiments on three diverse datasets, including the design of a new human evaluation experiment, we demonstrate that DynaMiTE produces high-quality topic evolutions and outperforms state-of-the-art DTMs. Ablation studies show that DynaMiTE effectively addresses a user's needs by retrieving category and time indicative terms. Through runtime analyses, we find that DynaMiTE is a computationally efficient and practical tool. Finally, we probe the discriminative dynamic embedding space of DynaMiTE to identify key shifts in computer science, politics and news.

8 Limitations

Time Granularity: The granularity of time we test DynaMiTE on ranges from spans of four years to months. After testing multiple ways to bucket our temporal corpora, we observed that the granularity of time only affected DynaMiTE when there were insufficient documents in each time step. Specifically, we found that there must be at least 100 documents per time step to expect reasonably good results.

Runtime: One drawback of DynaMiTE is that its runtime depends on the number of terms required at each time step. However, this can be avoided by mining more than one term during each iteration of the framework. We also observed that DynaMiTE, along with all other dynamic topic mining baselines, had a slower performance on datasets with longer text documents.

Risks: DynaMiTE is intended to be used as a tool to discover topic evolutions in temporal corpora suited to a user's interests, represented as category seeds. We only experimented with DynaMiTE in domains with trustworthy information. If DynaMiTE was used in document collections that contain misinformation, it could have the potential to mine inaccurate terms.

9 Acknowledgements

Research was supported in part by US DARPA KAIROS Program No. FA8750-19-2-1004 and INCAS Program No. HR001121C0165, National Science Foundation IIS-19-56151, IIS-17-41317, and IIS 17-04532, and the Molecule Maker Lab Institute: An AI Research Institutes program supported by NSF under Award No. 2019897, and the Institute for Geospatial Understanding through an Integrative Discovery Environment (I-GUIDE) by NSF under Award No. 2118329. Any opinions, findings, and conclusions or recommendations expressed herein are those of the authors and do not necessarily represent the views, either expressed or implied, of DARPA or the U.S. Government.

This work was also partly supported by Basic Science Research Program through the National Research Foundation of Korea (2021R1A6A3A14043765).

References

Nasser Alsaedi, Pete Burnap, and Omer Rana. 2016. Temporal tf-idf: A high performance approach for event summarization in twitter. In *2016 IEEE/WIC/ACM International Conference on Web Intelligence (WI)*, pages 515–521.

David Andrzejewski and Xiaojin Zhu. 2009. Latent Dirichlet Allocation with topic-in-set knowledge. In *Proceedings of the NAACL HLT 2009 Workshop on Semi-supervised Learning for Natural Language Processing*, pages 43–48, Boulder, Colorado. Association for Computational Linguistics.

arXiv.org submitters. 2023. arxiv dataset.

Robert Bamler and Stephan Mandt. 2017. Dynamic word embeddings. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 380–389. PMLR.

Alexander Baturo, Niheer Dasandi, and Slava J. Mikhaylov. 2017. Understanding state preferences with text as data: Introducing the un general debate corpus. *Research & Politics*, 4(2):2053168017712821.

Arnab Bhadury, Jianfei Chen, Jun Zhu, and Shixia Liu. 2016. Scaling up dynamic topic models. In *Proceedings of the 25th International Conference on World Wide Web*, WWW '16, page 381–390, Republic and Canton of Geneva, CHE. International World Wide Web Conferences Steering Committee.

David M Blei and John D Lafferty. 2006. Dynamic topic models. In *Proceedings of the 23rd international conference on Machine learning*, pages 113–120.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3(null):993–1022.

Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. 2021. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*.

Jordan Boyd-Graber, Yuening Hu, David Mimno, et al. 2017. Applications of topic models. *Foundations and Trends® in Information Retrieval*, 11(2-3):143–296.

Jonathan Chang, Sean Gerrish, Chong Wang, Jordan Boyd-graber, and David Blei. 2009. Reading tea leaves: How humans interpret topic models. In *Advances in Neural Information Processing Systems*, volume 22. Curran Associates, Inc.

Xilun Chen, K. Selcuk Candan, and Maria Luisa Sapino. 2018. Ims-dtm: Incremental multi-scale dynamic topic models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1).

Rob Churchill and Lisa Singh. 2022. Dynamic topic-noise models for social media. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD)*.

Robert Churchill, Lisa Singh, Rebecca Ryan, and Pamela Davis-Kean. 2022. A guided topic-noise model for short texts. In *Proceedings of the ACM Web Conference 2022*, WWW '22, page 2870–2878, New York, NY, USA. Association for Computing Machinery.

Adji B. Dieng, Francisco J. R. Ruiz, and David M. Blei. 2019. The dynamic embedded topic model. *CoRR*, abs/1907.05545.

E Jacquelin Dietz. 1983. Permutation tests for association between two distance matrices. *Systematic Biology*, 32(1):21–26.

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

Rotem Dror, Gili Baumer, Segev Shlomov, and Roi Reichart. 2018. The hitchhiker's guide to testing statistical significance in natural language processing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1383–1392, Melbourne, Australia. Association for Computational Linguistics.

Zheng Fang, Yulan He, and Rob Procter. 2021. A query-driven topic model. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1764–1777, Online. Association for Computational Linguistics.

Joseph L Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological bulletin*, 76(5):378.

Ryan J. Gallagher, Kyle Reing, David Kale, and Greg Ver Steeg. 2017. Anchored correlation explanation: Topic modeling with minimal domain knowledge. *Transactions of the Association for Computational Linguistics*, 5:529–542.

Tanya Goyal, Junyi Jessy Li, and Greg Durrett. 2022. News summarization and evaluation in the era of gpt-3. *arXiv preprint arXiv:2209.12356*.

Maarten R. Grootendorst. 2022. Bertopic: Neural topic modeling with a class-based tf-idf procedure. *ArXiv*, abs/2203.05794.

Jo Guldi. 2019. Parliament's debates about infrastructure: an exercise in using dynamic topic models to synthesize historical change. *Technology and Culture*, 60(1):1–33.

Alexander Miserlis Hoyle, Pranav Goel, Andrew Hian-Cheong, Denis Peskov, Jordan L. Boyd-Graber, and Philip Resnik. 2021. Is automated topic model evaluation broken? the incoherence of coherence. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 2018–2033.

Tomoharu Iwata, Shinji Watanabe, Takeshi Yamada, and Naonori Ueda. 2009. Topic tracking model for analyzing consumer purchase behavior. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence*, IJCAI'09, page 1427–1432, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Tomoharu Iwata, Takeshi Yamada, Yasushi Sakurai, and Naonori Ueda. 2010. Online multiscale dynamic topic models. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 663–672.

Jagadeesh Jagarlamudi, Hal Daumé III, and Raghavendra Udupa. 2012. Incorporating lexical priors into topic models. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 204–213.

Patrick Jähnichen, Florian Wenzel, Marius Kloft, and Stephan Mandt. 2018. Scalable generalized dynamic topic models. In *International Conference on Artificial Intelligence and Statistics*, pages 1427–1435. PMLR.

Zhuoren Jiang. 2015. Chronological scientific information recommendation via supervised dynamic topic modeling. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*, WSDM '15, page 453–458, New York, NY, USA. Association for Computing Machinery.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

Jon Kleinberg. 2002. Bursty and hierarchical structure in streams. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '02, page 91–101, New York, NY, USA. Association for Computing Machinery.

Jey Han Lau, David Newman, and Timothy Baldwin. 2014. Machine reading tea leaves: Automatically evaluating topic coherence and topic model quality. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 530–539, Gothenburg, Sweden. Association for Computational Linguistics.

Chung-Hong Lee, Chih-Hong Wu, and Tzan-Feng Chien. 2011. Burst: A dynamic term weighting scheme for mining microblogging messages. In *Advances in Neural Networks – ISNN 2011*, pages 548–557, Berlin, Heidelberg. Springer Berlin Heidelberg.

Jure Leskovec, Lars Backstrom, and Jon Kleinberg. 2009. Meme-tracking and the dynamics of the news cycle. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '09, page 497–506, New York, NY, USA. Association for Computing Machinery.

Omer Levy and Yoav Goldberg. 2014. Neural word embedding as implicit matrix factorization. *Advances in neural information processing systems*, 27.

Yue Li, Pratheeksha Nair, Zhi Wen, Imane Chafi, Anya Okhmatovskaia, Guido Powell, Yannan Shen, and David Buckeridge. 2020. Global surveillance of covid-19 by mining news media using a multi-source dynamic embedded topic model. In *Proceedings of the 11th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics*, BCB '20, New York, NY, USA. Association for Computing Machinery.

Yu Meng, Jiaxin Huang, Guangyuan Wang, Zihan Wang, Chao Zhang, Yu Zhang, and Jiawei Han. 2020. Discriminative topic mining via category-name guided text embedding. In *Proceedings of The Web Conference 2020*, pages 2121–2132.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

David Mimno, Hanna Wallach, Edmund Talley, Miriam Leenders, and Andrew McCallum. 2011. Optimizing semantic coherence in topic models. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 262–272, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Nuno Moniz and Luís Torgo. 2018. Multi-source social feedback of online news feeds. *arXiv preprint arXiv:1801.07055*.

Finn Müller-Hansen, Max W Callaghan, Yuan Ting Lee, Anna Leipprand, Christian Flachsland, and Jan C Minx. 2021. Who cares about coal? analyzing 70 years of german parliamentary debates on coal with dynamic topic modeling. *Energy Research & Social Science*, 72:101869.

Ramesh M. Nallapati, Susan Ditmore, John D. Lafferty, and Kin Ung. 2007. Multiscale topic tomography. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '07, page 520–529, New York, NY, USA. Association for Computing Machinery.

Eric W Noreen. 1989. *Computer-intensive methods for testing hypotheses*. Wiley New York.

Sungrae Park, Wonsung Lee, and Il-Chul Moon. 2015. Supervised dynamic topic models for associative topic extraction with a numerical time series. In *Proceedings of the 2015 Workshop on Topic Models: Post-Processing and Applications*, TM '15, page 49–54, New York, NY, USA. Association for Computing Machinery.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Juan Enrique Ramos. 2003. Using tf-idf to determine word relevance in document queries.

Stephen E Robertson, Steve Walker, Susan Jones, Micheline M Hancock-Beaulieu, Mike Gatford, et al. 1995. Okapi at trec-3. *Nist Special Publication Sp*, 109:109.

Maja Rudolph and David Blei. 2018. Dynamic embeddings for language evolution. In *Proceedings of the 2018 World Wide Web Conference*, WWW '18, page 1003–1011, Republic and Canton of Geneva, CHE. International World Wide Web Conferences Steering Committee.

Jingbo Shang, Jialu Liu, Meng Jiang, Xiang Ren, Clare R. Voss, and Jiawei Han. 2018. Automated phrase mining from massive text corpora. *IEEE Transactions on Knowledge and Data Engineering*, 30(10):1825–1837.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Thirty-first AAAI conference on artificial intelligence*.

Robert C. Streijl, Stefan Winkler, and David S. Hands. 2016. Mean opinion score (mos) revisited: methods and applications, limitations and alternatives. *Multimedia Systems*, 22(2):213–227.

Tian-Xiang Sun, Xiang-Yang Liu, Xi-Peng Qiu, and Xuan-Jing Huang. 2022. Paradigm shift in natural language processing. *Machine Intelligence Research*, 19(3):169–183.

Fangbo Tao, Honglei Zhuang, Chi Wang Yu, Qi Wang, Taylor Cassidy, Lance M Kaplan, Clare R Voss, and Jiawei Han. 2016. Multi-dimensional, phrase-based summarization in text cubes. *IEEE Data Eng. Bull.*, 39(3):74–84.

Jason Telner. 2021. Chatbot user experience: Speed and content are king. In *Advances in Artificial Intelligence, Software and Systems Engineering*, pages 47–54, Cham. Springer International Publishing.

Paul Tseng. 2001. Convergence of a block coordinate descent method for nondifferentiable minimization. *Journal of Optimization Theory and Applications*, 109:475–494.

Ana Sabina Uban, Cornelia Caragea, and Liviu P Dinu. 2021. Studying the evolution of scientific topics and their relationships. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1908–1922.

Carmen K Vaca, Amin Mantrach, Alejandro Jaimes, and Marco Saerens. 2014. A time-based collective factorization for topic discovery and monitoring in news. In *Proceedings of the 23rd international conference on World wide web*, pages 527–538.

Laurens van der Maaten and Geoffrey E. Hinton. 2008. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9:2579–2605.

Jan Vosecky, Di Jiang, Kenneth Wai-Ting Leung, and Wilfred Ng. 2013. Dynamic multi-faceted topic discovery in twitter. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*, pages 879–884.

Chong Wang, David Blei, and David Heckerman. 2008. Continuous time dynamic topic models. In *Proceedings of the Twenty-Fourth Conference on Uncertainty in Artificial Intelligence*, UAI'08, page 579–586, Arlington, Virginia, USA. AUAI Press.

Dingding Wang, Shenghuo Zhu, Tao Li, and Yihong Gong. 2009. Multi-document summarization using sentence-based topic models. In *Proceedings of the ACL-IJCNLP 2009 conference short papers*, pages 297–300.

Xuerui Wang and Andrew McCallum. 2006. Topics over time: a non-markov continuous-time model of topical trends. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 424–433.

Xing Wei, Jimeng Sun, and Xuerui Wang. 2007. Dynamic mixture models for multiple time series. In *Proceedings of the 20th International Joint Conference on Artifical Intelligence*, IJCAI'07, page 2909–2914, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Matthew West, Geoffrey L Herman, and Craig Zilles. 2015. Prairielearn: Mastery-based online problem solving with adaptive scoring and recommendations driven by machine learning. In *2015 ASEE Annual Conference & Exposition*, pages 26–1238.

Wikipedia contributors. 2023a. Iraqi invasion of kuwait — Wikipedia, the free encyclopedia. `https://en.wikipedia.org/w/index.php?title=Iraqi_invasion_of_Kuwait&oldid=1132623180`. [Online; accessed 19-January-2023].

Wikipedia contributors. 2023b. List of atlantic hurricane records — Wikipedia, the free encyclopedia. [Online; accessed 19-January-2023].

Robert F Woolson. 2007. Wilcoxon signed-rank test. *Wiley encyclopedia of clinical trials*, pages 1–3.

Zijun Yao, Yifan Sun, Weicong Ding, Nikhil Rao, and Hui Xiong. 2018. Dynamic word embeddings for evolving semantic discovery. In *Proceedings of the eleventh acm international conference on web search and data mining*, pages 673–681.

Susik Yoon, Hou Pong Chan, and Jiawei Han. 2023a. Pdsum: Prototype-driven continuous summarization of evolving multi-document sets stream. In *Proceedings of the ACM Web Conference 2023*, pages 1650–1661.

Susik Yoon, Yu Meng, Dongha Lee, and Jiawei Han. 2023b. Scstory: Self-supervised and continual online story discovery. In *Proceedings of the ACM Web Conference 2023*, pages 1853–1864.

Jerrold H Zar. 2005. Spearman rank correlation. *Encyclopedia of biostatistics*, 7.

Delvin Ce Zhang and Hady Lauw. 2022. Dynamic topic models for temporal document networks. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 26281–26292. PMLR.

Yu Zhang, Yu Meng, Xuan Wang, Sheng Wang, and Jiawei Han. 2022a. Seed-guided topic discovery with out-of-vocabulary seeds. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 279–290, Seattle, United States. Association for Computational Linguistics.

Yu Zhang, Yunyi Zhang, Martin Michalski, Yucheng Jiang, Yu Meng, and Jiawei Han. 2022b. Effective seed-guided topic discovery by integrating multiple types of contexts. *arXiv preprint arXiv:2212.06002*.

Yunyi Zhang, Fang Guo, Jiaming Shen, and Jiawei Han. 2022c. Unsupervised key event detection from massive text corpora. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '22, page 2535–2544, New York, NY, USA. Association for Computing Machinery.

A Discriminative Dynamic Word Embeddings Optimization

In this section, we detail the exact optimization process for Eq. 7, which follows similar steps as Yao et al. (2018). We first add an extra parameter designating the embedding matrix to the loss terms for local contexts, temporal contexts, and user preferences (e.g. $\lambda_{local}(t)$ becomes $\lambda_{local}(t, U)$, where U is the embedding matrix we seek to populate).

Minimizing Eq. 7 jointly for every $U(t)$ would require a large amount of memory to store all arrays. Hence, the first step is to decompose the objectives by time step, and instead solve the following equation for each $\lambda(t)$ using alternating minimization:

$$\begin{aligned}\lambda(t, U) &= \alpha\lambda_{local}(t, U) + \tau\lambda_{temporal}(t, U) \\ &+ \kappa\lambda_{user}(t, U) + \gamma\lambda_{low}(t, U)\end{aligned} \tag{16}$$

Minimizing each of these equations with gradient descent is computationally expensive. Instead, we introduce a second embedding matrix W to minimize the more relaxed problem below:

$$\begin{aligned}\lambda(t) &= \alpha\lambda_{local}(t, U) + \tau\lambda_{temporal}(t, U) \\ &+ \kappa\lambda_{user}(t, U) + \gamma\lambda_{low}(t, U) \\ &+ \alpha\lambda_{local}(t, W) + \tau\lambda_{temporal}(t, W) \\ &+ \kappa\lambda_{user}(t, W) + \gamma\lambda_{low}(t, W) \\ &+ \rho \left\| U(t) - W(t)U(t)^T \right\|_F^2\end{aligned} \tag{17}$$

Eq. 17 contains mirrored loss terms for both embedding matrices U and W. The final term ensures that U and W have identical embeddings, which can be accomplished by setting ρ to a very large value (in our case, we choose 100).

By formulating the equation in this way, which breaks the symmetry of factoring $Y(t)$, Yao et al. (2018) find that minimizing $\lambda(t)$, for both $U(t)$ and $W(t)$, is the solution of a ridge regression problem. For optimizing $U(t)$ (and equivalently, $W(t)$), taking the derivative of Eq. 17 leaves us with an equation in the form $U(t)A = B$, where A and B are defined as follows (we omit the $\frac{1}{2}$ scalar):

$$A = (1 + \kappa W(t)^T W(t)) + (\alpha + 2\tau + \gamma + \rho)I \tag{18}$$

$$\begin{aligned}B &= Y(t)W(t) + \rho W(t) \\ &+ \tau(U(t-1) + U(t+1)) + \kappa Z(t)U(t)\end{aligned} \tag{19}$$

Solving $U(t)A = B$ for every t can be accomplished efficiently by using Block Coordinate Descent (Tseng, 2001).

B Experimental Setup

B.1 Dataset Description

We provide thorough summary statistics of the Arxiv, UN, and Newspop datasets in Table 5.

All datasets (Arxiv, UN, Newspop) were obtained from publicly available sources. The original Arxiv dataset contains research papers from all scientific fields, so we select a subset of these papers by finding those which are categorized solely by "machine learning," "computer vision," or "natural language processing". The original UN dataset contains very long documents (around 4000 words), so we treat each paragraph as a document instead. The documents from the Newspop dataset were not modified.

On the UN dataset, the speaker name was present, but these speakers are public figures part of the United Nations General Assembly, and their speeches have been released to the public. Given the informative nature of each dataset, we did not find any other personal data or offensive content. To check this, we analyzed a random sample of 50 documents from each dataset. Apart from what was mentioned in the paper, we also modify the datasets by filtering noisy symbols with Regex[3] and converting all characters to ASCII with Unidecode[4]. To our knowledge, all datasets are entirely in English. We did not split any of the datasets into training, testing, or validation sets, since we did not perform any tasks which require inference and validation.

After this pre-processing, we perform phrase-chunking with AutoPhrase (Shang et al., 2018) on all datasets, treating each phrase as a single embedding, and remove phrases that appear in less than $\frac{1}{5000}$ documents. After these two steps, the vocab sizes for Arxiv, UN, and Newspop are 16073, 26184, and 8199, respectively. Models are trained on the pre-processed datasets to retrieve 5-term topic evolutions.

B.2 Model Inputs

For the Arxiv dataset, the inputs to each model were the pre-processed corpus and user-provided seeds (1) *natural language processing*, (2) *vision*, and (3) *neural network*. For the UN dataset, the inputs to each model were the pre-processed corpus and user-provided seeds (1) *disaster* and (2) *leader*.

[3] https://docs.python.org/3/library/re.html

[4] https://pypi.org/project/Unidecode/

Table 5: Detailed description of the Arxiv, UN, and Newspop datasets used in our experiments.

Dataset	#Docs	Time Range	#Time Steps	Granularity	Average #Words/Doc	Min #Docs in Time Steps	Max #Docs in Time Steps
Arxiv	214,178	2012 to 2022	11	Years	91.62	2112	44724
UN	250,997	1970 to 2014	12	4 Years	47.88	8119	45154
Newspop	93,080	Nov 2015 to Jul 2016	10	Months	24.49	273	12995

For the Newspop dataset, the inputs to each model were the pre-processed corpus and user-provided seeds (1) *technology*, *microsoft*, and (2) *politics*, *president barack obama*. We include *microsoft* and *president barack obama* as additional seeds because the documents discussing technology and politics in the Newspop dataset mostly surround these two topics.

B.3 Training Setup

We provide the Python code implementation of DynaMiTE in the supplementary material.

DynaMiTE is initialized with word2vec for faster convergence and trained with $\alpha = 100$,$\gamma = \kappa = \tau = 50$. We set $\beta = 0.2$, 0.05, 0.4 and BIDF window size $r = 5$,7,5 for Arxiv, UN, and Newspop, respectively.

In practice, we train DynaMiTE by combining Eq. 3 and Eq. 6 into a single loss term and treat each Θ_{tj} as one document. Both of these steps result in equivalent performance and help DynaMiTE run more efficiently. DynaMiTE considers local context window sizes of 7 for Arxiv and UN, and the entire text for Newspop (as headlines are short). The embedding size of DynaMiTE is set to 50. When retrieving topic evolutions for qualitative experiments, we also add a condition that any added term must not have a cosine similarity above 0.9 with any of the terms currently in the topic evolution to avoid redundancy, which is calculated through our discriminative dynamic word embeddings. As mentioned in the paper, DynaMiTE is trained entirely on CPUs and is limited to using only 10 CPUs.

We manually tuned all the above hyperparameters for DynaMiTE until a desirable set of hyperparameters was found. To evaluate topic evolution outputs with each set of parameters, we considered NPMI and also completed a qualitative evaluation by calculating MACC on a subset of the topic evolution. We prioritized the qualitative evaluation when selecting hyperparameters.

B.4 Baseline Implementations

We implement DNLDA using the official Python Georgetown DataLab Topic Modeling package[5] uploaded by the authors of the paper. We set most of the parameters to be the default values of the model. The only parameter we change is the number of topic evolutions outputted by the model, which we set to 200 to ensure that topic evolutions existed for each of our specified seeds. DNLDA was trained entirely on CPUs. To select topic evolutions, we manually search through the outputs, prioritizing those which contain any of our user-provided seeds.

We implement BERTopic using the official Python bertopic package[6] uploaded by the authors of the paper. We set all of the parameters to be the default value of the model. BERTopic was trained using multiple GPUs. We follow the same process as DNLDA to retrieve topic evolutions.

We implement Bernoulli using the Pytorch implementation[7]. We choose this one over the official implementation because it is computationally efficient. When testing both versions, we found no noticeable difference in performance, and thus elected for the Pytorch implementation. We set all parameters to be the default value of the model, with the exception of the word embedding size, which is set to 50. The Bernoulli model was trained using multiple GPUs. To select topic evolutions, we first find the embeddings of the user-provided seeds (averaging them if there are multiple seeds for a single topic evolution). Then, we find each seed's nearest neighbors for each time step using cosine similarity and retrieve these as the outputs for the topic evolution.

We implement DW2V using the official Python code[8] uploaded by the authors of the paper. We set all of the parameters to be the default value of the model and warm up DW2V with global word2vec

[5] https://github.com/GU-DataLab/gdtm

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

[7] https://github.com/llefebure/dynamic_bernoulli_embeddings

[8] https://github.com/yifan0sun/DynamicWord2Vec

embeddings. DW2V considers the same local window sizes as DynaMiTE to calculate PMI. The word embedding size is set to 50. DW2V was trained entirely on CPUs. We follow the same process as Dynamic Bernoulli Embeddings to retrieve topic evolutions.

We implement CatE using the official C code[9] uploaded by the authors of the paper. We set all of the parameters to be the default value of the model. CatE is a user-guided topic mining framework, so we did not have to retrieve terms through our own implementation. To make CatE dynamic, we run it recursively on each time-stamped document collection with the same parameters.

B.5 Quantitative Metrics

As stated in the paper, we report NPMI averaged over 25 runs. The standard error of these runs for Arxiv, UN, and Newspop were 0.0437, 0.0395, and 0.0188 respectively. We found that the outputs of DynaMiTE were consistent on most occasions. To obtain the topic evolutions for human evaluation (term accuracy and temporal ordering), we only consider a single run chosen at random.

We also report the detailed formulas for NPMI, MACC, and Rank, as well as the statistical tests we used to determine significance below:

NPMI or normalized pointwise mutual information is a standard measure of topic coherence. To calculate the NPMI for a topic evolution, we first calculate the normalized pointwise mutual information for each pair of terms at each time t, defined as follows:

$$\text{NPMI}(t) = \frac{1}{|\mathcal{C}|}\sum_{i=1}^{|\mathcal{C}|}\frac{1}{\binom{|\mathcal{S}_{ti}|}{2}}\sum_{w_j,w_k\in\mathcal{S}_i}\frac{\log\frac{P(w_j,w_k)}{P(w_j)P(w_j)}}{-\log P(w_j,w_k)}$$

$P(w_j, w_k)$ is the probability that w_j and w_k co-occur in a document, while $P(w_j)$ is the probability that w_j occurs in any document. We then calculate our NPMI metric as the sum of all NPMI(t) divided by the total number of time steps in $\mathcal{T}$. i.e.:

$$\text{NPMI} = \frac{1}{|\mathcal{T}|}\sum_{t=1}^{|\mathcal{T}|}\text{NPMI}(t)$$

We calculate the statistical significance of the NPMI values produced by each baseline with an approximate randomization test, using the list of NPMI values over 25 runs as the distribution.

MACC or mean accuracy measures term accuracy, defined as the proportion of retrieved terms that "belong" to the category name. To adapt MACC for dynamic topic mining, we flatten all terms retrieved by the dynamic topic mining frameworks and do not consider the temporal aspect. The exact formula for a single annotator is as follows:

$$\text{MACC} = \frac{1}{|\mathcal{T}||\mathcal{C}|}\sum_{t=1}^{|\mathcal{T}|}\sum_{i=1}^{|\mathcal{C}|}\frac{1}{|\mathcal{S}_{ti}|}\sum_{w_j\in\mathcal{S}_{ti}} I(w_j \in c_i)$$

I is the indicator function which denotes whether w_j belongs to category c_i, according to the annotator. We report our final results as these MACC scores averaged over all annotators.

To conduct a pairwise t-test for significance, we construct a list M for each model which contains the MACC scores for every dataset, seed, and annotator. We have 7 total seeds and 3 annotators, so M has a length of 21 for each baseline. As our sample size is small, we conduct Wilcoxon signed-rank tests using each list M.

Rank or Spearman's rank correlation coefficient is a value ranging between -1 and 1 to compare an annotator's ordering x_i and the ground truth ordering y_i for category i, where 1 is a perfect match and -1 is where the annotator's ordering is the ground truth order in reverse. We represent y_i as the list $\{t|0 < t \leq |\mathcal{T}|\}$, while the x_i will be some permutation of the ground truth order. Using x_i and y_i, Spearman's rank correlation coefficient is calculated as:

$$\frac{1}{|\mathcal{C}|}\sum_{i=1}^{|\mathcal{C}|}\left(1 - \frac{6\sum_{t=1}^{|\mathcal{T}|}(x_i(t) - y_i(t))^2}{|\mathcal{T}|(|\mathcal{T}|^2 - 1)}\right)$$

where $x_i(t)$ denotes the t-th element of list x_i. We report our final results as these Spearman's rank correlation coefficients averaged over all annotators.

Since our orderings contain a maximum of 12 elements, we cannot conduct the usual significance test for Spearman's rank correlation, as it requires at least 500 samples. Thus, we use a permutation test to compute the statistical significance[10], and mark models which obtain a significant human

[9] https://github.com/yumeng5/CatE

[10] https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.spearmanr.html

ordering (that is, a human ordering significantly close to the true ordering) for all seeds and annotators.

Conf measures the annotator's confidence during ranking, which is a discrete value from 1 to 5, based on Mean Opinion Score. The exact criteria for Conf can be viewed in Figure 5. We report the confidence values averaged over all annotators and seeds. For determining if Conf values were significant, we follow the same approach as MACC described above.

B.6 Human Experiments

We provide details on the term accuracy (Figure 4) and temporal quality (Figure 5) human evaluation experiments below:

Term Accuracy: First, we compile the topic evolutions of all baselines and ablation models of DynaMiTE (including our full version). We flatten the terms contained within each topic evolution and upload them to the tool. To avoid any positional biases, the order of terms is randomly shuffled for each annotator. Using a checkbox for each term, annotators are instructed to select terms that they believe belong to the category name, where ”belong” is defined as a non-synonym relationship between the category and term. To effectively complete the task, annotators are provided with all category names considered in the experiment, the relevant time steps, the dataset (or context) of the experiment, resources and examples for types of non-synonym relations, and a sample Google search query for ascertaining whether a term and category are related.

Temporal Quality: For each topic evolution, we remove the label that indicates which time step each set of terms belongs to. We present annotators with these terms in a randomized order, where each annotator sees a different randomized order. Annotators are instructed to order these sets of terms chronologically by using a drag-and-drop functionality integrated into the PrairieLearn interface. To effectively complete the task, annotators are provided with the dataset (or context) of the experiment, the relevant time steps, and a sample Google search query for ascertaining whether a set of terms precedes or succeeds another set of terms. After annotators have completed ordering the terms they are asked to rate their confidence on a scale of 1 to 5 based on Mean Opinion Score (Streijl et al., 2016), using a multiple choice question.

Both tools displayed in the Figures were created using the PrairieLearn (West et al., 2015) interface, which is traditionally used in classroom settings. Annotators can submit their results at any time by pressing “Save and Grade”. By pressing “Save,” annotators can save their current results and choose to come back to the experiment at a later time. We find that PrairieLearn's easy-to-use interface and integration of Python make it an ideal tool for setting up human evaluation experiments. We received no complaints from our annotators indicating that PrairieLearn was a difficult tool to navigate. We hope to work with the creators of PrairieLearn to make it publicly available for all types of human evaluations.

C Full Experiment Results

C.1 Topic Evolutions

We display the full 5-term topic evolution outputs produced by DynaMiTE on the Arxiv (Table 6), UN (Table 7), and Newspop (Table 8) datasets.

C.2 Category Shift Analysis

We display all category shift analyses on the seeds and datasets from our experiments in Table 9.

Figure 4: Screenshot from the human evaluation experiment for Term Accuracy (MACC).

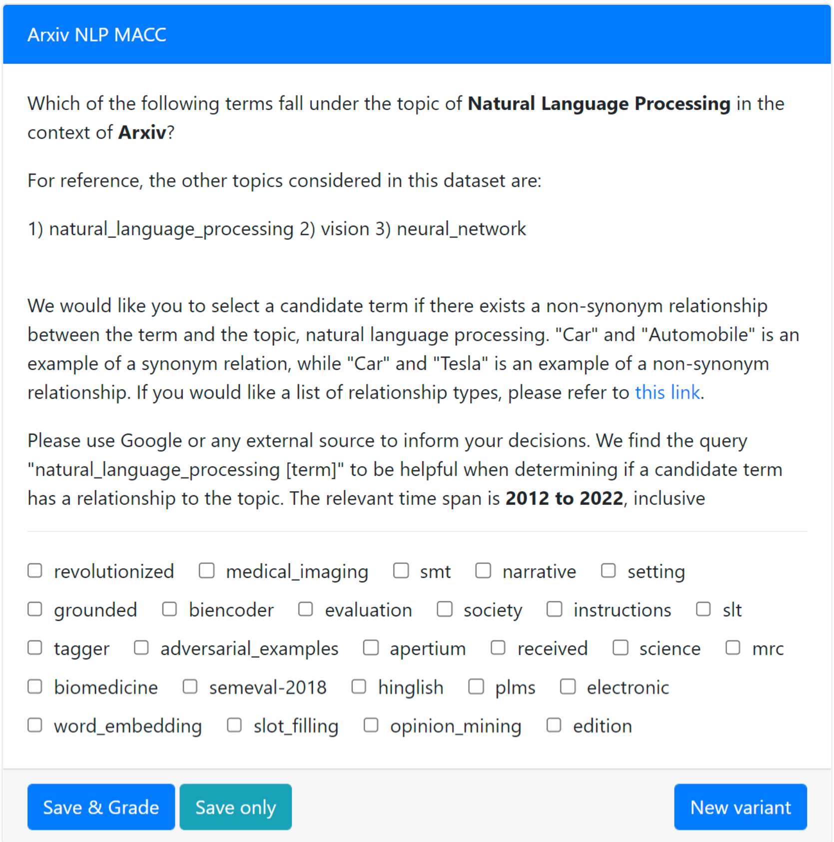

Figure 5: Screenshot from the human evaluation experiment for Temporal Ordering (Rank and Conf).

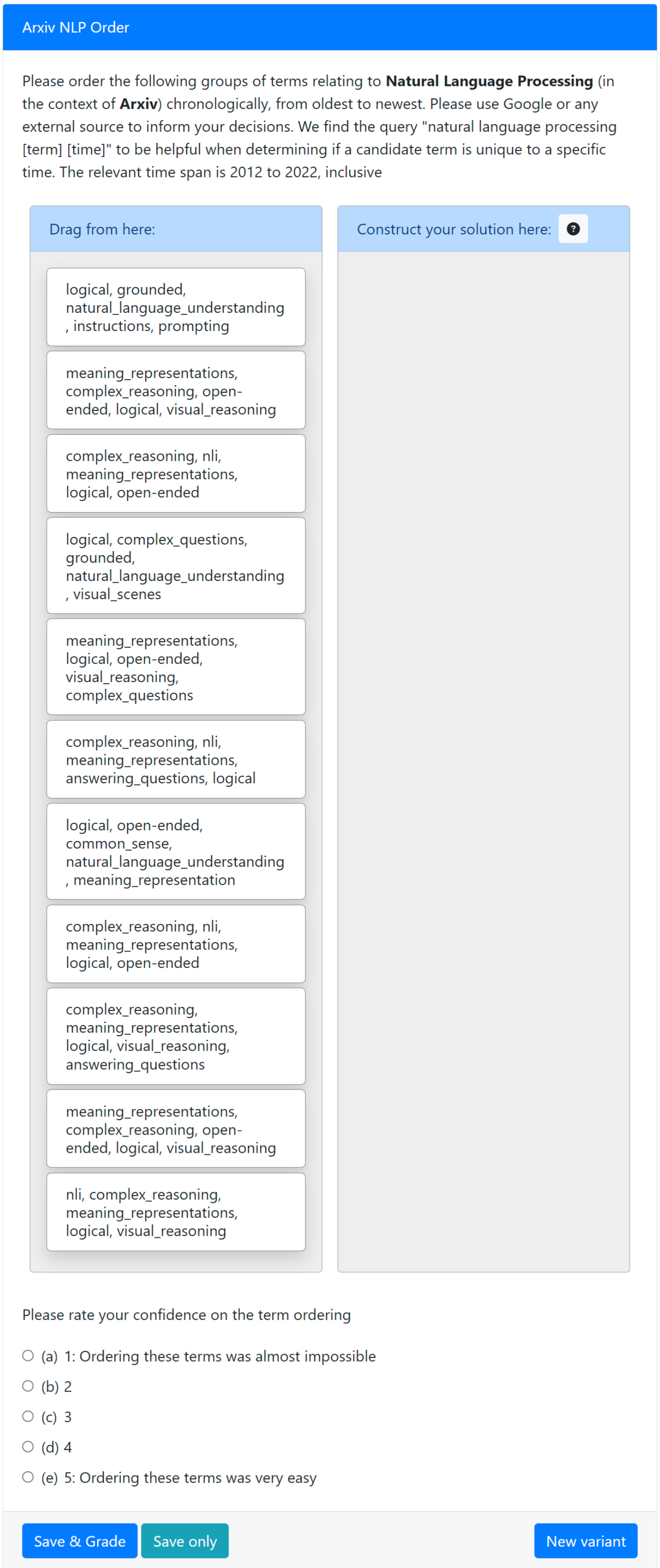

Table 6: Full DynaMiTE topic evolution output on the Arxiv dataset.

Time	*Natural language processing*	*Computer vision*	*Neural networks*
2012	sentiment classification	walking	pc
	linguists	vb	network structures
	successes	social interaction	regularization methods
	society	machine vision	feed forward
	social science	milestone	amino acids
2013	fsl	visual object tracking	tnn
	speech recognition	sports	neuron
	mt	ultimately	multiplicative noise
	inflection	scene recognition	cnn
	urdu	sparked	rectifier
2014	biomedicine	synthesis	arrhythmia
	statistical machine translation	supervisions	auto-encoder
	srl	silhouettes	cae
	prosody	theories	dae
	zero-shot	synthetically generated	dropout
2015	automatic speech recognition	event recognition	relu
	iwslt	kinship	feed-forward neural network
	word embeddings	pedestrian detection	anns
	slt	pedestrian	deep nets
	relation classification	railway	lstm
2016	patent	re-id	siamese
	speech recognition	ssc	nmt
	neural architectures	scene parsing	yolo
	relation classification	scene text detection	lstm
	image captioning	instance segmentation	recurrent network
2017	stance detection	sonar	gru
	nli	lipreading	over-parameterized
	sts	material recognition	pointnet
	prosody	scene flow estimation	smiles
	slot filling	scene segmentation	tensorflow
2018	sanskrit	scene graph	i3d
	roman	vehicle re-identification	bnn
	sentence encoders	sod	approximators
	contextualized word representations	lane detection	tnn
	code-mixing	object counting	qnn
2019	pretrained language models	tir	neural tangent kernel
	contextual embeddings	vos	bnn
	multilingual bert	thermal infrared	loss landscape
	roberta	str	pinn
	bert	rec	infinite-width
2020	pretrained language models	attracted considerable attention	neural tangent kernel
	multilingual bert	pansharpening	infinite-width
	mlm	qml	neural ordinary differential equations
	contextual embeddings	shadow removal	pinn
	xlm-r	rec	double descent
2021	plm	sonar	ntk
	xlm-roberta	shadow removal	infinite-width
	mbert	vl	qnn
	qe	rgbt tracking	neural ode
	gpt-3	hpe	pinn
2022	pretrained language models	vl	neural ordinary differential equations
	gpt-3	vision transformers	infinite-width
	mbert	wsol	mpnns
	xlm-r	rec	benign overfitting
	qe	video instance segmentation	symplectic

Table 7: Full DynaMiTE topic evolution output on the UN dataset.

Time	*Disaster*	*Leader*
1970 - 1971	east pakistan pakistanis physical environment bengal economic losses	allende gamal abdel nasser figueres gaulle cabral
1974 - 1977	desertification emergency situation energy crisis sahelian countries fourth world	chairman mao tsetung makarios houari boumediene archbishop
1978 - 1981	dominica grenada grenadines saint lucia saint vincent	agostinho neto robert mugabe mwalimu julius nyerere houari boumediene guzman
1982 - 1985	cilss devastating impact cyclical fragile economy com	jorge roberto jose figueiredo belaunde
1986 - 1989	chernobyl locusts hurricane hugo nuclear accident bengal	mr gorbachev shultz president reagan president bush mikhail gorbachev
1990 - 1993	chernobyl devastating earthquake iraqi invasion of kuwait herzegovina bosnia	npfl mr nelson mandela klerk non-racial african national congress
1994 - 1997	montserrat hurricane luis igadd monitoring group ecomog sarajevo	mahmoud npfl ulimo kofi annan mr boutros boutros-ghali
1998 - 2001	hurricane georges el nino pennsylvania financial crises humanitarian catastrophes	kabila secretary-general kofi annan predecessor mr hennadiy udovenko predecessor mr harri holkeri predecessor mr harri
2002 - 2005	hurricane katrina tsunami hurricane ivan locusts pennsylvania	mr sergio vieira de mello mahmoud abbas lula da silva tony blair kabila
2006 - 2009	locusts global financial crisis coastal erosion glaciers degrees celsius	zelaya morazan president obama sarkozy lula
2010 - 2013	global financial crisis darfur syrian refugees devastating earthquake eurozone	secretary-general ban ki-moon reappointment mr nassir abdulaziz al-nasser predecessor mr joseph deiss mr vuk jeremi
2014 - 2017	ebola virus existential threat existential disaster risk reduction ocean acidification	president obama pope francis rouhani leon saleh

Table 8: Full DynaMiTE topic evolution output on the Newspop dataset.

Time	*Technology, Microsoft*	*Politics, President Barack Obama*
October 2015	sql server zune steve ballmer surpassed sunrise	plea mocking rallies pro-palestine plan
November 2015	partnership using via xl volvo	obama white house thanksgiving syrian refugees republican
December 2015	nasdaq using windows 10 mobile operating giant	obama white house oval office terrorism sunday night
January 2016	minecraftedu web browser word flow keyboard cellular data ces	mosque baltimore solitary confinement religious freedom juveniles
February 2016	swiftkey xamarin underwater keyboards mid-range	mosque muslim-americans supreme court justice antonin scalia national prayer breakfast ray charles
March 2016	networking xamarin uwp hololens augmented reality gdc	nancy reagan state dinner nuclear security summit state visit tango
April 2016	word flow keyboard regulatory complaints dna financial results female employees	nuclear weapons nuclear security summit university of chicago law school roberta hanover germany
May 2016	solair iot sap xiaomi sharepoint	white house correspondents dinner rutgers university howard university commencement address commencement speech
June 2016	xiaomi social network kind financial cannabis 26.2	muhammad ali respects victims orlando nightclub
July 2016	worldwide partner yusuf mehdi project scorpio combine all-in-one	warsaw praising presumptive presumptive democratic presidential nominee hillary clinton forceful

Table 9: Category shift analysis (§6.4) on all seeds and datasets used in the experiments.

Dataset	**Category Name**	**Largest Shift**	**Term Causing Shift**
Arxiv	natural language processing computer vision neural networks	2021 to 2022 2012 to 2013 2013 to 2014	gpt-3 visual object tracking auto-encoder
UN	disaster leader	1986 - 1989 to 1990 - 1993 1990 - 1993 to 1994 - 1997	chernobyl npfl
Newspop	technology politics	January 2016 to February 2016 December 2015 to January 2016	underwater solitary confinement