Topic-Noise Models: Modeling Topic and Noise Distributions in Social Media Post Collections

Rob Churchill
Georgetown University
Washington, D.C. USA
rjc111@georgetown.edu

Lisa Singh
Georgetown University
Washington, D.C. USA
lisa.singh@georgetown.edu

***Abstract*—Most topic models define a document as a mixture of topics and each topic as a mixture of words. Generally, the difference in generative topic models is how these mixtures of topics are generated. We propose looking at topic models in a new way, as topic-noise models. Our topic-noise model defines a document as a mixture of topics and noise. Topic Noise Discriminator (TND) estimates both the topic and noise distributions using not only the relationships between words in documents, but also the linguistic relationships found using word embeddings. This type of model is important for short, sparse social media posts that contain both random and non-random noise. We also understand that topic quality is subjective and that researchers may have preferences. Therefore, we propose a variant of our model that combines the pre-trained noise distribution from TND in an ensemble with any generative topic model to filter noise words and produce more coherent and diverse topic sets. We present this approach using Latent Dirichlet Allocation (LDA) and show that it is effective for maintaining high quality LDA topics while removing noise within them. Finally, we show the value of using a context-specific noise list generated from TND to remove noise statically, after topics have been generated by any topic model, including non-generative ones. We demonstrate the effectiveness of all three of these approaches that explicitly model context-specific noise in document collections.**

***Index Terms*—generative topic modeling, topic noise model**

I. Introduction

Researchers trying to understand information shared through social media need tools that can be used to quickly make sense of these large volumes of data. One well known technique for understanding conversation is topic modeling. Unfortunately, identifying high quality topics is more challenging than ever. Generative topic models in particular rely on repetition of word pairs within the same document in order to form meaningful topics. In shorter social media posts, noise words infiltrate topic-word distributions with ease, cluttering topics, and degrading the overall quality of topic models.

These problems are only intensified in domain-specific social media data sets, which, in addition to traditional noise words, i.e. stopwords, also have context-specific noise words, including flood words [1], [2]. Flood words, such as ‘covid,’ ‘coronavirus,’ and ‘pandemic,’ in a data set specifically about the 2020 Covid-19 pandemic, appear so frequently in documents that they dominate all topics in a topic set, making it difficult to discern different topics from each other. Given the prevalence of flood words and non-random noise in social media text, we believe that noise cannot be ignored, and must instead be understood. In this paper, we accept that documents are composed of both topics and context-specific noise, and that both need to be modeled in order to accurately identify topics. Further, the size of the vocabulary and the shortness of posts also require us to reconsider the role of newer linguistics techniques for distinguishing topics from noise. Finally, given that the ‘best’ topics can be subjective, having the ability to use a constructed noise distribution with other generative topic models is also important for noisy domains.

Given these considerations, we propose the development of a new class of models, *topic-noise models*. Topic-noise models define a document as a mixture of topics and noise. Specifically, we propose **Topic Noise Discriminator (TND)**, a topic-noise model that estimates both the topic and noise distributions, thereby understanding both the contextual topics and contextual noise in a social media document collection. **TND** has the following properties: 1) it assumes that topic words and noise words can have similar frequencies and therefore need to be explicitly modeled in order to generate topics that are more coherent and contain small amounts of noise, 2) it adjusts the generative model to incorporate additional knowledge from embedding spaces when modeling both the topic and noise distributions in order to elevate the importance of contextually similar words, and 3) it produces a reusable noise distribution that can be integrated into existing generative models favored by certain research communities. While some previous work has considered modeling special word or background distributions [3], [4], our proposed generative process captures context-specific noise and topics extended by semantic insight from word embeddings. We believe that generating topics AND noise distributions on data is fundamentally a new way to think about topic modeling and will be foundational for a new generation of topic-noise models.

The contributions of this paper are as follows. 1) We propose a new generative topic-noise model (TND) that explicitly models both topic and noise distributions and adjusts the generative model to incorporate additional knowledge from embedding spaces. 2) We propose a variant of our model that combines a pre-trained noise distribution from TND in an ensemble with any generative model as a way for any existing topic model to filter noise words and produce more coherent and diverse topic sets. We show an example of this with LDA, and demonstrate its value by showing that NLDA, an integration of TND's noise distribution in an ensemble with LDA to filter noise words, produces more coherent topics than LDA. 3) We show the value of using a context-specific noise list generated from TND to remove noise in an ad hoc fashion to improve the quality of topic sets produced by other topic models, including non-generative ones. 4) We conduct an extensive empirical analysis using two large Twitter data sets (Covid-19 and an Election 2020), and the 20 Newsgroups data set and show the strength of explicitly modeling noise and using embeddings during the topic-noise modeling process. 5) We publish our model code and other methods used in our experiments, along with our evaluation metrics.[1]

The paper is organized as follows: Section II presents the related literature. Section III defines terminology used throughout the paper. Section IV presents our models. Section V contains quantitative and qualitative analyses of our models. Conclusions are presented in Section VI.

II. Related Literature

The most prevalent type of unsupervised topic model is the generative model, which is the basis of the topic-noise model we propose. Generative topic models rely on the key assumption that documents are generated according to a known distribution of terms. The most widely used of the generative class is Latent Dirichlet Allocation (LDA) [5], which inspires the vast majority of other generative models. LDA uses a bag-of-words model to find the parameters of the topic/term distribution that maximize the likelihood of documents in the data set over k topics. Among its direct descendants are Hierarchical Dirichlet Process (HDP) [6], Dynamic Topic Models (DTM) [7], Correlated Topic Models (CTM) [8], and Twitter-LDA [9]. Each of these iterations attempts to leverage the key assumption in a different manner to improve upon LDA. However, all of them use a single distribution to compute topics and ignore modeling noise.

There are a few examples of generative topic models that attempt to incorporate multiple distributions within the generative process. Chemudugunta et al. [3] propose a special words topic model with a background distribution (SWB) to model different aspects of documents. Based on LDA, the approach of Chemudugunta et al. differs by incorporating word distributions (special word and background distributions) adjacent to the traditional topic-word distribution. While our approach has similarities (we will detail the differences in Section IV), there are two main differences, our modeling of noise differs from their special word and background distributions, and our models use word embeddings to better model topics and noise.

A newer direction of topic modeling research looks at incorporating more sophisticated NLP techniques and mixture models into generative models. Embedding-based Topic Model (ETM) [10] uses word embeddings to aggregate short texts into long pseudo-texts, and then infers topics from the pseudo-texts. Yan et al. perform topic modeling on pairs of terms with high co-occurrence in their Biterm Topics model [11]. Wang et. al use LDA to get topic embeddings, and then use these embeddings along with pre-trained word embeddings to find topics in short texts [12]. Dieng et. al propose a generative model similar to LDA in essence, but which draws topic words directly from the embedding space [13]. While all these models use new NLP techniques, they do not explicitly model a noise distribution.

Another type of generative model employs the Dirichlet Multinomial Mixture (DMM), which differs from LDA in that it assumes each document has only one topic [14]. DMM has been a key building block to many topic models that attempt to better model data sets containing short documents [15], [16], [4], [17]. SATM [15] runs LDA on the larger documents that are combinations of shorter text to get the overarching topics, and uses DMM to infer the specific topic of each of the short texts. GSDMM [16] attempts to cluster documents into k topics in a round-robin approach, allowing documents to decide which topic to join by which other documents are most similar to it. Li et al. [4] use word embeddings in their sampling algorithm, sampling the related words of an observed word, to produce more coherent topics (GPUDMM). Again, none of these models incorporate any notion of a noise distribution.

Li et al. [18] deal with filtering noise from topics with their topic model, CSTM. The authors base their model on DMM [14], and incorporate two types of topics to try to capture noise and content words. The authors generate a document from a single 'functional' topic (traditional topic), as well as from a number of shared 'common' topics, which are used to aggregate noise words from all documents. Instead of a background distribution like that of SWB, the authors use topics to capture noise. This approach is similar in goal to ours, but identifies noise using a 'common' topics distribution that does not work well in a setting containing such large amounts of context-specific noise (as we will show in our empirical analysis). It also does not use word embeddings to incorporate additional context.

Finally, there are a number of approaches to topic modeling that do not incorporate generative models [19], [20], [21], [2], [1], [22], [23]. Because generative models are the standard for topic modeling and our focus is on extending generative models, our evaluation will compare the models we propose to LDA, DMM, GPUDMM, and CSTM. LDA is the most widely used generative model. DMM is a strong generative model designed for short texts. GPUDMM incorporates word embedding vectors. Finally, CSTM attempts to explicitly adjust for noise within the generative process.

III. Background & Notation

Let D represent a *data set* consisting of M documents or posts, where $D = \{d_0, d_1, ..., d_{M-1}\}$. A *document* d is a collection of N words, where $d = \{w_0, w_1, ..., w_{N-1}\}$. A

[1] The code repository can be found here: https://github.com/GU-DataLab/topic-modeling-tnd

topic t consists of a set of ℓ words, $t = \{w_0, w_1, ..w_{\ell-1}\}$, where the words in t are coherent and interpretable. A topic set T consists of k topics, where $T = \{t_0, t_1, ..t_{k-1}\}$. A noise set H consists of a set of p words, $H = \{w_0, w_1, ..w_p\}$, where the words in H represent noise.

Our central claim is that topic models must not ignore noise when the data set contains social media posts. From a quantitative perspective, high quality topics are coherent, interpretable, and contain little noise. High quality topic sets are diverse, i.e. more unique as opposed to similar. Noise in social media posts comes in different forms. We can divide these different types of noise into two broad categories, context-free noise and context-specific noise.

Context-free noise words are defined as words that are considered content-poor irrespective of the domain of the data. Stopwords are an example of context-free noise. Because stopwords are data set agnostic, they are known prior to the execution of a model and can be easily pruned from a data set. *Context-specific* noise words are noise within the context of the data set. Some context-specific noise words are not meaningful within the domain, but happen to occur more often than expected. We refer to these noise words as *generic noise words*. Examples of generic noise words in a data set about the 2020 Covid-19 Pandemic would include words like 'today,' 'made,' 'think,' and 'said.' These words do not add to the understanding of a topic about Covid-19. Another form of context-specific noise is flood words. Flood words are domain specific words that appear frequently and are highly relevant to the domain. However, they are relevant to a large number of topics and therefore, cannot be used to help distinguish topics. Examples of flood words in a data set about the 2020 Covid-19 Pandemic would be 'covid' and 'pandemic.' In this paper, H represents context-specific noise, both generic noise words and flood words.

Our focus, from a quantitative perspective, is to improve the coherence and diversity of topics within topic-noise models, generate a noise distribution that contains different types of noise, and reduce the amount of noise present in topics. We define *topic coherence* as the ability of a topic model to detect meaningful and interpretable topics in a data set. We define *topic diversity* as the ability of a topic model to detect unique topics in a data set (as opposed to a set of very similar topics). Together, topic coherence and diversity represent a model's ability to detect a range of topics that can be easily understood. We define *noise penetration* as the ability (or lack thereof) of a topic model to filter noise from its topic set. A high noise penetration level reflects poorly on a topic model's ability to detect words that strongly represent topics. We detail our exact computations of each of these metrics in Section V.

In summary, our goal is the following. Given a data set D, can we produce a topic set T that is coherent and diverse, and a noise set H that captures context-specific noise?

IV. Approach

In this section, we describe our proposed models in detail. In order to relate our models to the most relevant in the previous literature, we begin by presenting the plate notation and describing LDA [5] and (SWB) [3] (Section IV-A). We then describe our proposed topic-noise model (TND) (Section IV-B), and the extension using embedding sampling (Section IV-C). Finally, we describe our approach for combining existing generative and non-generative models with the noise distribution generated by TND (Section IV-D).

A. LDA and SWB Topic Models

Figure 1 shows the graphical representations of LDA (a) and SWB (b). While the entire generative process for LDA is presented by Blei et. al [5], we present the high-level generative process in our notation here.

For $d \in D$:

1) Draw the number of words N for d.
2) Draw the topic distribution θ from the Dirichlet distribution, conditioned on the parameter α.
3) For each word w_i, $0 \leq i < N$:
    a) Draw a topic z_i from θ.
    b) Draw a word w_i based on the probability of w_i given the topic z_i and conditioned on the parameter β.

The special words topic model with a background distribution (SWB), proposed by Chemudugunta et al. [3], improves on LDA by adding a special words distribution for each document, and a global background distribution. SWB's generative process works similarly to that of LDA, but with some important changes to account for its extra distributions. First, a word is not guaranteed to be drawn directly from the document's topic distribution. Instead, it can be drawn from the document's topic distribution, from the document's special words distribution (Ψ in Figure 1(b)), or from the independently computed global background distribution (Ω in Figure 1(b)). The decision of which distribution to draw from is controlled by x, which is sampled from a document-specific multinomial λ conditioned on γ.

B. Topic-Noise Discriminator (TND)

Recall that we define a document as a mixture of topics and noise. Therefore, our generative model, Topic-Noise Discriminator (TND) alters the generative process of the topic distribution to account for an underlying noise distribution. The graphical model for TND is shown in Figure 1(c). We identify noise by approximating the distribution of noise words across the document collection D. Intuitively, instead of each word in the document being drawn from the document's topic distribution (as in LDA), each word is drawn from *either* that document's topic distribution, *or* a global noise distribution, based on the probability of the individual word being in a topic or in the set of noise words. While this looks similar to the special word distribution in SWB, it is designed differently. SWB is designed to capture words that appear in a specific document and rarely anywhere else. The underlying assumption here is that these special words appear frequently in their respective documents, such as the word 'Hogwarts'

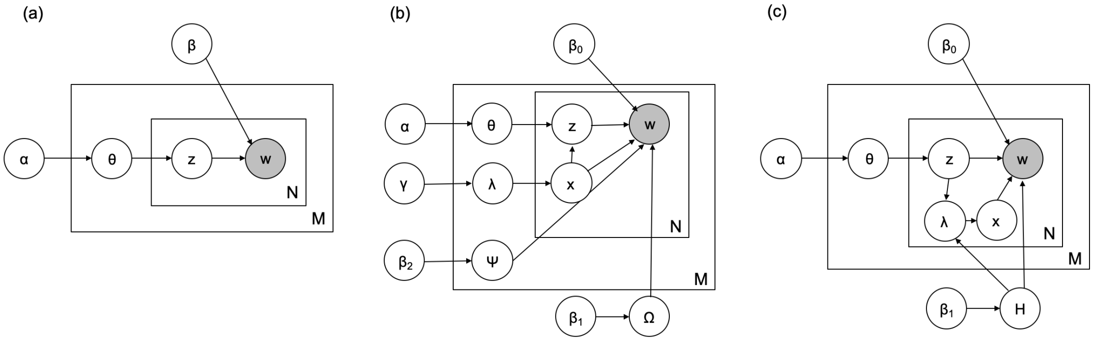


Fig. 1: LDA (a), SWB (b), and TND (c) Graphical Models.

would appear an irregularly high number of times in a Harry Potter book, and almost never in other contexts.

In social media data, documents are so small that with high certainty, words will not appear frequently enough in a single document for them to affect the composition of an entire topic, and any word that appears in a single document will be removed by reasonable preprocessing (such as removing words that appear only once in the data set). Therefore, the special words distributions are not needed for a topic model that is intended for social media data because that distribution cannot capture the 'right' words, thereby unnecessarily complicating a model designed for short posts. The background distribution is closer to how we model the noise distribution. However, the SWB background distribution is computed independently. In contrast, our noise distribution is not.

The decision of whether a word is a topic word or a noise word is determined using the Beta distribution (see Figure 1). The Beta distribution, λ, is the special case of the Dirichlet where k=2, and x is the switching variable controlling whether the word is drawn from z or H. This distribution is conditioned on the β_1 parameter. Setting the initial value of β_1 higher allows us to skew the distribution to favor topics if the expectation of noise is less than topics. In practice, using the Beta distribution helps produces topics that contain far less noise than traditional generative models such as LDA. Equation 1 shows the calculation of the Beta distribution for each word. The Beta distribution takes into account the topic frequency and noise frequency of the given word. Using the square root of the word's frequency in the topic and noise distributions reduces the likelihood of a word continually moving between topics and noise. The effect of this alteration in the generative process is that over many iterations, noise words slowly start to affect document-topic assignment less and less.

$$Beta(\sqrt{\theta_t^i + \beta_1}, \sqrt{H_i}) \tag{1}$$

The noise distribution is not a static list, like stopwords, nor is it a strictly frequency-related list like TF-IDF rankings. Instead, the noise distribution is generated with respect to a set of topics simultaneously being generated on the data set. As such, the noise distribution has knowledge of topic words baked into it, as opposed to approaches that attempt to identify noise words without approximating a topic-word distribution.

The generative process for TND is as follows.

For $d \in D$:

1) Draw the number of words N for d.
2) Draw the topic distribution θ from the Dirichlet distribution, conditioned on α.
3) For each word w_i, $0 \leq i < N$:
   a) Draw a topic z_i from the topic distribution θ.
   b) Draw a word from either z_i or the noise distribution H, according to the Beta distribution, conditioned on α.
   c) If drawing from z_i, draw w_i based on the probability of w_i given the topic z_i and conditioned on β_0
   d) If drawing from H, draw w_i according to the probability of w_i given H and conditioned on β_1.

C. Embedding Sampling

With recent advances in natural language processing, we propose using word embedding vectors to increase the probability of semantically related words appearing together in specific topics and in the noise distribution. GPUDMM, proposed by Li et al. [4], uses word embeddings in a similar fashion, altering the traditional Gibbs sampling algorithm so that whenever a word is sampled, words related to it in the given embedding space are also sampled. In Gibbs sampling, one word is sampled at a time. In Generalized Polya Urn (GPU) embedding sampling, the word is returned with other similar words. This increases the likelihood of related words being in the same topic.

This is a clever way of producing more coherent topics, but in social media, this also allows for noise words to pull even more noise words into topics. However, using this same sampling scheme within TND, where noise words are modeled in their own distribution, we should see noise words pulling more noise words into the noise distribution instead.

To ensure that we do not pull the wrong words into the wrong distributions, we wait τ iterations to begin GPU embedding sampling. After τ iterations, and every τ iterations thereafter, we re-evaluate the words eligible for GPU embedding sampling. Only words whose probability of being in the noise distribution or of being in a single topic is higher than ν standard deviations from the average are considered. By narrowing words down this way, we ensure that we do not pull the related words of low-probability words into topics.

Our sampling approach allows for the scaling of the impact of embeddings on TND. By setting the parameter $\mu \geq 0$, we can decide how many related words to sample for each word in GPU embedding sampling. Setting $\mu = 0$ is equivalent to traditional Gibbs sampling, while increasing μ means more and more impact of embeddings on the model.

D. Extending Existing Topic Models with TND

The noise distribution generated by TND can be integrated into any topic model that produces a topic-word distribution, as generative models do. By comparing a word's probability in a topic and in noise, noise can be efficiently filtered from a topic set, leaving more coherent, interpretable topics with little overhead. We show this approach here, combining TND and LDA to create NLDA.

1) Noiseless LDA (NLDA): While TND produces topics, it also provides a useful noise distribution that can be easily transferred to other topic models. In the case where we have a pre-trained topic model that uses a topic-word distribution to approximate topics, we can apply the pre-trained noise distribution from TND in an ensemble to probabilistically remove noise words in a similar manner to the process within TND. In Noiseless LDA (NLDA), we borrow the noise distribution generated by TND, and use it with LDA, thereby creating a version of LDA that contains topics with fewer noise words.

To create NLDA, we train a noise distribution H on D using TND, and we train an LDA model on D.[2] We then produce a topic set by combining the noise distribution of TND and the topic-word distribution of LDA. Similar to deciding whether a word is a topic or noise word, for each topic $t \in T$, we remove w_i from t according to a Beta distribution (Equation 2) conditioned on w_i's frequency in noise and in LDA's topic distribution.

In order to make noise distributions more transferable to different parameters of LDA, we add a topic weight parameter ϕ to the Beta distribution calculation to downsample or oversample the noise distribution. Equation 2 shows how ϕ is used to scale the noise distribution based on k, the number of topics in the LDA model.

$$Beta\left(\sqrt{\theta_t^i + \beta_1}, \sqrt{H_i(\phi/k)}\right) \qquad (2)$$

For each word w_i in topic t, once we have determined its status using the Beta distribution, we take one final step to facilitate better topic filtering. If w_i is removed from t, w_i's frequency in the noise distribution is incremented, marking it as noise once again. If w_i is retained in t, w_i's frequency in the noise distribution is increased by θ_t^i. By increasing w_i's noise frequency *after* it is included in a topic and maintaining the topic frequency, we are deterring its inclusion in future topics, which share the noise distribution. In this way, through the Beta distribution (Equation 2), we have increased the relative probability of future topics determining it to be noise.

Decreasing ϕ to a value lower than k ($\phi < k$) will result in a lower beta value, and therefore less harsh noise filtering, while increasing ϕ to a value greater than k ($\phi > k$) will result in a higher beta value, and harsher noise filtering. Setting $\phi = k$ results in an unweighted NLDA. The addition of ϕ allows for NLDA to be scaled to larger data sets and different values of k using the same original noise distribution. While this will be unnecessary for many use cases, the ability to essentially transfer a noise distribution to different parameter settings makes NLDA more usable and faster. It also requires less storage during model construction.

2) Context Noise List Usage: Not all topic models produce topic-word distributions, and often we have access to only a set of topics that we would like to filter noise from. In the case where we have a pre-trained topic model that does not use a topic-word distribution to approximate topics, or in the case where we have only a set of topics, we can apply the TND noise distribution in a more crude manner, using a context-specific noise list. In this approach, which we call Context Noise List Usage, we propose filtering words from a topic set that have a high probability in the noise distribution. For a given noise distribution H, we define H_c to be the set of c words in the noise distribution with the highest probabilities. For each topic $t \in T$, we remove word w_i from t if $w_i \in H_c$.

This approach is more likely to remove flood words than lower-frequency noise words, but it can still be beneficial to topic sets. We will demonstrate this in the next section.

V. Empirical Evaluation

In this section, we present our empirical evaluation. We evaluate the three variants proposed in Section IV: Topic Noise Discriminator (TND), Noiseless LDA (NLDA), and Context Noise List Usage for existing models. We begin by describing our experimental setup, including a description of the data sets, the preprocessing, and the model parameters (Section V-A). We then present our quantitative evaluation (Section V-B), followed by our qualitative analysis (Section V-C).

A. Experiment Setup

Baseline Algorithms. We compare our proposed models to the following state of the art models: Latent Dirichlet Allocation (LDA),[3] Gibbs Sampling Dirichlet Multinomial Mixture (DMM) [16], Generalized Polya Urn Dirichlet Multinomial Mixture (GPUDMM) [4], and Common Semantics Topic Model (CSTM) [18]. These topic models each represent a unique facet of generative topic models as explained in Section II. As mentioned in the previous section, because SWB is designed with fewer longer documents in mind, the computation cost is too high for large volumes of social media posts and the special words distribution is not meaningful for the short post environment.

[2] The k value does not have to be the same for the two models.

[3] Specifically the MALLET implementation of LDA [24]

Data Sets. In this analysis, we consider four data sets: a newsgroup data set, two Covid-19 data sets, and an election 2020 data set. Our first data set is a subset of the Twenty Newsgroups data set [25]. We use the training set, containing 11,024 documents, to assess how well the different models generate topics that map to the labeled data. While 20 Newsgroups is a relatively small data set, it provides a platform for reproducibility and allows us to see the impact of our algorithm on a data set that contains less noise than traditional social media data sets.

We also have two Twitter data sets. The first data set contains posts about the 2020 Covid-19 pandemic. Using Covid-19 related hashtags, we collected Covid-19 related tweets through the Twitter Streaming API. For this analysis, we consider two samples of these data. The *50k Covid-19* data is a random sample of 50,000 tweets about the 2020 Covid-19 pandemic, collected between mid-January and April 2020, a time period of massive change in the conversations revolving around the pandemic. The *Million Covid-19* data contains over 1 million tweets about the 2020 Covid-19 pandemic, collected between August 1 and September 30.

The other Twitter data set, *Election 2020*, contains posts about the 2020 United States Presidential election. Using relevant hashtags and keywords, we collected these data between January 1 and September 30 through the Twitter Streaming API. This data set consists of over 1.4 million tweets, focusing on topics related to the November election. Both the Million Covid-19 and Election 2020 data sets can be used to test the ability of the different models to produce high-quality topics on larger, noisier social media data sets.

Data Preprocessing. Data preprocessing can have a significant impact on topic models [26]. For each of our Twitter data sets, we remove deleted posts and remove user tags. For all of our data sets, we lowercase text and remove urls, punctuation (including hashtags), and stopwords.

Model Parameters. In order to provide a thorough sensitivity analysis for each of our models, we test each model with many different parameter settings.[4] Because of space limitations, we only present the results for the best performing models. For TND, the best parameters for producing its own topic set were $\alpha = 0.1$, $\beta_0 = 0.01$, $\beta_1 = 25$, $k = 30$, $\mu = 0$, and $\nu = 1.5$. However, the best noise distributions for use in NLDA occurred when $\mu > 0$. For NLDA, the best performing parameters are $\alpha = 0.1$, $\beta_0 = 0.01$, $\beta_1 = 25$, and $k = 30$. As we will see, the best parameter for μ and ϕ varied based on the data set. We found that β_0, α, and β_1 were far more stable parameters and that changes in their values did not have significant effects on the performance across data sets. μ and ϕ cause more noticeable effects on performance based on the data set. In the case of ϕ, tuning is quick in practice because it applies to the ensembling of TND and LDA, where values of ϕ can be quickly iterated through on the trained models. For LDA, they were $\alpha = 0.1$ and $\beta = 0.01$. For DMM and GPUDMM, we found $\alpha = 0.1$, $\beta = 0.1$ to be the best parameters. For GPUDMM, we used GloVe Twitter word embeddings [27] with both 50 and 100 dimensions, and found the difference in topic quality to be negligible. The results shown here use 50 dimensions. For CSTM, we used the suggested settings for nu_f and nu_c, 1 and 0.1, respectively. We found $\alpha = 0.1$ and $\beta = 0.01$ with 2 common topics to be the best parameters. While the other settings tested did reduce the quality of the topics obtained, their results were similar.

B. Quantitative Analysis

In this section, we use topic coherence and topic diversity to compare the different topics generated from either topic models or topic-noise models.

1) Evaluation Metrics: To assess a model's ability to detect coherent, meaningful topics, we use normalized pointwise mutual information (NPMI) [28]. NPMI is a distance measure that captures how closely related two words are given their relative cofrequency. Many recent topic modeling papers, including that of GPUDMM [4], have employed NPMI or one of its variants to assess the coherence of their models [13], [29], [10], [15]. For a pair of tokens (x, y), we define the probability of them appearing together in a document as $P(x, y)$. We use this probability to compute the NPMI of a topic $t \in T$ as follows:

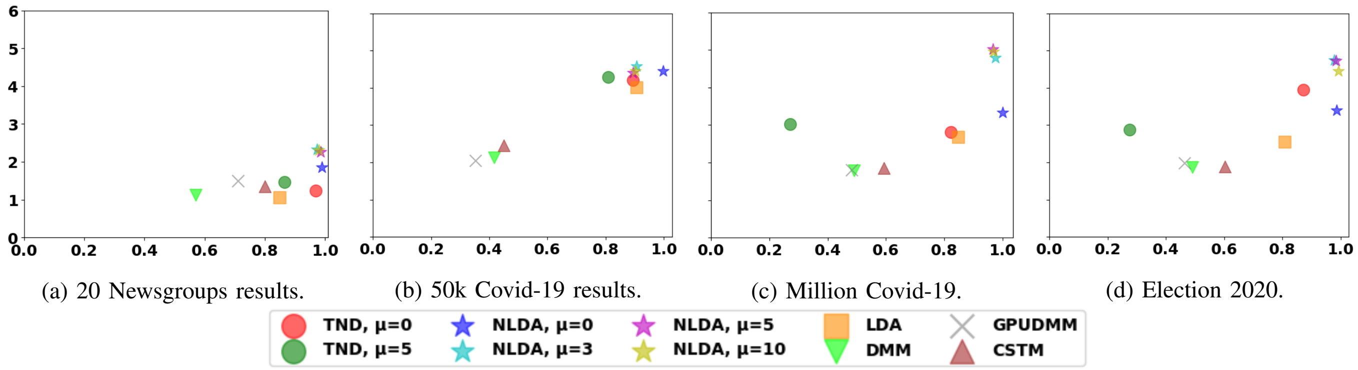
$$NPMI(t) = \frac{\sum_{x,y \in t} \frac{\log(\frac{P(x,y)}{P(x)P(y)})}{-\log(P(x,y))}}{\binom{|t|}{2}}$$

The higher the NPMI score, the higher the mutual information between pairs of words in the topic. This indicates high topic coherence, which in turn reflects on the ability of the model to detect meaningful topics.

In addition to assessing the meaningfulness of topics, we are interested in a model's ability to find distinct topics. A model that finds the same coherent topic ten times, but does not find other topics should not be considered as effective as a model that finds many unique topics that may be slightly less coherent. We measure this using topic diversity. Topic diversity is the fraction of unique words in the top 20 words of all topics in a topic set [29]. High topic diversity indicates a model was successful in finding unique topics, while low diversity indicates a failure to discern topics from each other.

2) Results: We begin by comparing the performance of models on the 20 Newsgroups data set. Figure 2a shows the coherence and diversity of each model. On the x-axis is topic diversity, and on the y-axis is topic coherence. The models closest to the top right corner of the plot have the best topic coherence and topic diversity. Figure 2a shows that NLDA is clearly the best model for both topic coherence and topic diversity. GPUDMM and TND ($\mu = 10$) have the second best topic coherences, and TND ($\mu = 0$) has the second best topic diversity. Of all the data sets, this one contains the least amount of noise. It is interesting that in this context, using the estimated noise distribution from TND within NLDA leads to stronger results than LDA alone or estimating both the topic and noise distributions together in TND. This highlights that

[4] Parameters for sensitivity analysis across models: $k = 10, 20, 30, 50, 100$; $\alpha, \beta_0 = 0.01, 0.1, 1.0$; $\beta_1 = 0, 16, 25, 36, 49$; $\phi = 5, 10, 15, 20, 25, 30$; $\mu = 0, 3, 5, 10$



(a) 20 Newsgroups results.

(b) 50k Covid-19 results.

(c) Million Covid-19.

(d) Election 2020.

Fig. 2: Comparison of TND and NLDA to Baselines. Coherence (y) and Diversity (x). $k = 30$. $\beta_1 = 25$ for TND

even in a less noisy data set, modeling noise is important. We surmise that GPUDMM performs well on this data set because the number of words is smaller and the context of words is more stable in newsgroup data.[5]

Next, we compare the results of the best settings for each model on the 50k Covid-19 data set (Figure 2b). Again, topic diversity is plotted on the x-axis, while topic coherence is on the y-axis. On the left, we can see a cluster of the DMM, GPUDMM, and CSTM results. All three models produce topic sets with similarly low topic coherence and topic diversity. TND produces more coherent and diverse topics than DMM, GPUDMM, and CSTM. LDA produces similar results to TND. However, NLDA is the best model overall. In other words, first building the topics using the context-specific noise words and then using the estimated noise distribution to iteratively reduce the noise in LDA topics improves the topic coherence by 5.6% over TND and 10.5% over LDA. It also increases the topic diversity by 11.6% over TND and 10% over LDA.

We pause to reflect on the fact that the topic coherence scores for the Twitter data sets are much higher than the newsgroup data set. We think this is a result of the sparsity of the Twitter data. The percentage of high frequency words in the newsgroups that are not flood words is much higher than in the Twitter data, leading to more words overlapping across topics than for the Twitter data. This highlights the importance of separating high frequency content-rich words from high frequency content-poor words.

In order to show that these models are effective on larger data sets, we show the results of our models on the Million Covid-19 and Election 2020 data sets, compared with the results of the best-performing baseline models. While TND is slower than LDA, it is still considerably faster than other models that attempt to account for noise distributions and embedding spaces, like CSTM. With this in mind, we use this section to show the transferability and reusability of TND's noise distributions and how NLDA's ϕ parameter allows us to easily adapt a noise distribution to any number of topics. The results we present use the following parameters for TND: $\alpha = 0.1$, $\beta_0 = 0.01$, $\beta_1 = 25$, $k = 30$, $\nu = 1.5$, and $\mu = \{0, 3, 5, 10\}$. We tested NLDA on $k = \{10, 20, 30, 50, 100\}$ and $\phi = \{5, 10, 15, 20\}$, but show only the best parameter settings for clarity.

Figure 2c presents the topic coherence and topic diversity of the models built using the Million Covid-19 data set. In Figure 2c, topic diversity is plotted on the x-axis, and topic coherence is on the y-axis. Again, NLDA produces results with consistently high topic coherence and topic diversity across k values with $\phi = 10$. It is clear here that for TND, using $\mu > 0$, meaning incorporating the embedding space to some extent, improves the coherence of NLDA substantially. However, as a standalone model, TND is far more coherent when $\mu = 0$. TND alone is always at least as good as LDA, and also produces a noise distribution that can be used by researchers to better understand the context-specific noise present in their data sets. NLDA's coherence improvement over its competitors is amplified on the Million Covid-19 data set. Its topic coherence increases by 19% over TND and 24% over LDA. It also increases the topic diversity by 21% over TND and 18% over LDA. The coherence of topics likely drops due to the size of the data set – as more documents are added to a data set, more words exist in the vocabulary, and the overall sparsity of the data set increases, thereby reducing the probability of words appearing together.

Figure 2d presents the topic diversity and coherence of the best models on the Election 2020 data set. NLDA again outperforms the field in both metrics, followed by TND. It is as good as NLDA in terms of coherence, and nearly as diverse. LDA is the next best model followed by CSTM. DMM and GPUDMM performed poorly for both topic coherence and topic diversity. This results because of the prevalence of context-specific noise in all of their topics. CSTM, another model designed to filter noise from social media texts, does get improved topic diversity compared to DMM on both the Election 2020 and Million Covid data sets, but it fails to produce more coherent topics.

Finally, we consider the noise penetration rate. We worked with social scientists and CNN researchers to develop a set

[5] A natural question here would be, given that there are 20 newsgroups, why not use $k = 20$? We found that every model produced better results with $k = 30$.

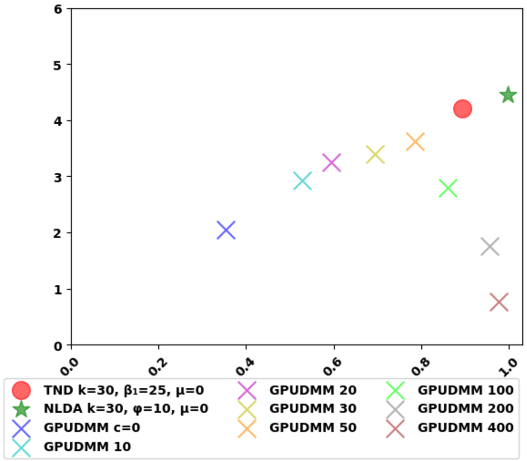


Fig. 3: 50k Covid-19 GPUDMM with a context-noise list.

of flood words (context-specific noise words) that were seen in open-ended survey responses about the 2020 presidential election. Throughout the election cycle, as noisy words appeared in responses that detracted from semi-automated topic generation, they were added to the list. We use that expert curated list of 50 context-specific noise words to help understand noise penetration. While this does not represent a full set of noise words in the Twitter data set, these noise words are the bellwethers of noise that detracts from the specificity and meaningfulness of topics identified from short text responses like social media posts. Examples of context-specific flood words included `Trump`, `Biden`, and `people`.

Table I shows the noise penetration rate for the Election 2020 data set. TND contains almost zero noise, highlighting its namesake – noise filtering. Both TND and NLDA have a significantly smaller noise penetration rate than LDA, DMM, and even CSTM, the other model designed to reduce noise. In other words, our approach for reducing noise is able to effectively remove large amounts of noise, with an improvement in penetration rate of more than 0.8 when compared to LDA for the Election 2020 data set. Table I highlights the tradeoff that we make when we move from TND to NLDA. TND has a smaller level of noise penetration in topics. NLDA has more diverse and coherent topics, but with a little more noise penetration.

Context Noise List. In addition to showing TND and NLDA's success on modeling noisy data sets, we also show the effectiveness of the context noise list on topic sets produced by other topic models. As we observed in the previous analysis, GPUDMM underperforms in comparison to NLDA on the Twitter data sets, while performing well on the 20 Newsgroups data set. This is a direct result of the large amount of context-specific noise in the Twitter data sets. In this experiment, we will generate a context-noise list using TND and use it to filter words from generated topic lists.

Model	LDA	DMM	GPUDMM	CSTM	TND	NLDA
Noise Pen. Rate	0.87	0.92	0.35	0.25	0.02	0.25

TABLE I: Noise Penetration in Election 2020 data set.

LDA	DMM	GPUDMM	CSTM	TND	NLDA
0.57	0.35	0.30	0.57	1.00	0.85

TABLE II: Fraction of unique topics agreed on by judges.

Model	Vice President	Covid-19	QAnon	Debates	Mail-in Voting	Other
LDA	2	1	2	4	0	5
DMM	0	0	1	10	1	2
GPUDMM	1	0	1	9	0	2
CSTM	1	1	1	5	2	4
TND	0	1	0	0	0	3
NLDA	2	2	0	1	1	7

TABLE III: Topic Labeling Judge Agreement.

Specifically, we fix $k = 30$ for TND, NLDA, and GPUDMM, and we use $\alpha = 0.1$, $\beta_0 = 0.01$, $\beta_1 = 25$, $k = 30$, and $\mu = 0$ as the parameters for TND to get an accurate noise distribution for use in NLDA and in the context-noise list. Figure 3 shows the impact of using a context-noise list of varying sizes with the GPUDMM topic set on topic coherence and topic diversity. Both TND ($\mu = 0$) and NLDA are shown for comparison purposes. We can see the topic diversity of GPUDMM increase as c increases, meaning that noise is to blame for much of the lack of diversity in the model. Even without incorporating the embedding space, the improvement in coherence is significant. When we look at topic coherence, we notice that when c gets very high ($c \geq 100$), the coherence of GPUDMM starts to fall off, even as its diversity continues to increase. In other words, removing small levels of context-specific noise can be useful for improving the topic coherence. Most of these words are flood words that do not get removed through traditional avenues of preprocessing. For example, in the Covid data set, words that would be removed by the context-noise list include flood words like 'covid19,' 'coronavirus,' and 'covid,' and general noise words like 'people,' 'today,' and 'many.' Removing these words from topics will improve topic diversity and coherence by virtue of the replacements for these words being more informative for their respective topics. TND and NLDA are able to selectively remove only the noise words that are not closely tied to coherent topics, leading them to have higher topic diversity and topic coherence than models using the context noise list. However, we believe that researchers will still find it valuable to be able to remove context-specific noise when using models that are already part of their pipeline.

C. Qualitative Analysis

For the Election 2020 data set, human judges were asked to label topics from LDA, DMM, GPUDMM, CSTM, TND, and NLDA. Our evaluation was conducted by 18 people, 10 male and 8 female. Most judges were college students. Judges were presented with five 'selected topics' from the Election 2020 data set that were dominant topics during the campaign. Judges were asked to label topics generated by each of the

Masks/Social Distancing					Testing/Symptoms					Vaccine				
LDA	CSTM	TND	NLDA μ=10	NLDA	LDA	CSTM	TND	NLDA μ=10	NLDA	LDA	CSTM	TND	NLDA μ=10	NLDA
social	covid19	fight^2	mask^2	mask^2	positive	covid19	care	test^2	test^3	vaccine^2	covid19	covid	vaccine	vaccine^2
coronavirus	mask^2	covid	spread	spread	test^4	positive	uk	symptoms	minister	russia	vaccine	covidupdates	treatment	study
china	spread	lets	face	social	symptoms	tested	social	study	home	worlds	coronavirus	usa	russia	sarscov2
video	help	lives	wear^2	face	results	hospital	covid	sarscov2	free	trials	russia	cdc	trials	disease
safe	wear	mask	protect	protect	sarscov2	coronavirus	test	disease	state	trial	first	thing	worlds	early
stay	protect	save	social	wear^2	friday	minister	staff	results	big	clinical	covid	vaccine	effective	treatment
city	coronavirus	message	stay	stop	monday	anyone	nhs	big	result^2	effective	trials	covidusa	trial	russia
home	wearing	line	safe	stay	free	covid	distancing	infection^2	infected	scientists	breaking	cnn	research	trial^2
distancing	covid	wear	stop	prevent	died	admitted	sign	heart	negative	event	says	research	event	app
wuhan	deaths	staysafe	distancing	immunity	infection	say	continue	found	admitted	company	died	cure	clinical	clinical

Mail-in Voting					Healthcare					Climate Change				
LDA	CSTM	TND	NLDA μ=5	NLDA	LDA	CSTM	TND	NLDA μ=5	NLDA	LDA	CSTM	TND	NLDA μ=5	NLDA
trump2020	realdonaldtrump	republican	vote^2	vote^2	people	taxes	care	sanders^2	healthcare^3	demdebate	demdebate	change	make	warren^2
maga	vote^2	put	2020election	2020election^2	dont	jobs	plan	warren^2	plan	change	sanders^2	ive	change	change
kag	election	call	call	call	healthcare^3	biden^3	proud	healthcare^2	coronavirus	climate	biden^2	climate	climate	climate
voting	whether	fact	election	mail	berniesanders	coun	health	klobuchar^2	public	economy	democraticdebat	medicare	tonight	shes
call	mail	mail	mail	service	americans	realjameswoods	wait	shes	congress	work	warren	demdebate	watch	feel
wwg1wga	im	political	person	ballots^2	talking	john	healthcare	give	reminder	jobs	pete	home	people	stage
patriots	tried	florida	start	mailin	campaign	abortions	demdebate	reminder	message	yang^2	kylekulinski	message	crisis	folks
follow	absentee	system	florida	florida	working	dear	men	senator	word	national	debate	voter	plan	senator
mail	true	demdebate	republicans	postal	plan	raise	lost	moderator	free	crisis	demconvention	twitter	andrewyang	close
florida	trump	true	ballot	poll	pay	left	usa	child	single	plan	people	word	hard	cmclymer

Fig. 4: Topic comparison between TND ($\mu = 0$), NLDA ($\mu = \{10, 0\}$), LDA, and CSTM. Million Covid-19 topics are on the top row, and Election 2020 topics are on the bottom row. Words are annotated with superscript numbers corresponding to the number of variants of the word in the top ten words.

models as one of the selected topics. If judges did not believe a selected topic was present, they could suggest another topic that applied, or they could indicate that no real topic existed. Thirty topics from each topic model were used in the human judgment experiment. Based on our topic coherence and topic diversity results, we expected variation in terms of the number of topics that would be interpretable by human judgement. Each topic was labeled independently by three judges. In our results, we considered a topic successfully labeled only if all three judges agreed on its label since that provided the best results for the baselines.

Table III shows the number of topic labels agreed upon by all three judges for each model. All the models except TND had 13 or 14 topics that were interpretable and had topic agreement. This suggests that there is a possible upper limit on the number of topics a generative model can successfully detect for a given k parameter. Surprisingly, TND does not perform as well on the qualitative analysis in terms of topic agreement. In other words, even though it is one of the top models in terms of quantitative measures, that did not hold true for qualitative measures on our Election data set. However, removal of noise is clearly important since two of the top three models include noise removal. In terms of topic coverage, only CSTM had 100% (5/5) topic coverage of the specified topics, followed by NLDA and LDA with 80% (4/5). The other three models had poor topic coverage.

Next, we assess topic uniqueness. Some topics created by topic models are repetitive while others are more unique. Table II shows the fraction of unique topics returned. Here we see the real strength of both TND and NLDA. All the topics for TND are unique - none overlap. NLDA only labels two duplicate topics (Vice President and Covid-19), while nearly half of the topics that LDA and CSTM find are duplicates. DMM and GPUDMM find almost exclusively the Debate topic, leading them to have very few unique topics.

As a final display of the quality of TND and NLDA topics, we show topics from the Million Covid-19 and Election 2020 data sets for TND, NLDA, LDA, and CSTM. Figure 4 shows six topics, three from Million Covid-19 (top row), and three from Election 2020 (bottom row). We specifically picked topics that the other methods showed more coherence on. As we mentioned in the introduction, the flood word 'covid-19' and similar words are common in LDA, CSTM, and TND. However, these flood words are absent from the NLDA topics.

Despite the appearance of a flood word in TND's Covid-19 topics, TND and NLDA's quality is apparent in both data sets. In the Election 2020 topic set, TND and NLDA are particularly effective compared to LDA, which contains far more noise than in the Million Covid-19 topic set. CSTM fails to separate noise from content in most topics in these domain specific data sets.

In the Million Covid-19 and Election 2020 data sets, TND and NLDA are particularly effective, finding strong topics for each depicted in Figure 4. NLDA, in some cases, is more coherent than TND. LDA and CSTM are less effective, and each fails to find a strong topic for at least one selected topic in each of the data sets. LDA and CSTM are capable of finding coherent topics, as they do in the Testing/Symptoms, Vaccine, and Climate Change (only LDA in this case) topics, but due to noise, other topics miss the mark.

VI. Conclusions

In this paper we have shown the importance of modeling both topics and noise for social media documents. We proposed creating topic-noise models that explicitly models both the topic and noise distributions of a data set. We present a new topic model, Topic Noise Discriminator (TND) that models both distributions and incorporates word embedding vectors to enhance the sampling algorithm of the generative model, leading to a better noise distribution in TND. We designed TND so that its noise distribution can be reused and integrated with other models, cutting down on computation costs. Second, we proposed an ensemble method with TND and LDA [5], Noiseless-LDA (NLDA), that leverages the noise distribution produced by TND with LDA to create high-coherence, high-diversity, low-noise topics. Third, we proposed creating and using a context noise list to remove noise from topic sets in an ad hoc way, after the topics have been generated, allowing noise removal to be used with any topic modeling algorithm.

We presented the effectiveness of these topic-noise models through extensive experiments using a standard data set (20 Newsgroups), and two novel, larger data sets obtained from Twitter. We showed through a quantitative and qualitative analysis that TND and NLDA are both capable of producing high-caliber topics from noisy data sets where traditional models fall short. We showed that the TND noise distribution can be integrated as a Context Noise List with other topic models to improve their coherence and diversity. Finally, we share our models and evaluation code on GitHub for others to use and innovate on.[6]

Future directions include adding a temporal aspect to TND, adding guidance based on user domain knowledge, and creating an online approximation of TND for faster inference of the noise distribution.

Acknowledgements

This work was supported by the National Science Foundation grant numbers #1934925 and #1934494, and by the Massive Data Institute (MDI) at Georgetown University. We would like to thank our funders.

References

[1] R. Churchill and L. Singh, "Percolation-based topic modeling for tweets," in *WISDOM 2020: KDD Workshop on Issues of Sentiment Discovery and Opinion Mining*, 2020.

[2] R. Churchill, L. Singh, and C. Kirov, "A temporal topic model for noisy mediums," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD)*, 2018.

[3] C. Chemudugunta, P. Smyth, and M. Steyvers, "Modeling general and specific aspects of documents with a probabilistic topic model," in *Advances in Neural Information Processing Systems (NIPS)*, 2007.

[4] C. Li, H. Wang, Z. Zhang, A. Sun, and Z. Ma, "Topic modeling for short texts with auxiliary word embeddings," in *ACM SIGIR Conference on Research and Development in Information Retrieval*, 2016, pp. 165–174.

[5] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, pp. 993–1022, 2003.

[6] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei, "Hierarchical dirichlet processes," *Journal of the American Statistical Association*, vol. 101, no. 476, pp. 1566–1581, 2006.

[7] D. M. Blei and J. D. Lafferty, "Dynamic topic models," in *International Conference on Machine Learning (ICML)*, 2006.

[8] J. D. Lafferty and D. M. Blei, "Correlated topic models," in *Advances in Neural Information Processing Systems (NIPS)*, 2006.

[9] W. X. Zhao, J. Jiang, J. Weng, J. He, E.-P. Lim, H. Yan, and X. Li, "Comparing twitter and traditional media using topic models," in *European Conference on Information Retrieval (ECIR)*, 2011.

[10] J. Qiang, P. Chen, T. Wang, and X. Wu, "Topic modeling over short texts by incorporating word embeddings," *CoRR*, vol. abs/1609.08496, 2016.

[11] X. Yan, J. Guo, Y. Lan, and X. Cheng, "A biterm topic model for short texts," in *International Conference on World Wide Web (WWW)*, 2013.

[12] J. Wang, L. Chen, L. Qin, and X. Wu, "Astm: An attentional segmentation based topic model for short texts," in *IEEE International Conference on Data Mining (ICDM)*, 2018.

[13] A. B. Dieng, F. J. Ruiz, and D. M. Blei, "Topic modeling in embedding spaces," *arXiv preprint arXiv:1907.04907*, 2019.

[14] K. Nigam, A. K. McCallum, S. Thrun, and T. Mitchell, "Text classification from labeled and unlabeled documents using em," *Machine Learning*, vol. 39, no. 2-3, pp. 103–134, 2000.

[15] X. Quan, C. Kit, Y. Ge, and S. J. Pan, "Short and sparse text topic modeling via self-aggregation," in *International Joint Conference on Artificial Intelligence*, 2015.

[16] J. Yin and J. Wang, "A dirichlet multinomial mixture model-based approach for short text clustering," in *ACM International Conference on Knowledge Discovery and Data Mining (KDD)*, 2014.

[17] D. Q. Nguyen, R. Billingsley, L. Du, and M. Johnson, "Improving topic models with latent feature word representations," *Transactions of the Association for Computational Linguistics*, vol. 3, pp. 299–313, 2015.

[18] X. Li, Y. Wang, A. Zhang, C. Li, J. Chi, and J. Ouyang, "Filtering out the noise in short text topic modeling," *Information Sciences*, vol. 456, pp. 83–96, 2018.

[19] F. Shahnaz, M. W. Berry, V. Pauca, and R. J. Plemmons, "Document clustering using nonnegative matrix factorization," *Information Processing Management*, pp. 373–386, 2006.

[20] S. P. Kasiviswanathan, P. Melville, A. Banerjee, and V. Sindhwani, "Emerging topic detection using dictionary learning," in *ACM International Conference on Information and Knowledge Management*, 2011.

[21] X. Yan, J. Guo, S. Liu, X. Cheng, and Y. Wang, "Learning topics in short texts by non-negative matrix factorization on term correlation matrix," in *SIAM International Conference on Data Mining (SDM)*, 2013.

[22] M. Cataldi, L. Di Caro, and C. Schifanella, "Emerging topic detection on twitter based on temporal and social terms evaluation," in *ACM KDD Workshop on Multimedia Data Mining*, 2010.

[23] H. F. de Arruda, L. da Fontoura Costa, and D. R. Amancio, "Topic segmentation via community detection in complex networks," *Chaos*, vol. 26, 2016.

[24] A. K. McCallum, "Mallet: A machine learning for language toolkit." 2002, http://mallet.cs.umass.edu.

[25] K. Lang, "20 newsgroups dataset," 1995. [Online]. Available: http://people.csail.mit.edu/jrennie/20Newsgroups/

[26] R. Churchill and L. Singh, "textprep: A text preprocessing toolkit for topic modeling on social media data," in *International ference on Data Science, Technology, and Applications (DATA)*, 2021.

[27] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1532–1543.

[28] J. H. Lau, D. Newman, and T. Baldwin, "Machine reading tea leaves: Automatically evaluating topic coherence and topic model quality," in *Conference of the European Chapter of the Association for Computational Linguistics*, 2014, pp. 530–539.

[29] A. B. Dieng, F. J. R. Ruiz, and D. M. Blei, "The dynamic embedded topic model," *CoRR*, vol. abs/1907.05545, 2019.

[6] The code repository can be found here: https://github.com/GU-DataLab/topic-modeling-tnd