Knowledge and Information Systems (2023) 65:2159–2186
https://doi.org/10.1007/s10115-022-01805-2

REGULAR PAPER

Using topic-noise models to generate domain-specific topics across data sources

Rob Churchill[1] · **Lisa Singh**[1]

Received: 8 February 2022 / Revised: 4 December 2022 / Accepted: 5 December 2022 /
Published online: 16 January 2023

Abstract
Domain-specific document collections, such as data sets about the COVID-19 pandemic, politics, and sports, have become more common as platforms grow and develop better ways to connect people whose interests align. These data sets come from many different sources, ranging from traditional sources like open-ended surveys and newspaper articles to one of the dozens of online social media platforms. Most topic models are equipped to generate topics from one or more of these data sources, but models rarely work well across all types of documents. The main problem that many models face is the varying noise levels inherent in different types of documents. We propose topic-noise models, a new type of topic model that jointly models topic and noise distributions to produce a more accurate, flexible representation of documents regardless of their origin and varying qualities. Our topic-noise model, Topic Noise Discriminator (TND) approximates topic and noise distributions side-by-side with the help of word embedding spaces. While topic-noise models are important for the types of short, noisy documents that often originate on social media platforms, TND can also be used with more traditional data sources like newspapers. TND itself generates a noise distribution that when ensembled with other generative topic models can produce more coherent and diverse topic sets. We show the effectiveness of this approach using Latent Dirichlet Allocation (LDA), and demonstrate the ability of TND to improve the quality of LDA topics in noisy document collections. Finally, researchers are beginning to generate topics using multiple sources and finding that they need a way to identify a core set based on text from different sources. We propose using cross-source topic blending (CSTB), an approach that maps topics sets to an s-partite graph and identifies core topics that blend topics from across s sources by identifying subgraphs with certain linkage properties. We demonstrate the effectiveness of topic-noise models and CSTB empirically on large real-world data sets from multiple domains and data sources.

R. Churchill and L. Singh have contributed equally to this work.

✉ Rob Churchill
rjc111@georgetown.edu

✉ Lisa Singh
lisa.singh@georgetown.edu

[1] Department of Computer Science, Georgetown University, 3700 O Street, Washington, D.C. 20007, USA

Keywords Generative topic modeling · Topic noise model · Topic blending

1 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. In order to form meaningful topics, generative topic models rely on repetition of word pairs within the same document. In shorter social media posts, word pairs are repeated at a far lower rate than in traditional documents. At these low frequencies, pairs of related words are often indistinguishable from pairs of unrelated words. As a result, 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. Domain-specific data sets contain a type of noise that we call *context-specific* noise. Context-specific noise is dependent on the domain of the data and on the topic set being generated. For instance, the word 'Hogwarts' belongs to a topic set generated using a document collection containing the Harry Potter books or containing a discussion about the Harry Potter books. However, the same word would most likely be considered noise in a data set about the COVID-19 pandemic. In domain-specific data sets, there is also a subset of context-specific noise that is relevant to the domain, but still pollute topics. These high-frequency context-specific noise words, *flood words* [1, 2], appear so frequently in documents that they dominate all topics in a topic set, making it difficult to discern different topics from each other. A few examples of flood words in a data set specifically about the COVID-19 pandemic would be *covid*, *coronavirus*, and *pandemic*.

We believe that given the prevalence of context-specific noise across social media data sets, we must understand noise instead of just ignoring it. 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, F 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 important for noisy domains.

To address these challenges, this extended paper proposes 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)** [5], 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 topic distributions *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.

Domain-specific data sets often only represent one of many facets of the conversation about the relevant domain. For instance, discourse about the COVID-19 pandemic has not taken place on Twitter alone. It has taken place all over the internet, in newsrooms, on television, and in books. Data sets generated from all these sources are important to understand from a topic modeling perspective. Topic models can and will be run on many of these data sets individually, but the means to compare topic sets from competing sources remains primitive, consisting mostly of costly and time-consuming human labeling and matching of topics. The incoherence and overlap of topics generated on noisy social media data makes it especially hard for humans to label and match topics. To support the comparison of topic sets from different data sources within the same domain, we propose *Cross-Source Topic Blending* (CSTB). CSTB is a method for combining or blending topics across topic sets by comparing the most probable words in topics generated by different data sources. CSTB uses an s-partite graph to combine topics from across s sources by identifying subgraphs with certain linkage properties. CSTB allows us to generate the *core topics* in a domain. Core topics help us understand the main themes within a domain, but also help us understand the origins of different points of view by isolating topics not shared across sources.

The contributions of this extended paper are as follows[1] (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 propose a method, Cross-Source Topic Blending (CSTB), for finding the core topics across different data sources within a domain using topics generated from each data source independently. (5) We conduct an extensive empirical analysis using two large Twitter data sets from the COVID-19 and Election 2020 domains, and the 20 Newsgroups data set and show the strength of explicitly modeling noise and using embeddings during the topic-noise modeling process. (6) We then use two other data sources, Reddit comments and newspaper articles, from different domains to show the quality of NLDA on different types of data, and to show the effectiveness of CSTB. (7) We publish our model code and other methods used in our experiments, along with our evaluation metrics.[2]

The paper is organized as follows: Sect. 2 presents the related literature. Section 3 defines terminology used throughout the paper. Section 4 presents our models. Section 5 proposes the CSTB method. Section 6 contains quantitative and qualitative experiments. Conclusions are presented in Sect. 7.

[1] This paper is an extension of *Topic-Noise Models: Modeling Topic and Noise Distributions in Social Media Post Collections*, which appeared in the IEEE International Conference on Data Mining (ICDM) 2021 [5]. The first three contributions were introduced in the original paper and the next three are new contributions in this paper.

[2] The code repository can be found here: https://github.com/GU-DataLab/gdtm.

2 Related literature

At their core, topic-noise models are unsupervised generative topic models. There are many types and variants of unsupervised topic models (see [6] for a survey), but in the paper, we focus on generative models. 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) [7], 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) [8], Dynamic Topic Models (DTM) [9], and Correlated Topic Models (CTM) [10]. 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 Sect. 4), 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.

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 [11]. DMM has been a key building block to many topic models that attempt to better model data sets containing short documents [4, 12–14]. GSDMM [13] 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.

Invented by Bengio et al. [15] and brought to the masses in the form of Word2Vec by Mikolov et al. [16, 17], large-scale word embedding vectors have become a popular NLP model to incorporate into topic models. Word embeddings are mathematical representations of words that can be added, subtracted, and compared like numbers. Words that are close to each other within an embedding space are more likely to be semantically related. A model called Lda2Vec alters the Word2Vec model to create embedding vectors for documents as well as for words [18]. Topics are also represented as vectors within the same embedding space, allowing for the measurement of similarity between words, documents, and topics. Generalized Polya Urn DMM (GPUDMM) [4] uses the DMM model, and alters the sampling algorithm to incorporate word embeddings. The related words of an observed word are sampled alongside it to produce more coherent topics in noisy data. 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 [19]. Dieng et al. propose a generative model similar to LDA in essence, but which draws topic words directly from the embedding space [20]. This approach has been applied to temporal topic modeling as well [21]. While all these models use new NLP techniques, they do not explicitly model a noise distribution. LF-LDA and LF-DMM replace the topic-word distribution of the models with a mixture of topic-word distribution and latent features derived from word embeddings [14].

Twitter-LDA [22] attempts to create better topics on social media data in a unique way. Given a set of tweets, along with the metadata such as the author of each tweet, Twitter-LDA first identifies every unique author in the data set. It then retrieves each author's entire corpus of public tweets and aggregates them into a single, larger document for each author. As a result, LDA is presented with longer pseudo-documents in which to find topics. The idea behind long pseudo-documents replacing shorter texts is that word co-occurrence will increase, resulting in better topics. While this is a good approach for a topic model whose purpose is to approximate topics for the general Twitter platform, this does not extend to domain-specific data sets. People are free to tweet about whatever they please, so there is no guarantee that the tweets retrieved for each author will remain within the desired domain. In fact, it is more likely than not that out-of-domain tweets will lead to topic drift, resulting in less accurate topics for the domain.

Other models have attempted to aggregate tweets into longer pseudo-documents in a manner similar to Twitter-LDA. Embedding-based Topic Model (ETM) [23] uses word embeddings to aggregate short texts into long pseudo-documents, and then infers topics from the pseudo-texts. This approach does not require the retrieval of each author's tweets, making it more practical than Twitter-LDA for domain-specific topic modeling. Self-Aggregating Topic Model (SATM) [12] also aggregates short texts into pseudo-documents. Instead of relying on word embeddings for aggregation, the authors run LDA on the original data set, and then use the approximated topics to aggregate documents into longer texts. They run LDA a second time on the longer texts to get their final topics. Pseudo-document-based Topic Model (PTM) [24] attempts to improve on SATM by adding the assumption that short texts are generated from the longer pseudo-documents. PTM observes the original short texts, and draws a longer pseudo-document from which the short text may have been generated. The topic for the short text is drawn from the distribution of the pseudo-document, thereby modeling the short text as the pseudo-document with missing words.

Yan et al. perform topic modeling on pairs of terms with high co-occurrence in their Biterm Topic Model (BTM) [25]. Instead of modeling document-level word patterns, BTM attempts to model data set-level word patterns. The authors extract pairs of words that appear frequently within documents and perform inference directly on the word pairs, instead of the documents. Neural Variational Document Model (NVDM) is a neural language model that can be used as a topic model [26]. It uses a variational auto-encoder to produce a document embedding for each document in the data set. These embeddings can then be clustered into topics. There are also reinforcement learning-based neural models that use generative adversarial models to find topics [19, 27–29].

Li et al. [30] deal with filtering noise from topics with their topic model, CSTM. The authors base their model on DMM [11], 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 [1, 2, 31–35]. 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.

3 Background and 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 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 $T = \{z_0, z_1, ..z_{k-1}\}$. 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 clarity to 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 topic 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 Sect. 6.

In summary, this paper attempts to answer the following question. 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?

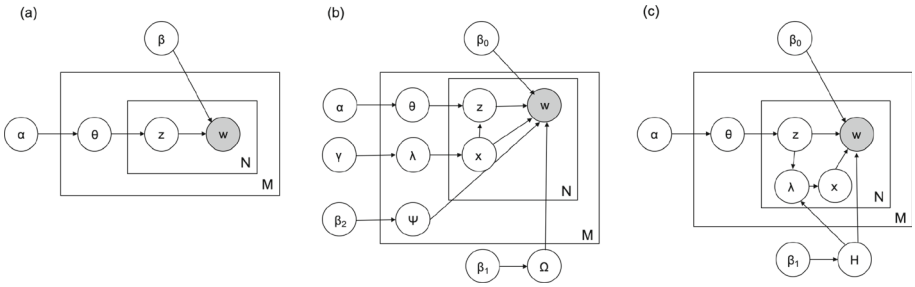


Fig. 1 LDA (**a**), SWB (**b**), and TND (**c**) Graphical Models

4 Topic-noise models

In this section, we describe our proposed models in detail. In order to relate our models to the most relevant prior work, we begin by presenting the plate notation and describing LDA [7] and SWB [3] (Sect. 4.1). We then describe our proposed topic-noise model (TND) (Sect. 4.2), and the extension using embedding sampling (Sect. 4.3). Finally, we describe our approach for combining existing generative and non-generative models with the noise distribution generated by TND (Sect. 4.4).

4.1 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. [7], 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 Fig. 1b), or from the independently computed global background distribution (Ω in Fig. 1b). The decision of which distribution to draw from is controlled by x, which is sampled from a document-specific multinomial λ conditioned on γ.

4.2 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 Fig. 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* 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 Fig. 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.

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

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 TNDis 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.

4.3 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.

4.4 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.

4.4.1 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.[3] 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 $z \in T$, we remove w_i from z according to a Beta distribution (Eq. 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.

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

For each word w_i in topic z, 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 z, w_i's frequency in the noise distribution is incremented, marking it as noise once again. If w_i is retained in z, w_i's frequency in the noise distribution is increased by θ_z^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 (Eq. 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.

4.4.2 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 $z \in T$, we remove word w_i from z 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.

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

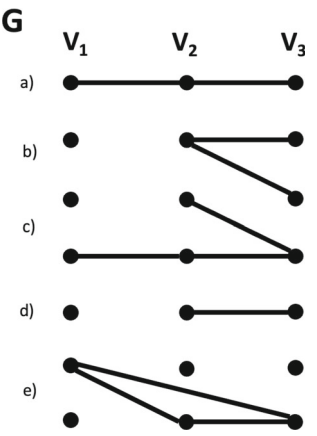


Fig. 2 Example of Graph created for CSTB. $\ell = 3$

5 Cross-source topic blending

Topic-noise models are very useful for producing coherent, diverse topic sets on domain-specific data sets. There are times when it is also beneficial to blend topic sets that are related to a single domain, but constructed using different data sources. We would like to explore a principled way to combine these topic sets to identify a combined set that gives more insight into *core topics* that exist across data sources for a specific domain.

This leads us to cross-source topic blending (CSTB). Instead of running one large model on a blended data set with parameters that might not be optimal for any one data source, we run one model on each individual data source, with the best parameter settings for each. We then use CSTB to blend the topic sets trained on individual data sources, merging similar ones to avoid repetition. This approach allows us to incorporate many very different data sources into our final topic set without having to worry about finding an illusive perfect set of parameters across sources that likely have varying noise distributions.

At a high level, cross-source topic blending works as follows: (1) Given s data sources in a domain, run a topic model for each source, resulting in s topic sets. (2) Construct an s-partite graph G, where each partition V_i is a topic set, and each node is a topic, z. (3) Add an edge between two nodes if the topics they represent are similar, i.e. have sufficient overlap. (4) Identify connected components C in G that do not contain more than one node from each partition s, and for each connected component in C, merge the associated topics to create a single blended topic. (5) Return the final set of core topics.

The rest of this section describes the graph construction for G and the topic blending using G.

5.1 Graph construction

Let G be a graph with t vertices, where each vertex represents a topic. Let s be the number of data sources within a domain. We define G to be an s-partite graph whose vertices can be divided into s disjoint sets. In this s-partite graph, an edge cannot exist between vertices within the same partition. For ease of exposition, suppose $s = 3$. Then, $G = (V_1, V_2, V_3, E)$ where the vertex set $V = (V_1 \cup V_2 \cup V_3)$, and $(V_1 \cap V_2 \cap V_3) = \{\}$. For each vertex pair V_i

and V_j, if topics z_i and z_j share at least χ of their ψ most-probable words, then an edge is added to G that connects V_i and V_j.

This graph structure links related topics across data sources. Similar topics form connected components, isolated from other topics that do not share the same meaning. It allows for the detection of transitive similarity between topics generated from different sources that might not directly share the same words (i.e. topic A is similar to topic B, and topic B is similar to topic C, therefore, topic A is similar to topic C).

5.2 Finding blended topics

Using G, we find the set of connected components of size ℓ that contain at most one vertex from each partition, where ℓ is a number between 1 and s. Each connected component is blended into a single core topic. Figure 2 shows an example of a CSTB graph G, with $s = 3$ and $\ell = 3$. We can see in the example that there are five connected components containing at least two vertices, labeled a through e. Connected components a and e are core topics because they contain at most one node from each partition and are of size at least $\ell = 3$. In contrast, connected components b and c are not core topics because they contain more than one node from a single partition. Finally, d is not a core topic because it contains fewer than $\ell = 3$ vertices. The process of finding core topics takes $O(s(V + E))$, which is the time it takes to find all connected components in an undirected s-partite graph. In practice, $|V_i| < 100$, so this computation is fast.

CSTB is simple, but hinges on having high quality topics. The graph structure rewards coherent topics and punishes noisy topics or topic sets with many overlapping words. If a topic set contains topics that have many overlapping words, then they are more likely to appear in the same connected component. This will result in that connected component not being deemed a core topic. If a topic set contains noisy topics, then the noise words pose the threat of joining topics in a component that do not belong together, resulting in either poor core topics, or missing core topics. However, if a topic set contains coherent, diverse topics, the graph structure will lead to clear delineations between connected components, giving us coherent, diverse core topics. In the next section, we will demonstrate that CSTB is a reasonable strategy for combining topics efficiently for topics generated from topic-noise models.

6 Empirical evaluation

In this section, we present our empirical evaluation. We evaluate the three variants proposed in Sect. 4: 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 (Sect. 6.1). We then present our quantitative evaluation (Sect. 6.2), followed by our qualitative analysis (Sect. 6.3) and our analysis of Cross-Source Topic Blending (Sect. 6.5).

6.1 Experiment setup

6.1.1 Baseline algorithms

We compare our proposed models to the following state-of-the-art models: Latent Dirichlet Allocation (LDA),[4] Gibbs Sampling Dirichlet Multinomial Mixture (DMM) [13], Generalized Polya Urn Dirichlet Multinomial Mixture (GPUDMM) [4], and Common Semantics Topic Model (CSTM) [30]. These topic models each represent a unique facet of generative topic models as explained in Sect. 2. 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.

6.1.2 Data sets

In this analysis, we consider multiple data sets. We begin with the twenty newsgroups data set. We also consider data sets from sources revolving around the domains of COVID-19 and Election 2020. Our first data set is a subset of the Twenty Newsgroups data set [37]. 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.

For the *Election 2020* domain, posts were collected 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. The Twitter data set consists of over 1.4 million tweets, focusing on topics related to the November election. We also use a Reddit data set consisting of 1,284,324 comments on posts about the election.

For the COVID-19 domain, we have two Twitter data sets. These data sets contain posts about the 2020 COVID-19 pandemic. Using Covid-19 related hashtags, we collected COVID-19-related tweets through the Twitter Streaming API. The *50k Covid-19* Twitter data are 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* Twitter data contains over 1 million tweets about the 2020 Covid-19 pandemic. The Reddit data contain 147,436 comments on posts related to the pandemic. In the COVID-19 domain, in addition the Twitter and Reddit data sources, we were also able to collect newspaper articles. The News data set contains 215,261 news articles related to the pandemic, collected from newspapers in the USA. The Million Twitter, Reddit, and News data sets were collected between August 1 and September 30. The COVID-19 and Election 2020 data sets can be used to test the ability of the different models to produce high-quality topics on varying types of data sets.

6.1.3 Data preprocessing

Data preprocessing can have a significant impact on topic models [38]. 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.

[4] Specifically the MALLET implementation of LDA [36].

6.1.4 Model parameters

In order to provide a thorough sensitivity analysis for each of our models, we test each model with many different parameter settings.[5] For ease of exposition, 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 [39] 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.

6.2 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.

6.2.1 Evaluation metrics

To assess a model's ability to detect coherent, meaningful topics, we use normalized pointwise mutual information (NPMI) [40]. 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 [12, 20, 21, 23]. 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:

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

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

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

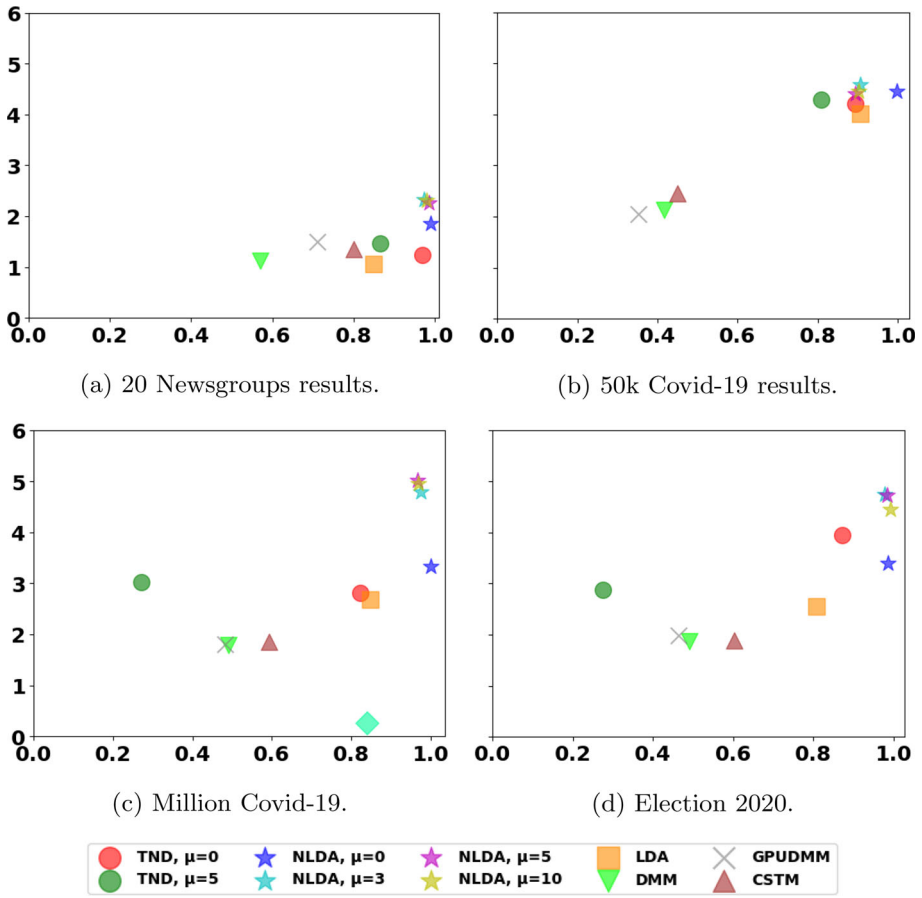


Fig. 3 Comparison of TND and NLDA to Baselines. Coherence (y) and Diversity (x). $k = 30$. $\beta_1 = 25$ for TND

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 [21]. High topic diversity indicates a model successfully found unique topics, while low diversity indicates a failure to discern topics from each other.

6.2.2 Results

We begin by comparing the performance of models on the 20 Newsgroups data set. Figure 3a 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 3a 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 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.[6]

Next, we compare the results of the best settings for each model on the 50k COVID-19 data set (Fig. 3b). 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.

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 Twitter and Election 2020 Twitter 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 $\phi = \{5, 10, 15, 20, 25, 30\}$, but show only the best parameter settings for clarity.

Figure 3c presents the topic coherence and topic diversity of the models built using the Million COVID-19 data set with and without embedding sampling. By adjusting μ, we can control how many words are sampled from the embedding space. Increasing μ causes more words to be sampled, whereas setting it to zero will cause no words to be sampled. In Fig. 3c, 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. It is interesting to note that there is not much difference in the results when $\mu = 3$ and $\mu = 10$. In other words, the number of words used for the embedding sample is less important than just incorporating it. 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 Twitter 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 more sets of words appearing together.

Figure 3d presents the topic diversity and coherence of the best models on the Election 2020 Twitter 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. Similar to the Million Covid-19 data set, adding embeddings to the sampling does improve the topic coherence for

[6] 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 $k = 30$.

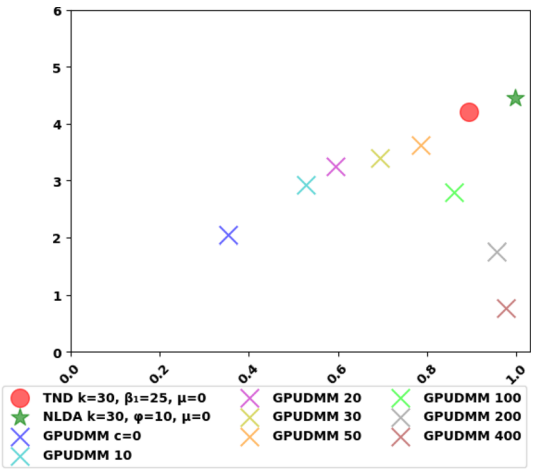


Fig. 4 50k COVID-19 GPUDMM with a context-noise list

Table 1 Noise Penetration in Election 2020 data set

Model	LDA	DMM	GPUDMM	CSTM	TND	NLDA
Noise pen. rate	0.87	0.92	0.35	0.25	0.02	0.25

NLDA, but not TND. 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 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 1 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 1 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.

Table 2 Fraction of unique topics agreed on by judges

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

Table 3 Topic Labeling Judge Agreement. For each Model and Topic Labels, we show the number of topics labeled with the Topic Label 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

Masks/Social Distancing

LDA	CSTM	TND	NLDA μ=10	NLDA
social	covid19	fight²	mask²	mask²
coronavirus	mask²	covid	spread	spread
china	spread	lets	face	social
video	help	lives	wear²	face
safe	wear	mask	protect	protect
stay	protect	save	social	wear²
city	coronavirus	message	stay	stop
home	wearing	line	safe	stay
distancing	covid	wear	stop	prevent
wuhan	deaths	staysafe	distancing	immunity

Testing/Symptoms

LDA	CSTM	TND	NLDA μ=10	NLDA
positive	covid19	care	test²	test³
test⁴	positive	uk	symptoms	minister
symptoms	tested	social	study	home
results	hospital	covid	sarscov2	free
sarscov2	coronavirus	test	disease	state
friday	minister	staff	results	big
monday	anyone	nhs	big	result²
free	covid	distancing	infection²	infected
died	admitted	sign	heart	negative
infection	say	continue	found	admitted

Vaccine

LDA	CSTM	TND	NLDA μ=10	NLDA
vaccine²	covid19	covid	vaccine	vaccine²
russia	vaccine	covidupdates	treatment	study
worlds	coronavirus	usa	russia	sarscov2
trials	russia	cdc	trials	disease
trial	first	thing	worlds	early
clinical	covid	vaccine	effective	treatment
effective	trials	covidusa	trial	russia
scientists	breaking	cnn	research	trial²
event	says	research	event	app
company	died	cure	clinical	clinical

Climate Change

LDA	CSTM	TND	NLDA μ=5	NLDA
demdebate	demdebate	change	make	warren²
change	sanders²	ive	change	change
climate	biden²	climate	climate	climate
economy	emocraticdeb	medicare	tonight	shes
work	warren	demdebate	watch	feel
jobs	pete	home	people	stage
yang²	kylekulinski	message	crisis	folks
national	debate	voter	plan	senator
crisis	lemconventio	twitter	andrewyang	close
plan	people	word	hard	cmclymer

Mail-in Voting

LDA	CSTM	TND	NLDA μ=5	NLDA
trump2020	aldonaldtrum	republican	vote²	vote²
maga	vote²	put	2020election	2020election²
kag	election	call	call	call
voting	whether	fact	election	mail
call	mail	mail	mail	service
wwg1wga	im	political	person	ballots²
patriots	tried	florida	start	mailin
follow	absentee	system	florida	florida
mail	true	demdebate	republicans	postal
florida	trump	true	ballot	poll

Healthcare

LDA	CSTM	TND	NLDA μ=5	NLDA
people	taxes	care	sanders²	healthcare³
dont	jobs	plan	warren²	plan
healthcare³	biden³	proud	healthcare²	coronavirus
berniesanders	coun	health	klobuchar²	public
americans	aljameswood	wait	shes	congress
talking	john	healthcare	give	reminder
campaign	abortions	demdebate	reminder	message
working	dear	men	senator	word
plan	raise	lost	moderator	free
pay	left	usa	child	single

Fig. 5 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

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 with 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.

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 4 shows the impact of using a context-noise list of varying sizes (c = size of the noise list) 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. 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 for GMM, but removing too many impacts its ability to create more coherent topics. When looking at the words in the noise list, we find that most of these words are flood words that do not get removed through traditional avenues of preprocessing. For example, in the Covid-19 Twitter 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.

6.3 Qualitative analysis

For the Election 2020 Twitter 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 Twitter data set that were dominant topics during the campaign. Judges were asked to label topics generated by each of the 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 judgment. 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 3 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

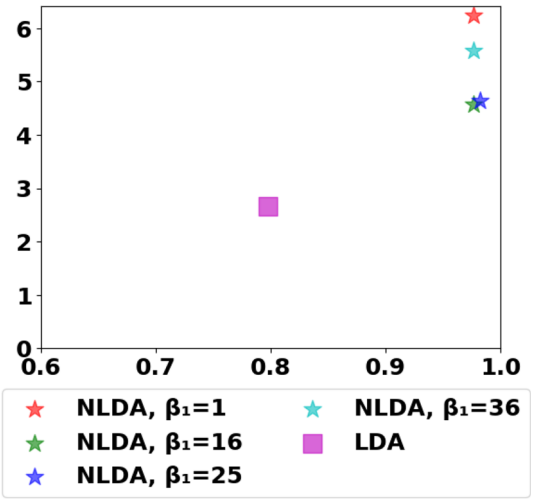


Fig. 6 COVID-19 Reddit β_1 Sensitivity Analysis. Coherence (y) and Diversity (x). $k = 30$. $\phi = 25$

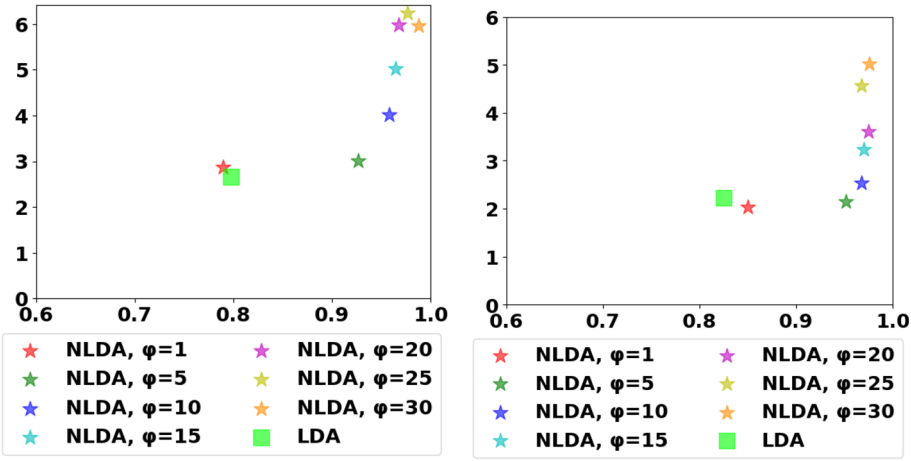
on the qualitative analysis in terms of topic agreement. In other words, even though it is one of the top models in terms of quantitative measures, that did not hold true for qualitative measures on our Election data set. However, removal of noise is clearly important since two of the top three models 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 2 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.

To display of the quality of TND and NLDA topics, we show topics from the Million COVID-19 Twitter and Election 2020 Twitter data sets for TND, NLDA, LDA, and CSTM. Figure 5 shows six topics, three from Million COVID-19 Twitter (top row), and three from Election 2020 Twitter (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 Twitter topics, TND and NLDA's quality is apparent in both data sets. In the Election 2020 Twitter 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 Twitter and Election 2020 Twitter data sets, TND and NLDA are particularly effective, finding strong topics for each depicted in Fig. 5. 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.



(a) Covid-19. (b) Election 2020.

Fig. 7 Sensitivity Analysis of ϕ-values on Reddit data sets. Coherence (y) and Diversity (x). $k = 30$. $\beta_1 = 1$ for TND

6.4 Evaluating topic-noise model performance across data sources

Our main analysis of the quality of TND and NLDA was performed on Twitter data and the colloquial Twenty Newsgroups data set. However, these models are not limited to Twitter and baseline data sets. In this section, we present the results of using NLDA on the Reddit and newspaper data sets for the COVID-19 domain, and on the Reddit data set for the Election 2020 domain. We compare the coherence and diversity across data sources to that of LDA and compare the optimal parameter settings for each data source, explaining what leads to parameter changes and why. We then display topics as they are captured in different data sources to highlight both the robustness of topic-noise models and the differences in conversations that occur in different forums.

First, we were interested in tuning the β_1 parameter. For the Twitter data sets, we had to set it to 25 in order to remove enough noise to get good topics. Figure 6 shows the effects of changing β_1 on the Covid-19 Reddit data set. We fix $\phi = 25$ to conduct our sensitivity analysis for β_1. Topic diversity is on the x-axis, and topic coherence is on the y-axis. We can clearly see that while diversity is almost completely unaffected, topic coherence skyrockets when β_1 is lowered. We saw nearly identical trends in the Election 2020 Reddit data set and in the COVID-19 News data set. This finding is unsurprising. Twitter data are known to be noisier than other social media platforms, due to hashtags, links, user handles, slang, and shorter documents. Reddit comments and newspaper articles, which are typically longer and better written than tweets, require less strenuous noise detection than Twitter data. We find that the noise words identified are better with a lower β_1 setting because, in Reddit and newspaper data, they are noticeably different than most topic words. Tuning β_1 for data sets with different levels of noise allows for important data source customization.

Figures 7a and b show the coherence and diversity scores of the Reddit data sets for the COVID-19 and Election 2020 domains as we vary ϕ. We can see a significant trend toward higher ϕ-values producing significantly higher coherence scores, while low values produce lower coherence and diversity.

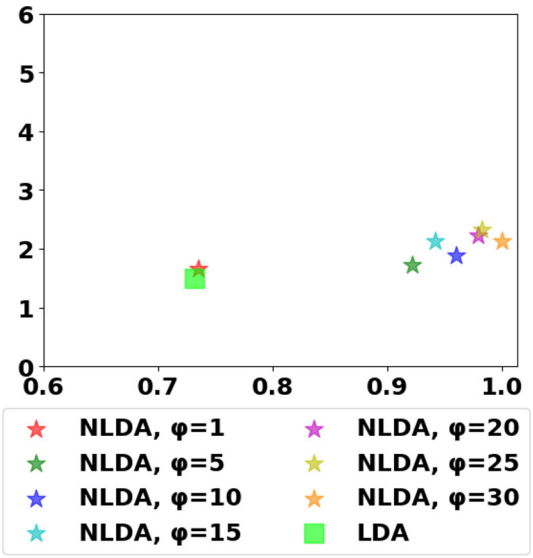


Fig. 8 COVID-19 News Sensitivity Analysis of ϕ-values. Coherence (y) and Diversity (x). $k = 30$. $\beta_1 = 1$

If we remove no noise words at all, our coherence and diversity scores should be identical to that of LDA. This is approximately what we see when $\phi = 1$, what is essentially the least harsh noise removal setting for NLDA. By removing very few noise words, we barely differ in quality from LDA. However, we can see that the higher we set ϕ, the higher coherence and diversity we get. This is because when we set β_1 to a lower value and are able to model noise accurately, we want to aggressively remove those noise words. Where with Twitter data we must tread lightly lest we accidentally remove topic words, in these less noisy sources, we are more certain that the words in the noise distribution are truly noise words. By lowering β_1 and increasing ϕ, we are stating that we are able to more accurately capture noise and dispose of those noise words, thereby improving the topic coherence.

Figure 8 shows the coherence and diversity scores of the News data set for COVID-19. While the increase in coherence is not as pronounced as in the Reddit data sets, there is a clear improvement in topic diversity, along with a small improvement in coherence. The higher word co-occurrence rates implicit in longer documents such as newspaper articles means that topic coherence is somewhat limited compared to data sources with shorter documents. The improvement of coherence and diversity in the Reddit and News data sets by increasing ϕ is, similar to the reduction of β_1, unsurprising. By more aggressively removing noise that we are more certain should be removed, we are improving the quality of our topics.

Figure 9 shows the same topics from Fig. 5 as they were found by NLDA in the Twitter, Reddit, and News data sources. The words shown are the most probable words per topic, rearranged to show patterns over data sources. We can see significant similarities between these topics in each data source, likely because they are some of the most commonly referred-to topics in the domain. Still, we can see slight differences. For instance, in the Masks and Social Distancing topic, we can see that in the Reddit conversation, people are more concerned with transmission, and masks covering one's mouth and nose. In the News data source, the focus is on rules, guidelines, and businesses, differing from both Twitter and Reddit. The News data source again breaks with the Reddit and Twitter topics for the testing category, as it focuses on testing on college campuses. On the Vaccine topic, the News data source focuses on studies, experts, and drug trials. The Reddit data source includes words like *data*, *side*

Masks/Social Distancing			Testing/Symptoms			Vaccines		
Twitter	Reddit	News	Twitter	Reddit	News	Twitter	Reddit	News
mask²	mask²	mask²	test³	test⁴	test⁴	vaccine²	vaccine²	vaccine²
wear²	wear²	wear²	minister	positive	positive	study	effective	effective
face	face	face	infected	infection²	university	trial²	trial²	trial
social	spread	social	negative	negative	campus	disease	phase	disease
protect	protect	distancing	state	number	contact	early	herd	experts
spread	virus	spread	big	days	staff	treatment	immunity	drug
stop	nose	rules	result²	case	quarantine	russia	safety²	research
stay	mouth	guidelines	home	rate	results	sarscov2	data	study
prevent	transmission	york	free	asymptomatic	college	app	side	scientists
immunity	people	businesses	admitted	symptoms	symptoms	clinical	shot	infection

Fig. 9 COVID-19 Topics from Reddit and News data sources

[effects], *herd*, and *immunity*, hinting at peoples' hesitation to get vaccinated. No two data sources seem to produce more similar topics than the other consistently. Twitter and Reddit produce a similar *Testing/Symptoms* topic, but do not align on the *Vaccine* topic. On the *Masks* topic, each data source shares some similarities with the others, but are distinctly different. The ability to see how topics vary across different data sources is a valuable capability, especially in a situation like a pandemic where access to information can have widespread effects on the population.

6.5 Cross-source topic blending

Using data from different sources, we now show how we can find topics that matter most within a specific domain. We will being use a graph G to blend topics based on their most probable words. If topics from different domains share χ of their ψ most probable words, then they are blended together to make a larger, cross-domain topic. This can be done with two or more domains, although the more domains, the more difficult it can be to find related topics. To better understand the impact of different parameter settings on the final set of core topics, we conduct a detailed sensitivity analysis and then highlight the discovered topics.

Sensitivity analysis If one wants a larger set of core topics, χ should be set low, and ψ should be set high. This will result in more edges in G, as more topics will match each other. The number of edges in G does not, however, guarantee a larger set of core topics. Recall that a connected component must contain at most one node from each partition to be a core topic. If there are too many edges in G, then components will be more likely to include multiple vertices from the same partition, resulting in their disqualification from the core topic set. However, having too few edges in G will result in fewer components of size ℓ or larger. In practice, setting χ and ψ, the parameters that decide the proportion of most-likely words that must be shared between two topics, is data source and domain dependent. We had three data sources (Twitter, Reddit, and News), so we set $\ell = \{2, 3\}$, to determine whether or not there was a high overlap in topics over all three data sources. We tested $\chi = \{1, 3, 5, 10\}$ with $\psi = \{1, 3, 5, 10, 15, 20, 30, 50\}$.

We did not conduct an external validation of these topics since we did that for the individual sources. Instead, we focused on identifying a set of guidelines for capturing core topics. For our data sets, we found that $\chi < 5$ was too small a number to find meaningful connections between topics. When $\ell = 2$ and $\chi = 1$, we found that many core topics were clearly two separate topics which happened to share a single word. Figure 10 shows an example of a mismatched topic when $\chi = 3$. The mismatched topic is a blend of two topics that do not

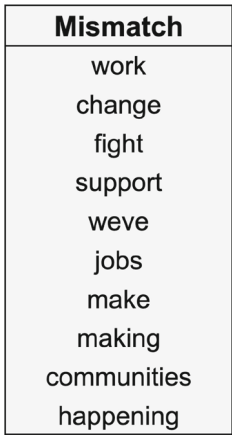


Fig. 10 An example of a mismatched topic when χ is set too low

truly belong together. This occurs because a low χ value allows too many edges in G. When $\ell = 3$, very few if any core topics were found, because larger connected components tended to have two vertices in the same partition and therefore, were not merged. We found that $\chi = 10$ was too many words to match when $\psi < 30$, so few core topics were found. When $\psi \geq 30$, core topics were too easily found, resulting in noisier core topics being included. The best parameters we found were $\chi = 5$, $\psi = 15$. While this is the same 1:3 ratio of $\chi : \psi$ as $\chi = 10$, $\psi = 30$, in general the individual probabilities of each of the first fifteen words in each topic are much higher than those of the first thirty. So, the likelihood of noise words causing a false match of topics when $\psi = 15$ is much lower than when $\psi = 30$. We found that with sufficient settings for χ and ψ, there was indeed a reasonable overlap in topics over all three data sources.

The final core topics presented in this paper were derived with $\chi = 5$, $\psi = 15$, $\ell = 3$. Figure 11 shows the core topics for the COVID-19 domain. These five topics, *Cases*, *Testing*, *Vaccines*, *Masks*, and *Government* represent important facets and phases of the pandemic. COVID-19 cases were carefully tracked throughout the pandemic. Testing was the first line of defense against the virus and enabled the re-opening of many schools and universities. Vaccines were developed to immunize the world in an effort to slow the rate of infections, particularly severe infections. Masks and social distancing were mandated almost ubiquitously around the world in an effort to keep people from transmitting the disease while in public. Finally, governments were front and center during the pandemic. Aside from the 2020 United States presidential election, governments led the pandemic response, setting public health policies, tracking cases, procuring tests, funding and regulating vaccines, and more.

In order to show that our interpretation of these topics is in line with the interpretations of others, we asked a panel of seven judges, 5 female and 2 male, to provide a label for each of the five core topics. The task was open-ended. Judges were not given labels to choose from. Table 4 shows the percent of judges who provided labels in agreement with each other.[7] Three of the topics were agreed upon by all judges, and one was agreed on by six out of seven. The Testing topic was split. Three judges provided the label *Testing*, but three others provided a label along the lines of *College Covid Policies*. Looking at the topic, there are four variants of the word *test*, but many words relate to colleges and universities coping with the

[7] Examples of agreeing labels would be *covid cases* and *covid stats*, or *masks* and *mask regulations*.

Cases	Testing	Vaccines	Masks	Government
cases	test[4]	vaccine[2]	mask[2]	trump[2]
deaths	positive	study	spread	president
county	university	health	social	coronavirus
numbers	minister	sarscov2	wearing	plan
total	home	herd	face	biden
reported	free	immunity	wear	campaign
data	infected	disease	distancing	media
health	state	trials	people	democratic
number	case	virus	stop	americans
india	students	early	public	national

Fig. 11 Core Topics in the COVID-19 domain as found by CSTB

Table 4 Percent judge agreement on COVID-19 core topics

Topic	Cases	Testing	Vaccines	Masks	Government
Agreed %	100	43	85	100	100

pandemic. As we noted above, colleges were at the forefront of the testing debate, so finding these words together is unsurprising. Overall, there was high agreement by judges on four of the five topics. Cross-source topic blending is intuitive to use and interpret, and informative when one wants to better understand a domain using multiple data sources.

7 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 [7], 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. Fourth, we proposed the cross-source topic blending (CSTB) heuristic for finding the core topics across data sources within a domain using a novel graph structure.

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. We showed the capability of CSTB to find the most relevant topics in a domain using three data sources

from the COVID-19 domain. Finally, we share our models and evaluation code on GitHub for others to use and innovate on.[8]

Future directions include applying TND and CSTB to other domains and data sources, as well as improving the efficiency of our models.

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. We would also like to thank the S3MC project and the CNN Breakthrough project for their help with identification of noise words for the 2020 US election.

References

1. Churchill R, Singh L (2020) Percolation-based topic modeling for tweets. In: WISDOM 2020: KDD workshop on issues of sentiment discovery and opinion mining
2. Churchill R, Singh L, Kirov C (2018) A temporal topic model for noisy mediums. In: pacific-asia conference on knowledge discovery and data mining (PAKDD)
3. Chemudugunta C, Smyth P, Steyvers M (2007) Modeling general and specific aspects of documents with a probabilistic topic model. In: Advances in neural information processing systems (NIPS)
4. Li C, Wang H, Zhang Z, Sun A, Ma Z (2016) Topic modeling for short texts with auxiliary word embeddings. In: ACM SIGIR conference on research and development in information retrieval, pp. 165–174
5. Churchill R, Singh L (2021) Topic-noise models: modeling topic and noise distributions in social media post collections. In: International conference on data mining (ICDM)
6. Churchill R, Singh L (2021) The evolution of topic modeling. ACM Comput. Surv
7. Blei DM, Ng AY, Jordan MI (2003) Latent dirichlet allocation. J Mach Learn Res 3:993–1022
8. Teh YW, Jordan MI, Beal MJ, Blei DM (2006) Hierarchical dirichlet processes. J Am Stat Assoc 101(476):1566–1581
9. Blei DM, Lafferty JD (2006) Dynamic topic models. In: International conference on machine learning (ICML)
10. Lafferty JD, Blei DM (2006) Correlated topic models. In: Advances in neural information processing systems (NIPS)
11. Nigam K, McCallum AK, Thrun S, Mitchell T (2000) Text classification from labeled and unlabeled documents using EM. Mach Learn 39(2–3):103–134
12. Quan X, Kit C, Ge Y, Pan SJ (2015) Short and sparse text topic modeling via self-aggregation. In: International joint conference on artificial intelligence
13. Yin J, Wang J (2014) A dirichlet multinomial mixture model-based approach for short text clustering. In: ACM international conference on knowledge discovery and data mining (KDD)
14. Nguyen DQ, Billingsley R, Du L, Johnson M (2015) Improving topic models with latent feature word representations. Trans Assoc Comput Linguist 3:299–313
15. Bengio Y, Ducharme R, Vincent P, Jauvin C (2003) A neural probabilistic language model. J Mach Learn Res 3(Feb):1137–1155
16. Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J (2013) Distributed representations of words and phrases and their compositionality. In: Advances in neural information processing systems (NIPS), pp. 3111–3119
17. Mikolov T, Chen K, Corrado G, Dean J (2013) Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781
18. Moody CE (2016) Mixing dirichlet topic models and word embeddings to make lda2vec. CoRR arXiv:1605.02019
19. Wang J, Chen L, Qin L, Wu X (2018) Astm: An attentional segmentation based topic model for short texts. In: IEEE international conference on data mining (ICDM)
20. Dieng AB, Ruiz FJ, Blei DM (2019) Topic modeling in embedding spaces. arXiv preprint arXiv:1907.04907
21. Dieng AB, Ruiz FJR, Blei DM (2019) The dynamic embedded topic model. CoRR arXiv:1907.05545
22. Zhao WX, Jiang J, Weng J, He J, Lim E-P, Yan H, Li X (2011) Comparing twitter and traditional media using topic models. In: European conference on information retrieval (ECIR)
23. Qiang J, Chen P, Wang T, Wu X (2016) Topic modeling over short texts by incorporating word embeddings. CoRR arXiv:1609.08496

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

24. Zuo Y, Wu J, Zhang H, Lin H, Wang F, Xu K, Xiong H (2016) Topic modeling of short texts: a pseudo-document view. In: International conference on knowledge discovery & data mining (KDD), pp. 2105–2114
25. Yan X, Guo J, Lan Y, Cheng X (2013) A biterm topic model for short texts. In: International conference on world wide web (WWW)
26. Miao Y, Yu L, Blunsom P (2016) Neural variational inference for text processing. In: international conference on machine learning (ICML), vol. 48, pp. 1727–1736
27. Gui L, Leng J, Pergola G, Zhou Y, Xu R, He Y (2019) Neural topic model with reinforcement learning. In: Conference on empirical methods in natural language processing and the international joint conference on natural language processing (EMNLP-IJCNLP), pp. 3478–3483
28. Wang R, Zhou D, He Y (2019) Atm: adversarial-neural topic model. Inf Process Manag 56(6):102098
29. Wang R, Hu X, Zhou D, He Y, Xiong Y, Ye C, Xu H (2020) Neural topic modeling with bidirectional adversarial training. arXiv preprint arXiv:2004.12331
30. Li X, Wang Y, Zhang A, Li C, Chi J, Ouyang J (2018) Filtering out the noise in short text topic modeling. Inf Sci 456:83–96
31. Shahnaz F, Berry MW, Pauca VP, Plemmons RJ (2006) Document clustering using nonnegative matrix factorization. Inf Process Manag 42:373–386
32. Kasiviswanathan SP, Melville P, Banerjee A, Sindhwani V (2011) Emerging topic detection using dictionary learning. In: ACM international conference on information and knowledge management
33. Yan X, Guo J, Liu S, Cheng X, Wang Y (2013) Learning topics in short texts by non-negative matrix factorization on term correlation matrix. In: SIAM international conference on data mining (SDM)
34. Cataldi M, Di Caro L, Schifanella C (2010) Emerging topic detection on twitter based on temporal and social terms evaluation. In: ACM KDD workshop on multimedia data mining
35. de Arruda HF, da Fontoura Costa L, Amancio DR (2016) Topic segmentation via community detection in complex networks. Chaos 26
36. McCallum AK (2002) Mallet: a machine learning for language toolkit
37. Lang K (1995) 20 Newsgroups Dataset. http://people.csail.mit.edu/jrennie/20Newsgroups/
38. Churchill R, Singh L (2021) textprep: a text preprocessing toolkit for topic modeling on social media data. In: International conference on data science, technology, and applications (DATA)
39. Pennington J, Socher R, Manning CD (2014) Glove: global vectors for word representation. In: Empirical methods in natural language processing (EMNLP), pp. 1532–1543
40. Lau JH, Newman D, Baldwin T (2014) Machine reading tea leaves: automatically evaluating topic coherence and topic model quality. In: Conference of the european chapter of the association for computational linguistics, pp. 530–539

Rob Churchill Rob Churchill is the Head of Data Science at 5-Out Solutions. He received his Ph.D. in Computer Science from Georgetown University in 2021, where is research focused on developing and applying topic models to social media data. He has authored a number of peer-reviewed papers on the subject of topic models.

Lisa Singh is the Director of the Massive Data Institute (MDI) and a Professor in the Department of Computer at Georgetown University. She has authored/co-authored over 90 peer reviewed publications and book chapters related to data-centric computing, i.e. data mining, data privacy, and data science, and is the co-author of Words That Matter: How News and Social Media Shaped the 2016 Presidential Election.