The Evolution of Topic Modeling

ROB CHURCHILL, Georgetown University
LISA SINGH, Georgetown University

Topic models have been applied to everything from books to newspapers to social media posts in an effort to identify the most prevalent themes of a text corpus. We provide an in-depth analysis of unsupervised topic models from their inception to today. We trace the origins of different types of contemporary topic models, beginning in the 1990s, and we compare their proposed algorithms, as well as their different evaluation approaches. Throughout, we also describe settings in which topic models have worked well and areas where new research is needed, setting the stage for the next generation of topic models.

CCS Concepts: • **General and reference** → **Surveys and overviews**; • **Computing methodologies** → *Modeling methodologies*; *Machine learning algorithms.*

Additional Key Words and Phrases: topic modeling, social media, online topic modeling, temporal topic modeling

1 INTRODUCTION

Understanding the core themes associated with a document collection is a fundamental task in today's information era. Topic models are a class of unsupervised machine learning techniques designed for this task. They can compress a corpus of thousands of documents into a short summary that captures the most prevalent subjects present in the corpus. The short summary takes the form of topics, or sets of related words, hence a topic model. We define a topic model to be an unsupervised mathematical model that takes as input a set of documents D, and returns a set of topics T that represent the content of D in an accurate and coherent manner.

Figure 1 shows an example of the kind of input (left), the output (center), and the usage (right) one would expect when employing topic modeling. One inputs documents. The topic modeling algorithm returns a set of themes/topics pertaining to those documents. Some algorithms also return a weight or degree of relevance for each word in the topic. The documents in the collection can then be labeled with these topics, allowing users to understand the importance of the topic in each document and in the collection as a whole. The example documents in Figure 1 are all books set in magical or medieval worlds, and as such, the four topics returned have to do with relevant magical themes described in these books. A topic model identifies themes by identifying repeated patterns of words and grouping these patterns of words into topics that reflect the content of the documents. It is not unusual for some topics to contain the same words (like "fairy" in the example). Intuitively, topics that are coherent, interpretable, and have a small number of overlapping words are considered higher quality. Topics do not usually come with labels as they do in Figure 1. These topic labels were added to make it easier to understand the distribution of the identified sets of words in documents, i.e. the document-topic distributions. This lack of a proper label can make their results very difficult for humans to interpret if topics are not coherent - containing sets of words that intuitively seem to belong together. Originally designed to classify documents like books, research papers, and news articles, seminal topic models rely on the predictable

Authors' addresses: Rob Churchill, rjc111@georgetown.edu, Georgetown University; Lisa Singh, lisa.singh@georgetown.edu, Georgetown University.

0360-0300/2022/1-ART1 $15.00
https://doi.org/10.1145/3507900

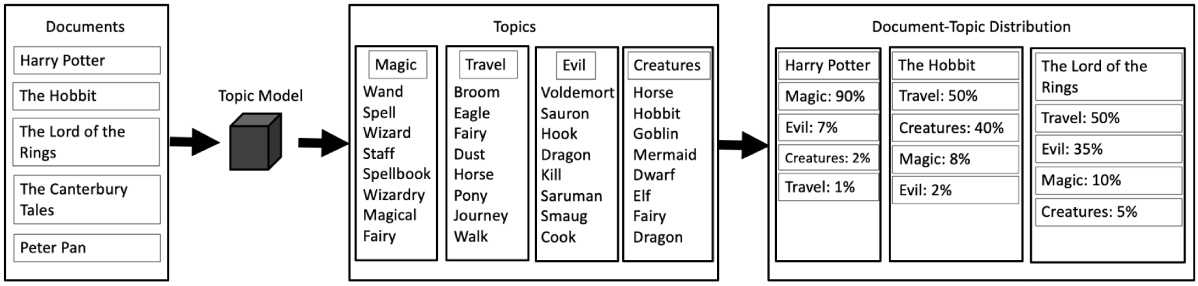


Fig. 1. An Example Input and Output of a Topic Model. Topic labels such as 'magic,' and 'travel' are not normally assigned by topic models, but are included here for clarity.

and frequent repetition of patterns of words across the document collection in order to produce accurate and coherent models. In other words, the more predictable the patterns, the better the quality of the topics generated by different topic models.

Topic models have been utilized across many disciplines to understand different types of texts – from tweets to books. Researchers have applied topic modeling to understand themes from historic newspapers [55, 90]. Sleeman et al. [71] incorporated topics models in their analysis of the evolution of climate change literature. Jocker and Mimno used topic modeling to identify themes in 19th century literature [34]. Ryan et al. [66] used topic models in a semi-iterative approach to discerning the most popular parenting topics on Twitter, while Bode et al. [11] employ a similar approach for understanding the 2016 US Presidential election using newspapers, tweets, and open-ended survey.

Topic models can also be incorporated with other learning models, to increase predictive power and interpretability. Singh et al. [70] use topics as input into hierarchical Bayesian models to help predict forced migration of displaced persons. Topic models have also been adapted to help predict the effects of genetic variants [3]. They have been used to generate tags for content tagging systems [37], to organize online recommendation systems [1], and to explain latent factors that lead to recommendations [65]. They have also been employed to improve sentiment analysis in text [35, 46]. Topic models are a useful tool in information retrieval, aiding in query expansion and document smoothing [92, 93]. What should be apparent from these examples is that topic models are being used broadly, both as a final result and as a feature into other predictive models.

As the number of applications of topic modeling increases, it has become clear that not all topic modeling algorithms are well suited for all types of text. Ryan et al. [66] use of an iterative manually-aided topic modeling approach to get an acceptable set of topics on social media data is indicative of the continued failure of topic models in that area. However, Sleeman et al. [71] demonstrate with scientific articles just how powerful topic models can be. This leads us to strive for that kind of power and accuracy in lower quality text like social media posts. Traditional books and scientific articles have certain types of discernible patterns of words, while social media posts may have very different ones. In this new world, social media posts are a new type of document and topic models can be important for understanding the themes of this online conversation. These new types of documents come with their own challenges. 1. *Document length*: Instead of well-edited books and articles, many text documents are short, unedited social media posts containing very few words. In the case of Twitter, tweets are restricted to 280 characters. 2. *Sparsity*: Instead of texts consisting of thousands of carefully edited words and phrases, social media posts contain a few dozen hastily typed words that are not grammatically consistent, often accompanied by a hashtag and a URL. Social media data has a vocabulary that is continually evolving. New words and hashtags are created constantly. Posts containing multiple languages are not uncommon, and

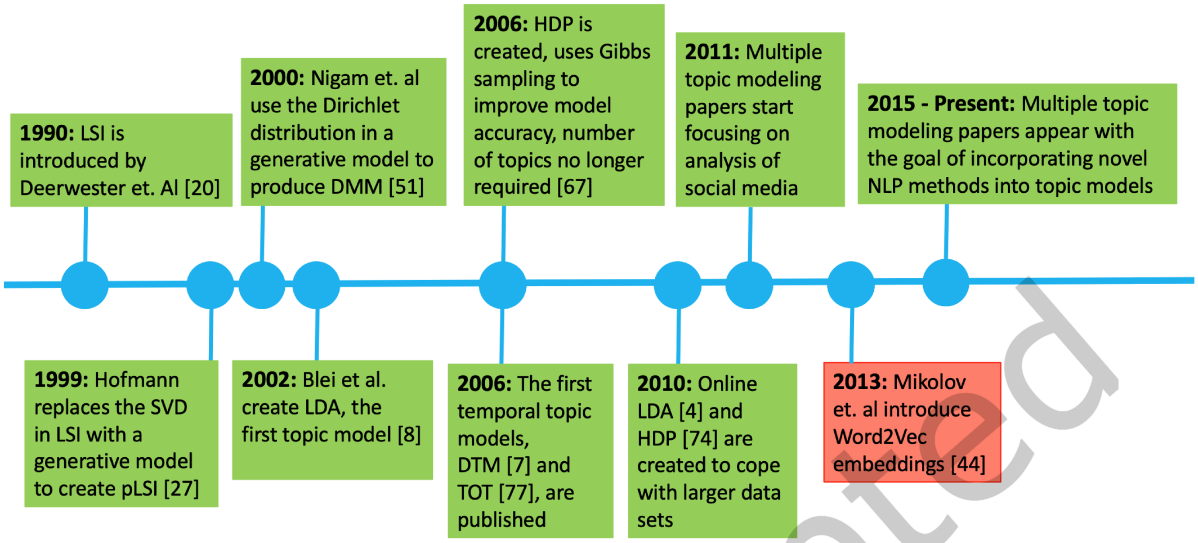


Fig. 2. A Timeline of Topic Models from Inception to the introduction of Modern Techniques

abbreviations are the norm. Reinforced word co-occurrence patterns within social media posts simply do not exist at the rate that they do in longer texts. This lack of strong word co-occurrence can result in noisy, incoherent topics when topic models not designed for sparse, short text are used with social media data. 3. *Volume*: Instead of the thousands of books and research papers being published every year, there are millions of social media posts generated every day. Many of the original topic models were built on statistical models supported by intractable inference problems that require relaxation just to be feasible for thousands of documents. 4. *Rate of Change*: Instead of information being published on a yearly, quarterly, monthly, or even on a weekly basis, social media facilitates the publishing of information in real time. This means that the subject of today was probably not that of yesterday, and will likely not be that of tomorrow. Given these new challenges, a topic model built for social media data should be capable of the following: a) building models in sparse domains with low word cofrequencies, b) considering the temporal dimension and adapting topics with time in mind, and c) filtering noise efficiently to produce coherent topics.

Due to the variety of document types, no single topic model can be expected to perform the best in every setting. This is the perfect time to look at the landscape of topic models and assess which are best for the new types of documents, and where there is still a need for improvement. For each topic model, we will highlight the strengths and weaknesses of the model, as well as the challenges addressed. This survey takes a timeline-like approach and looks at the evolution of *unsupervised* topic models in an effort to trace the origins of different aspects of current state-of-the-art topic models. We compare topic models based on their methodology, whether they are designed for static, temporal, or online document collections, the length of document they are designed for, the data they have been tested on, and the metrics used to evaluate them. Table 1 provides a quick reference for every model examined in this paper. For each model, the table indicates the base methodology of each model, whether it was designed for a specific setting, and whether it was designed for short (social media) or long (papers, articles) documents. Most topic models were designed for static data sets, but some were designed for online, temporal, or multiple settings. While the older models were designed prior to big innovations in natural language

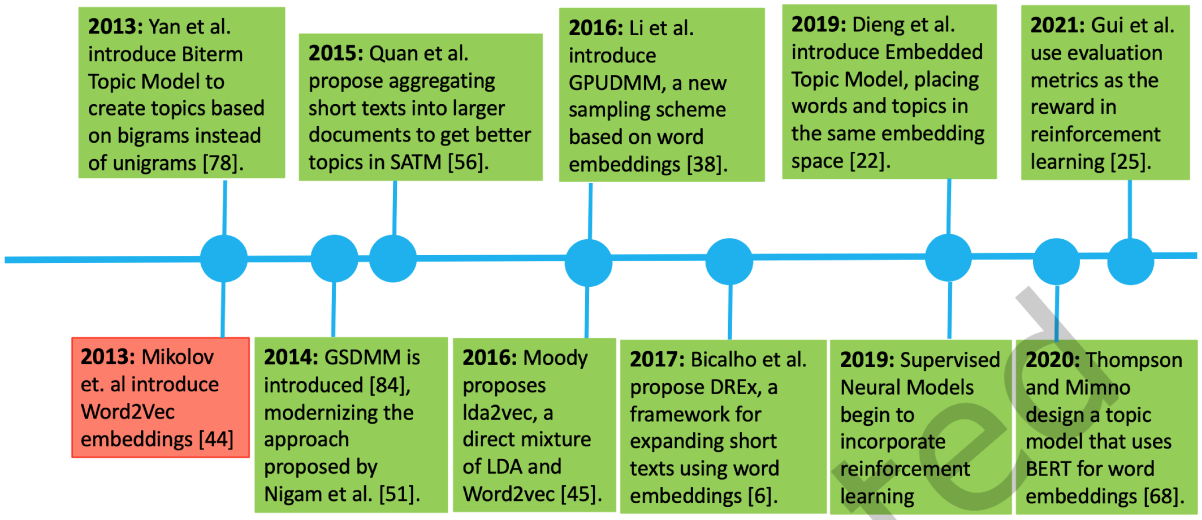


Fig. 3. A Timeline of Topic Models from the introduction of Word2Vec to Present

processing (NLP), the NLP-aided column shows just how many newer models have opted to take advantage of these new methods. In the last two columns, we delineate between long and short document types. While many of these models can be used in practice on all types of documents, we indicate here which types of documents these models were designed for and tested on in their respective papers.

We also discuss the evolution of the evaluation of topic models, which has been largely inconsistent over the history of topic models. Topic modeling is a subjective task, leaving much of the determination of the quality of topic models up to the humans who use topic models. Because evaluation is not done consistently for this task, we begin our survey by presenting the prevalent evaluation metrics in Section 2.

We then begin the history of topic models at the birth of unsupervised topic models and their early variants (Section 3). Figure 2 depicts the evolution of topic models from inception up until the incorporation of more modern language models such as word embedding spaces into topic models, and includes topic models from Sections 3 and 4. We describe some classical approaches to online and temporal topic models (Section 4), and move through time in order to understand how we arrive at where we are today (Section 5), and what there is still to be done in order to create accurate topic models (Section 6). Figure 3 depicts the evolution of topic models from the introduction of Word2Vec word embeddings to the present, and includes topic models from Section 5. In total, we compare over thirty topic models. While a number of supervised, semi-supervised, and topic models with knowledge have been proposed, this paper focuses on unsupervised models since they are more prevalent for large data sets. We briefly discuss other variants at the end of section 5.3. Ultimately, our goal is to provide insight and intuition about the basics of how each topic model is designed, and the strengths and weaknesses of the different topic modeling algorithms given the changing landscape of modern documents. Lastly, we discuss present day topic modeling, and what we believe are areas that still need attention.

2 EVALUATION METHODS

An important part of topic modeling is evaluation, both data sets and evaluation metrics. Traditionally, newspaper and research papers served as the popular sources of labeled data sets for topic model evaluation. The Twenty Newsgroups data set [41] is by far the most popular available data set for evaluating topic models. It contains just

Topic Model Comparison									
	Methodology				Designed For			Document Type	
Model	Generative	Graph-based	Matrix-based	NLP-aided	Static	Online	Temporal	Short	Long
LSI [23]			x		x				x
pLSI [30]	x				x				x
DMM [57]	x				x				x
LDA [8]	x				x				x
HDP [74]	x				x				x
CTM [40]	x				x				x
DTM [7]	x						x		x
TOT [86]	x						x		x
SWB [15]	x				x				x
Online LDA [4]	x				x	x			x
Online LDA [29]	x				x	x			x
Online HDP [83]	x				x	x			x
MTTM [52]	x						x		x
cDTM [82]	x						x		x
NMF [69]			x		x				x
NMF [36]			x				x	x	x
NMF [89]			x		x			x	
ETM [13]		x					x	x	
TS [22]		x			x			x	
TFM [21]		x				x	x	x	
PTM [18]		x			x			x	
BTM [88]	x			x	x			x	x
SATM [62]	x			x	x			x	
LF-DMM [56]	x			x	x			x	
LF-LDA [56]	x			x	x			x	
NVDM [49]	x			x	x				x
lda2vec [51]	x			x	x			x	
ETM [60]	x			x	x			x	
GSDMM [94]	x				x				x
GPUDMM [42]	x			x	x			x	
GPUPDMM [42]	x			x	x			x	
DREx [6]	x			x	x				x
CSTM [44]	x			x	x				x
WELDA [12]	x			x	x				x
LapDMM [45]	x			x	x			x	
CluWords [78]			x	x	x			x	
CluHTM [79]			x	x	x			x	
ETM [25]	x			x	x				x
D-ETM [26]	x			x			x		x
TND [20]	x			x	x			x	

Table 1. Topic Model Design Characteristics by Model. Ordered by appearance in this survey.

over 22,000 newspaper articles labeled with topics from a set of twenty different topics. Other newspaper article sources such as the New York Times and the Associated Press made for good labeled data sets since each article was associated with specific topics by the newspaper. Research conferences and journals, such as the Neural Information Processing Systems Conference (NIPS) and the journal *Science* were also labeled by authors. Since research has moved toward evaluating topic models on social media data, data privacy concerns have limited the sharing of these labeled data sets. However, Twitter allows researchers to download posts and many topic models are being tested on subsets of these data.

In the rest of this section, we describe some of the more popular evaluation methods for topic models. Different groups have conducted research on the effectiveness of certain evaluation methods [14, 54, 81]. We focus on those methods used by more than one approach we present, and organize them into three general categories: evaluation of coverage, evaluation of coherence, and qualitative evaluation.

2.1 Coverage

Coverage refers to how well the concepts in the document collection are represented. Coverage can be divided into two types of coverage, topic coverage and document coverage. Topic coverage measures assess how representative the topics themselves are, i.e. are the topics in a document collection identified by the model. The most prevalent topic coverage measure is *topic recall*. Topic recall is the fraction of ground truth topics recovered by the topic model.

Document coverage measures evaluate how well documents are represented by topics. Topic model *accuracy* is a typical measure for evaluating document coverage. It is defined as the fraction of documents that are accurately labeled by the topic model. In order to evaluate topic recall and accuracy, ground truth topics must be available. As will be seen, topic recall is used more sparingly than topic accuracy.

When a full set of ground truth topics are not readily available, other metrics are used. *Purity* is the accuracy of the model if documents are always assigned the dominant topic. This metric attempts to penalize models that assign a large number of low probability topics to documents, as opposed to a model that assigns a high probability to a single topic across the document collection.

Another way to measure convergence is to compare topics across topic models. *KL-Divergence* is used to show how a new model's probability distribution over topics differs from that of a baseline topic model. KL-Divergence is the expectation of the log difference between the underlying probability distributions of the topic models [38]. Instead of relying on ground truth topic sets, KL-Divergence allows one to compare the topic set of a new model to a previously established state of the art model.

All of the coverage methods described so far require, to some degree, prior knowledge topics, either in the form of ground truth topics or in the form of a state of the art topic model's topic distribution. In 1999, Hofmann took a different approach when he introduced *perplexity* [30]. Instead of relying on ground-truth data to compute topic or document coverage, perplexity is a probabilistic measure of how well a model can predict a held out sample of documents. It is a way to compare probabilistic topic models. More precisely, perplexity is the average negative log-likelihood of held-out documents. A lower perplexity means that the topic model does a better job of mapping the held-out documents to the theoretical distribution of words in the documents, thereby having better document coverage. Others have used the log-likelihood of held-out documents (not negated) as an evaluation method. It should be noted that it is debated whether or not perplexity is a reasonable evaluation method for determining topic quality. Change and colleagues showed that perplexity and human judgment were not well correlated [14].

2.2 Coherence

While coverage is an important measure for high quality topics, by itself, coverage is not sufficient for guaranteeing individual topic quality. To combat the potential for noisy and unintelligible topics, authors turn to what we call *coherence* evaluation methods. These methods attempt to discern the usefulness of individual topics, and of words in individual topics.

Many of the earlier works took a classic approach to coherence evaluation – *precision*. Given ground truth topics, and the top words from each approximated topic, a precision score can be calculated by taking the fraction of words from each approximated topic that falls into the best-fitting ground truth topic.

As we will see, coherence evaluation methods were largely abandoned after some of the initial topic models were introduced. Only recently has coherence resurfaced as an important component of topic model evaluation. The most common coherence metric is *pointwise mutual information (PMI)*. There are many different variants of PMI, but at its core, PMI attempts to measure the closeness of words in each topic based on their relative cofrequencies with each other. Its popularity has increased because unlike precision, no ground truth topic set is required to calculate PMI. A number of other coherence measures have been used to evaluate topic models. C_V, as defined by Röder et al. [64], is a combination of cosine similarity and normalized PMI under a boolean sliding window. It attempts to capture proximity between words as well as the mutual information and vector similarity.

Diversity is an evaluation method that attempts to account for word overlap in topics [25]. If a model cannot decide which topic a word belongs to, it may assign it with reasonably high probability to multiple topics. This makes a topic set less coherent. Topic diversity is the percentage of unique words in the topic set, considering the top-k words of each topic. Like PMI, diversity does not require knowledge of ground truth topics. Methods similar to topic diversity have appeared under different names over the years, including topic uniqueness [53] and topic overlap [21].

Adjusted Rand Index (ARI) measures the agreement of topic-document classifications. For a pair of documents, either they should or should not be clustered together based on some similarity measure. The Rand Index calculates the percentage of document pairs that are correctly placed together or correctly not placed together. ARI corrects the Rand Index for chance. ARI is most commonly used when the topic-documents labels are known, so that the similarity measure is whether the documents actually belong to the same topic. However, ARI could also employ a similarity measure that does not rely on ground truth labels.

Signal-to-Noise Ratio (SNR), similar to precision, uses ground truth topics to compare the number of correct and incorrect topic words in each approximated topic. A high SNR indicates more coherent topics, while a low SNR indicates high volumes of noise and therefore less coherent topics. Finally, *word intrusion*, defined by Chang et al. [14], is a human-judged coherence method that consists of giving a human six words, five from the same topic, and one that does not belong in the topic. If humans can consistently pick which word does not belong, then the topic model is judged to be more coherent.

2.3 Qualitative

While different coverage and coherence metrics give us insight into the quality of topic models, human evaluations capture insights that quantitative analysis cannot, including evaluating multiple criteria simultaneously and looking at the topics using a 'common sense' lens. As such, coverage and coherence evaluations in and of themselves are insufficient judges of a topic model. A qualitative evaluation of a topic model is any evaluation that displays the topics produced by a model for readers to assess themselves [8]. Qualitative evaluation consists of presenting topics generated during the experiments to human evaluators. In some cases, a qualitative evaluation is used to compare a single topic across all the models tested. For example, a human evaluator may be given a topic (or set of topics) output by multiple models and asked which one seems to capture the topic the best. Qualitative analyses can also be used to show the diversity of an entire topic set generated by the new model.

For example, a human evaluator may be asked to identify topics that are most similar or most different from a specific topic. In the case of some temporal models, a qualitative analysis shows topics along with when they are most prevalent along a time scale. These can then be mapped to relevant events to show the evolution of each topic. A qualitative evaluation of a topic model is used to portray what math and statistics often cannot: human understandability. A qualitative analysis can be misleading if authors choose to display only a small subset of the best topics. However, a well-done qualitative analysis can be very valuable if it helps researchers better understand the quality of topics generated and/or how a new model's topics might differ from baseline models.

2.4 Evaluation mapping to proposed models

Table 2 shows which models used which evaluation methods for their experiments. Table 2 contains only the evaluation methods that were defined and used in more than one paper examined in this survey. Many more evaluation methods have been used in the past to demonstrate the quality of topic models. While our list is not exhaustive, it does show a trend in methods over time. Coverage and coherence methods have waxed and waned in popularity throughout the years, with perplexity falling in favor of pointwise mutual information. The qualitative evaluation has become just as necessary as a good quantitative evaluation for portraying topic model quality. Other evaluation methods find a home here and there, but PMI and qualitative evaluation have become common practice.

3 EARLY TOPIC MODELS (1990-2006)

3.1 Precursors to Topic Models: Latent Semantic Indexing

Topic models can trace their origins back to 1990, as we can see in Figure 2. In their paper *Indexing by Latent Semantic Analysis*, Deerwester et. al described how one could use latent semantic analysis to automatically index and retrieve documents from large databases [23]. The authors devise a model called Latent Semantic Indexing (LSI).

Given a vocabulary, a set of documents can be represented by a word-document matrix, where each document is a vector the size of the vocabulary, and each entry in the vector is the frequency of the relative vocabulary word in the document. LSI takes the word-document matrix and performs a singular value decomposition (SVD) to reduce the dimensionality on the document-side of the matrix but retain the meaningfulness of the words. In doing so, LSI created the first topic sets from documents. These topics were vectors of word frequencies that were derived using SVD, and could be compared to the specific word frequency vectors of individual documents in order to classify the documents into topics. Notably, LSI defines what will come to be known as the bag of words model. The bag of words model is oblivious to the ordering of words in a document - it cares only about the frequency of words.

In 1999, Thomas Hofmann kicked off what turned out to be the field of topic modeling. Hofmann introduced Probabilistic Latent Semantic Indexing (pLSI), in an attempt to produce more consistently accurate results in domain-specific documents than LSI [30]. Hofmann never actually refers to topics in his paper, calling them *factors* instead. By replacing the SVD with a generative data model, called an *aspect model* in the paper, he was able to use an expectation maximization algorithm to train the model. Instead of topics being the result of an SVD, the topics were a probabilistic *mixture* of words based on their joint probabilities with documents.

3.2 DMM and LDA: The First Topic Models

3.2.1 DMM. Originally interested in improving the accuracy of text classifiers that until then had used labelled data to classify text, in 2000, Nigam, McCallum, Thrun, and Mitchell set out to determine how to incorporate unlabelled data into text classification [57]. They would again use expectation maximization along with a generative model, but they added the Dirichlet distribution to the conversation. The Dirichlet distribution is the

Evaluation Method Prevalence											
	Coverage						Coherence				Qualitative
Model	Recall	Accuracy	LL	Perplexity	KLD	Purity	Precision	PMI	Diversity	ARI	Qualitative
LSI [23]	x						x				
pLSI [30]				x			x				
DMM [57]	x						x				
LDA [8]				x							x
HDP [74]				x							x
CTM [40]			x	x							x
DTM [7]			x								x
TOT [86]		x			x						x
SWB [15]				x			x				x
Online LDA [4]								x			x
Online LDA [29]				x							
Online HDP [83]			x								x
MTTM [52]				x							x
cDTM [82]				x							x
NMF [69]		x									
NMF [36]	x	x					x				
NMF [89]		x						x		x	x
ETM [13]											x
TS [22]		x									
TFM [21]	x	x					x				x
PTM [18]								x	x		x
BTM [88]						x		x		x	x
SATM [62]		x				x		x			
LF-DMM [56]						x		x			x
LF-LDA [56]						x		x			x
NVDM [49]	x	x		x	x		x	x			x
lda2vec [51]											x
PTM [97]	x	x						x			x
ETM [60]								x			x
GSDMM [94]								x		x	
GPUDMM [42]		x						x			
GPUPDMM [42]		x						x			
DREx [6]	x						x	x			x
CSTM [44]		x				x		x			x
WELDA [12]								x			x
LapDMM [45]		x						x			
CluWords [78]								x			x
CluHTM [79]								x			
ETM [25]								x	x		x
D-ETM [26]								x	x		x
TND [20]	x							x	x		x

Table 2. Topic Model Evaluation Methods by Model. Ordered by appearance in this survey.

multivariate generalization of the Beta distribution. Instead of using just α and β to define its distribution (as in the Beta distribution), the Dirichlet distribution is defined with respect to k, where k can be viewed as the number of dimensions associated with the Dirichlet. Together, these dimensions are used to form a probability distribution that is normalized using α. The Dirichlet distribution is ideal for topic modeling because each topic can be cast as one of these k dimensions in the distribution. Intuitively, each of the k topics has its own probability of appearing given the distribution and observed document. In their algorithm, a document is sampled by one topic and generated according to that topic based on a prior distribution represented by a Dirichlet distribution. Their experiments show that while in some cases their model improves results over labelled data by 30%, the model can hurt performance in other cases. While originally not framed as a topic model, this model would come to be known in the topic modeling literature as the Dirichlet Multinomial Mixture (DMM) and the mixture of unigrams model.

3.2.2 LDA. The term *topic model* was coined in 2001 by Blei, Ng, and Jordan when they proposed Latent Dirichlet Allocation (LDA) [8] [9]. The authors identified problems with the pLSI model, including the large number of parameters needed to fine-tune pLSI, and the inability of one to query the pLSI model with new unseen documents. Furthermore, the authors of LDA made a key observation that enabled them to be more successful in finding accurate topics: documents are not generated according to only one topic.

LDA borrows from the ideas of pLSI [30] and creates its own generative model, this time based on the use of the Dirichlet distribution [57]. It uses the same bag of words model as pLSI, and the same document-term matrix defined by pLSI. The key innovation of LDA over DMM and pLSI is that LDA does not sample one topic per document. Instead, it samples a distribution of topics for a document, allowing a document to be probabilistically generated from many topics. LDA's goal is to find the parameters of a topic-word distribution that maximizes the likelihood of documents in the data set over k topics. Each document has a topic distribution specific to it that is proportional to the probability of each of its words in the topic-word distribution. LDA uses expectation maximization to train its model, and has two main parameters aside from k: α and β. α corresponds to topics per document ratio. Setting α higher results in more topics per document. β corresponds to words per topic ratio. Setting β lower results in fewer words per topic. For a set of M documents D, the generative process of LDA works as follows:

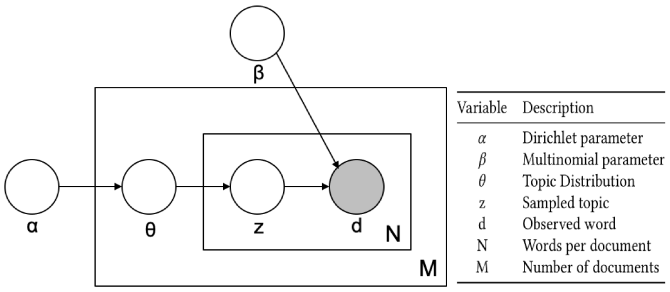
For $d \in D$:

(1) Randomly draw the number of words N for d.
(2) Randomly 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.

Figure 4 shows the graphical representation of LDA using the notation above.

The authors show that LDA performs better in terms of perplexity than pLSI. The main data set is a corpus of 16,000 newspaper articles from the Associated Press, containing a vocabulary of 23,075 terms. The authors identify a problem of term sparsity, writing, "the large vocabulary size that is characteristic of many document corpora creates serious problems of sparsity. A new document is very likely to contain words that did not appear in any of the documents in a training corpus" [9]. The authors attempt to account for this by using Laplace smoothing, essentially assigning some very low probability to every word so that it can be classified by the model. The problem of sparsity persists all these years later in even more extreme fashion due to the vast amount of slang and lack of structure in social media posts, open-ended survey responses, and other short texts.

LDA is still synonymous with topic modeling, and many other models to this day are variations of this model that was originally conceived twenty years ago. LDA has many variations and has been applied to many different



Variable	Description
α	Dirichlet parameter
β	Multinomial parameter
θ	Topic Distribution
z	Sampled topic
d	Observed word
N	Words per document
M	Number of documents

Fig. 4. LDA Graphical Model (Plate Notation).

tasks in and around the natural language processing realm. Jelodar et. al take an in-depth look at LDA and its variants in their survey of LDA-based topic models [33].

3.3 Early LDA Variants

In the years after designing LDA, researchers began to improve upon this model in order to solve some problems not addressed by LDA, including the selection of k, the topic independence assumption, and the dynamic nature of topics.

3.3.1 Hierarchical Dirichlet Process. One problem with LDA is that it has a parameter, k, representing the number of topics that should be sampled, that needs to be specified. This was a problem because most people do not know how many topics an unexplored data set contained. The process of finding the right k for a new data set was tedious, requiring users of LDA to test using numerous settings of k, manually evaluating the results after each iteration. Teh, Jordan, Beal, and Blei introduced the Hierarchical Dirichlet Process (HDP), a non-parametric adaptation of LDA that approximates the parameter k using a probabilistic model [74].

In HDP, the Dirichlet processes representing topic distributions are distributed according to another Dirichlet process. Essentially, not only are topics sampled probabilistically, the number of topics is also sampled probabilistically. HDP still allows for the fine-tuning of α and β in the same way that LDA does, but the k value is chosen probabilistically so as to take the tedious process out of the user's hands. Due to the extra layer of Dirichlet processes, HDP is even more intractable to compute precisely. The authors replace the deterministic expectation maximization algorithm from LDA with a Gibbs sampler to improve its efficiency.

In their experiments, the authors show that HDP provides topic sets with consistently low perplexity for a nematode biology abstract data set (5,838 abstracts), and a *Neural Information Processing Systems* (NIPS) articles data set (1,447 articles).

3.3.2 Correlated Topic Models. Another limitation of LDA is that it assumes independence on topics, where in the real world, topics are often correlated with each other. For instance, topics about weather and sailing are probably more correlated than topics about weather and video games. Blei and Lafferty introduce a correlated topic model (CTM) to address this issue [40].

The key difference between CTM and LDA is that in CTM, the authors replace the Dirichlet distribution with a logistic normal distribution, and add a covariance matrix between topics to model correlation. The generative process is identical to that of LDA; the only thing that changes is the distribution from which topic probabilities are drawn. The assumption that exists with a Dirichlet distribution - that topics are independent of each other - is not necessary in CTM, since the covariance matrix of the logistic normal distribution provides a measure of

how close topics are to each other. This allows for the model to pick not only words based on topics that are drawn from the observed document, but also from topics that are correlated with the probable topics.

To demonstrate the effectiveness of their approach, the authors perform topic modeling on articles from the journal *Science*. The domain of *Science* articles is spread across many fields of study, but intuitively there should be some correlation over many of these fields. Again, the authors use perplexity as their evaluation method. The authors show that the CTM is able to maintain strong performance even as the number of topics grows, whereas LDA falls off at around thirty topics.

3.3.3 Dynamic Topic Models. LDA captures the topics of a static data set at one moment in time, but it does not capture the evolution of topics across time. Blei and Lafferty create the first temporal topic model, called Dynamic Topic Model (DTM), with the goal of capturing *topic evolution* [7]. A *temporal topic model* is a topic model that adds a time-based component to topics. Instead of a topic consisting of a set of words, a temporal topic model produces topics that consist of a set of words and a timestamp or time range. A simple temporal topic model could separate documents into a range of dates and perform modeling on each subset of documents in turn.

The authors again use the journal *Science*, which has been published continuously since 1880. As technology and the knowledge of humans has advanced, so too have the topics covered in the journal, so the authors hypothesize that if DTM is a good model, it should be able to capture the evolution of topics throughout the history of *Science*, visualizing the birth and death of topics.

To create the dynamic topic model, the authors slice a data set into time periods, wherein only a given time period documents are exchangeable. The authors replace the Dirichlet distribution of topics with a sequence of Gaussian distributions, each of which represents the topic distribution for a different time slice. The authors replace the Dirichlet prior α with a Gaussian prior γ in each time period. The authors state that their Gaussian distributions are an extension of the logistic normal distribution to deal with time series data. The approach that the authors use is not so much a temporal topic model as it is a series of topic models tied together in sequence. The topic distribution and γ prior are passed on to the next time period as a starting point. The topics for time slice t are conditioned on the topics from time slice $t-1$, and by the documents in time slice t. This leads to topics that seemingly evolve through time, appearing as they become relevant and disappearing as they lose popularity.

The authors show how topics related to Darwin and Einstein spike as each become famous for their work, and slowly die out as time passes on. Another example that they point out in the paper is the topic about the moon, which is moderately popular throughout the late 1800s and early 1900s, but which spikes during the space race of the Cold War. The authors further show the ability of DTM to predict future topics given its current topic distribution. They compare this predictive power to the approach of chaining LDA through the same time slices, and show that DTM is far more accurate when tasked with predicting future topics.

In 2006, Wang and McCallum proposed a temporal topic model based on LDA that works on a continuous-time model [86]. The advantage of this approach, called Topics over Time (TOT), is that events are not uniformly distributed over time, so uniformly distributed time epochs do not fully capture the evolution of topics over time. TOT models time alongside word co-occurrence in order to capture time-sensitive events, such as different wars and the rise of technology. The continuous distribution over time allows for the user to not specify the length of an epoch, instead relying on the algorithm to find significant word co-occurrences over relevant time periods.

Wang and McCallum mostly provide a qualitative analysis of their topic model, but perform a small quantitative analysis of their model compared to LDA, where they use KL-divergence to show that TOT produces slightly more distinct topics than LDA. They tested TOT on three different data sets, one containing 208 presidential State-of-the-Union addresses, one containing 13,300 emails between computer science researchers, and the final consisting of 2,326 research papers from the NIPS conference between 1987 and 2003.

DTM and TOT represent the first attempts at applying LDA in a temporal fashion in order to capture the evolution of topics.

3.4 Topic models considering background noise

In 2006, Chemudugunta et al. [15] proposed a topic model, SWB, that incorporated two secondary distributions to capture special words (specific to a document) and background words (what we would call noise words). The goal of the proposed method is to capture the difference between generic documents, highly specific documents, and those in between. In theory, this should produce documents that are true to the approximated topics, but that also vary in specificity as is the case in the real world. They use LDA [9] as the basis of their model, and insert the special words and background words distributions directly into the scheme. The authors propose using a random variable to act as a switch, which determines if the next word in a document is generated from the special words distribution, the background distribution, or the chosen topic. The switch variable is sampled from a document-specific multinomial conditioned on a Dirichlet prior, so that every document is individually tailored. The special words distribution is sampled from a document-specific multinomial conditioned on another Dirichlet prior, such that each document has its own set of special words. The background words distribution is sampled from a data set-specific multinomial conditioned on another Dirichlet prior. This final distribution is data set-wide so that commonly used words can be identified across all documents.

In their experiments, the authors show the effectiveness of the background distribution on four different data sets, and compare their model variants to LDA using perplexity and precision. The four data sets used in the experiments are of unspecified size, but consist of NIPS articles, Associated Press articles, U.S. Patents, and Federal Register articles, respectively. The authors show that their model variants beat LDA in terms of perplexity, and beat LDA [9] and LSI [23] in terms of precision when tasked with retrieving documents containing at least one query word (a downstream information retrieval task). The authors use a newspaper data set consisting of 3,104 articles about Enron from the New York Times to perform a qualitative analysis, highlighting topic words, special words, and background words.

While the authors clearly intended their model be used for information retrieval purposes, their proposal is the first real attempt to filter noise words from topics. We believe that they were ahead of their time in this regard. As we will see, identifying and filtering noise in topic models becomes crucial as the type of documents that we care about transitions from long, well-edited documents to short, noisy ones.

3.5 Computational Complexity of Generative Models

The early implementations of LDA and its descendants were very computationally expensive. Sontag and Roy proved the complexity bounds of generative topic models [73]. They showed that if a document is generated by a mixture of the entire topic set, inference is NP-Hard. However, if a document is generated by a constant number of topics strictly smaller than k, inference can be solved in $O(N^4)$, where N is the number of documents in the data set. These complexity bounds severely limit how fast topics can be approximated in some of the most popular topic models. While approximation algorithms do exist, improving the efficiency of generative topic models is still an important research direction.

4 TEMPORAL AND ONLINE TOPIC MODELS (2006-2011)

4.1 Temporal Topic Models

As discussed in Section 3, by 2006, Blei and Lafferty [7], as well as Wang and McCallum [86], had already been experimenting with ways of detecting topics as they rise and fall over the course of time. While their approaches were relatively successful, they were computationally costly.

In 2007, Nallapati et al. proposed a new generative model, MTTM, to improve on the Dynamic Topic Model's temporal aspect [52]. The authors replace LDA's multinomial distribution over words with a Poisson distribution over the words. For each word in each epoch, an expected frequency in each topic is kept in the form of a Poisson mean. The authors represent each topic as a vector of Poisson means, which allows for the evolution of topics

over time epochs to be monitored more efficiently. The authors show that MTTM works in a hierarchical manner, starting with the smallest time epochs, and building up into bigger and bigger epochs. This hierarchical structure allows for users to "zoom in" on certain time periods for a particular topic.

In 2008, Wang et al. proposed cDTM [82], a continuous version of the DTM. In it, as in TOT, cDTM does not require time to be discretized, opting for a continuous distribution that is less computationally intensive than the original DTM. The authors first probabilistically select a time frame for each topic before inferring the word distribution. This approach allows for faster inference of a more fine-grained timeline of topics.

In 2009, Iwata et al. proposed Topic Tracking Model (TTM) [32], a model created for analyzing consumer purchase behavior on e-commerce websites. It borrows the idea of passing Dirichlet priors to future time periods from DTM [7], but changes how it occurs. The authors attempt to model the items that a user buys at each time period as topics, in order to inform future time periods of items likely to be bought together. In the abstract concept of topic models, users in their scheme represent documents, and the items that a user buys represent individual words in a document. Because it was designed to track the trends of consumers, they pass user-specific priors into future time periods. Whereas in DTM the topic distribution and global γ prior are passed onto the next time period, TTM passes on the topic distribution and a specific distribution for each user representing their interests, i.e. their frequently used words. This effectively allows the model to track not only the evolution of topics throughout time, but the individual evolution of users throughout time. The authors also allow for 'long-term dependencies' to be passed on, where instead of just passing the topic distribution and interests from the previous time period, those of the past L time periods are passed along. These dependencies are included with the acknowledgment that users' future interests are not solely derived from the most recent time period.

The authors test TTM against variations of LDA, on two data sets of transactions. The data sets contain transactions from movie and cartoon downloading websites, respectively. The former contains 70,122 users, and the latter contains 143,212 users. To show the capabilities of TTM, they show the N-best accuracy of TTM and the baseline models, which measures the percentage that purchased items are contained in the set of N-highest probability items. The accuracies of the movie data set are low for every model, but TTM beats all LDA variants on both data sets. The authors also show that TTM's runtime is much faster than the slowest LDA variant, and only 30-40% slower than the fastest LDA variants (including Banerjee and Basu's version of online LDA [4]).

This is a novel implementation of a topic model in a temporal setting, and although it moves away from the traditional use of topic models on documents, it presents a framework for working with dynamic topic models. If we view each post in a Reddit thread as a transaction, then this approach could be a foundation for temporal modeling in social media data.

4.2 Online Topic Models

These early models also relied on a fixed vocabulary that was determined during initialization of the model. However, there are times when new documents are added that contain some new words. In this section, we explore how LDA and HDP were transformed from the original batch-based Bayesian inference to online versions that require fewer computations, and allow for additional vocabulary to be added after initialization. This can occur if the data set is too large to load into memory all at once, or if the data set is being continuously generated (streaming setting), such as from a streaming API or RSS feed.

4.2.1 Faster Topic Model Inference. In 2008, Porteous et al. introduced a faster implementation of a Gibbs sampler for LDA that was up to eight times faster than the original [59]. The main contribution of Porteous et al. is that instead of inferring a document's probability of being in every topic, they infer only the probability of a document being in a few of the most likely topics for that document. This saves a lot of inference time in large data sets, and the authors show that in two large data sets with large numbers of topics, the top twenty topics for a given document explain about 90% of that document. The authors tested their models on four data sets. The

first contained 1,500 papers from the NIPS conference, the second contained 39,861 emails from Enron, the third 300,000 news articles from the New York Times, and the final contained 8,200,000 abstracts from the PubMed journal. The authors show that on the smaller data sets, the speedup of their model is considerable over LDA, but that the effect is diminished in the larger data sets.

In 2009, Yao et al. described multiple approaches to making the inference of LDA faster while retaining accuracy [91]. The authors introduce three different variations of Gibbs sampling that improve overall inference speed. All methods assume that inference is being performed on smaller chunks of data, adding more chunks in each iteration. The first method resamples all topics over all documents (new and old) each iteration. This is faster than normal Gibbs sampling because it is steadily increasing the number of documents per iteration instead of inferring the entire data set every iteration. The second method resamples topics only for new documents. This is much faster than the previous method, because it fixes the topic assignments of previously seen documents. It uses those topic assignments as training data for assignments of topics to future documents. The third method is an online version of Gibbs sampling, which independently processes each document. The topic-word distribution used draws only from the current document and the training data. They also propose a logistic regression classifier and Naive Bayes classifier as other means of performing inference. The authors perform experiments on relatively small data sets (1,740 NIPS papers and 51,616 paper abstracts). Yao and colleagues show that the online versions of their Gibbs sampler converge much faster than the traditional Gibbs samplers on these small data sets. Although their versions are faster, the authors note that each individual iteration of inference is much longer than those of the traditional Gibbs samplers.

4.2.2 Online LDA. In 2007, Banerjee and Basu [4] wrote a short survey of topic models for text streams. They implement batch and online topic models, and present the first online scheme for LDA. Their approach is straightforward, running LDA [9] on small batches of documents in a sliding window format. The topic distribution and Dirichlet priors are initialized for the next batch using the approximations from the previous batch. The authors show that their version of online LDA is much faster than the batch version. They also test their models using normalized pointwise mutual information, the earliest occurrence that we have seen of its use in topic model evaluation. They use the full Twenty Newsgroups data set [41], which contains approximately 20,000 news articles, and five small subsets of the data set to test the models.

In 2010, Hoffman et al. developed an online variational Bayes (VB) algorithm to allow for faster inference of topics using LDA [29]. The authors created the online VB algorithm by computing an approximate expectation step based on the current observed document and the prior topic-word distribution. The authors compute the approximate distribution, assuming that the entire corpus consists of this one document. They then adjust the true topic-word distribution to a weighted average between the previous distribution and the approximated distribution. This new online version relies only on the current observed document, and the approximated topic-word distribution from all previously observed documents. In implementation, instead of observing one document at a time, the authors observe "mini-batches," or small collections of a handful of documents. This allows for more accurate, but still fast, inference of topics.

In their experiments, the authors show that increasing batch size from one up to 256 documents results in better accuracy in terms of perplexity. The authors show that on a corpus of 3.3 million Wikipedia articles, it took three days to complete, with about half of that time devoted to the inference (preprocessing accounted for the other half). They note that even one iteration of the batch version of the algorithm would have taken many days to complete. The authors do not provide specifications for the machine that experiments were performed on, so it is not possible to say how long such experiments would take today. However, for the sake of comparison between online and batch LDA, it seems safe to say that online is considerably faster.

4.2.3 Online HDP. In 2011, Wang et al. [83] developed an online variational Bayes algorithm for HDP. The authors apply the same approach, using mini-batches of documents to produce approximations of the expectation step of the expectation maximization algorithm.

Due to HDP's inherently more complex computational nature (it infers the number of topics as well), the authors used much smaller data sets in their evaluation of their online HDP variant. The authors showed results for 352,549 *Nature* articles and 82,519 *PNAS* documents. Wang and colleagues note that traditional HDP can only be run on batches of 20,000 documents due to time constraints. The authors change the experiment setup from that of online LDA, choosing to run each model for six hours, instead of running models to convergence or completion. They show that online HDP produces a higher accuracy after six hours of simulation. Like in the experiments for online LDA, Wang et al. do not provide machine specifications for their experiments. Assuming all models were run on the same caliber machine, online HDP outperforms online LDA and batch HDP significantly on the two test data sets.

While these online algorithms are an important step in the evolution of topic models, they are still limited by the number of documents they can handle. In the modern age of social media, where millions of documents are produced every hour, these online methods may not be fast enough.

5 MODERN TOPIC MODELS (2011-PRESENT)

In the last twenty years, the documents that we input into topic models have changed far more drastically than the topic models themselves. At the turn of the century, we input scientific articles, books, and newspaper articles. Now, we input social media posts including tweets, blog posts, Reddit posts, and other short texts. Topic modeling research has evolved to address the problems that arise when attempting to infer topics on these new types of data. There are a few new approaches, but many of the old mathematical components from twenty years ago are still the accepted best practices. In each of the following subsections, the first model is the seminal model for the category of models, and we explain it in depth, followed by descriptions of other models in the category.

Figures 2 and 3 depict the bigger developments in topic modeling's thirty year history. In this section, we cover the most recent developments that have impacted the field of topic modeling — the application of topic models to social media data, the invention of Word2Vec and other word embedding models, as well as the incorporation of novel natural language processing techniques into topic models.

5.1 Non-negative Matrix Factorization

Non-negative matrix factorization (NMF) is a mathematical process in which a matrix of non-negative values is factorized into two new matrices such that the product of the two new matrices is equal to the original. Like the inference of generative models, NMF is known to be NP-Hard [77]. The two factor matrices have a much smaller dimensionality than the original matrix. In the case of topic modeling, the large matrix is the document-word matrix. In other words, the large matrix is the set of documents, where each document is represented as a vector of words. The two smaller matrices that NMF factors the large matrix into are the topic-word matrix and topic-document matrix. This topic-word matrix can be interpreted as a topic distribution over words, and topics can be derived from it by choosing the highest weighted words for each topic. Likewise, the topic-document matrix can be interpreted as a topic distribution over documents, and each document's individual topic distribution can be derived from this matrix. A k value representing the number of topics desired by the user decides the final dimensionality of the smaller matrices. If there are D documents and V words in the vocabulary, the large matrix will be of size DxV, and the smaller matrices will be of size kxV and Dxk, respectively. The setup of the problem is simple, but solving it exactly is intractable, so an approximation algorithm is used. NMF works by minimizing an error function between the original matrix and the product of the two factor matrices. Non-negative matrix

factorization has been shown to be mathematically equivalent to pLSI (described in Section 3) when using KL-divergence as the error function [27].

In 2006, Shahnaz et al. demonstrated NMF's capabilities as a topic model [69]. The authors use a sparse encoding of the document-term matrix to save memory and run the algorithm more efficiently. The experiments consisted of two data sets, one with 21,578 Reuters news articles, and the other containing 7,919 transcripts from different news sources. The authors limit their experiments to documents pre-labeled with a single topic. In the first data set, the accuracy of the model ranges from 96% when they label documents using only two topics, to 54% when labeling twenty topics. The second data set ranges from 99% accuracy at two topics to 80% at twenty topics. The authors attribute the disparity in performance to the broad topic labels in the Reuters data set.

In 2011, Kasiviswanathan et al. published a temporal model that used NMF as its basis [36]. While NMF was not designed specifically for a streaming setting, the authors used NMF on documents in each epoch in order to produce sets of *emerging topics*. Emerging topics are temporal topics that are receiving more attention, i.e. rising. The authors tested their algorithm on a set of 9,394 documents, and on the Twenty Newsgroups data set, consisting of 18,774 news articles. Instead of other topic models, the authors compared their model against clustering algorithms. The authors order the documents not by date but by pre-labeled topic, and introduce documents in order of their clusters with some overlap in each epoch. While the model itself is unsupervised, the ordering of the documents in their experiments constitutes some level of supervision. In most settings, data are not generated in order of their pre-labeled topic. This potentially gives an unfair accuracy advantage to the model. After testing on the two pre-labeled data sets, the authors also test their model on a set of 5,199 tweets about the performance of IBM's Watson on Jeopardy. The authors ran the model on two weeks' worth of tweets leading up to the airing of the show (8,434 tweets) as a baseline for finding emerging topics.

In 2013, Yan et al. took a new perspective of the NMF approach [89]. Instead of attempting to detect topics using a document-term matrix, they employ a term correlation matrix in order to detect topics in short texts. They argue that word co-occurrence within documents is not adequate for detecting topics in short texts due to the low frequency of words within these documents, and that the true nature of topics in these texts can be captured by analyzing the correlation between words themselves. Instead of building a matrix of documents with word counts, the authors build a matrix of words with positive pointwise mutual information scores (positive to maintain non-negative factorization). The authors employ NMF on this matrix in order to learn the topics directly from word correlations rather than from document correlations. The authors test their algorithm on three data sets consisting of 4,520 tweets, 2,630 news article titles, and 36,219 online questions, respectively. The authors compared their algorithm to LDA and other implementations of NMF for topic modeling. The authors evaluate topic sets based on purity, normalized PMI and ARI.

NMF and its variants are a dimensionality reduction approach to topic modeling. In environments where noise is pervasive, dimensionality reduction methods that are able to remove noise and extract features from sparse high dimensional spaces may be promising.

5.2 Graph-based Models

Recently, some work has focused on finding topics using graph-based approaches. When building the graph, words are generally nodes and their co-occurrence is represented as a weighted edge. The basic intuition is that words in tightly connected communities represent topics.

An early implementation of a graph-based topic model is that of Cataldi et al. [13], a model designed to detect emerging topics. In their model, the authors employ a directed graph, where each node is a word, and each edge is a weighted co-occurrence of two words. Edges are directed because the weight can be different in each direction. Edge weighting is calculated based on the co-occurrence frequency of a word pair, relative to the total frequency

of the word at the source of the edge. A word with very high frequency, but many co-occurring words, is likely to have lower outgoing edge weights than one with a lower frequency but fewer co-occurring words.

The graph is created for each time epoch in sequence, and the frequency of words in each epoch are saved for the future epochs. Once the graph has been created, a localized edge pruning is performed, removing lower-weight edges. Due to the necessity of pruning edges, this algorithm must iterate through all edges in the graph. The number of possible edges is $\frac{|V|(|V|-1)}{2}$, where $|V|$ is the number of words in the vocabulary, so the complexity of the algorithm is $O(|V|^2)$, although due to graph sparsity, the runtime is usually much faster. Topics are detected on this graph by running a depth-first search (DFS) out from a set of "emerging terms." To compute emerging terms in each epoch, frequencies from previous epochs are compared to current frequencies. Words are chosen as emerging terms if frequency has increased significantly. From each term reached during the DFS, a new DFS is run. If a DFS reaches the original emerging term, then the term is added to a topic containing the emerging term. This process also guarantees that no two topics will include the same term, resulting in zero topic overlap. This double-DFS approach to finding topics in a directed graph ensures that any two words included in the same topic are strongly connected to each other, at least through some chain of other strongly connected words.

The authors show their results on a generic Twitter data set containing more than 3 million tweets over two weeks in 2010. While they do not compare their accuracy to other topic models, they show topics that are recovered throughout the two week period. Their topics are small (four words each), but there is no analysis that shows whether the approach captures all emerging topics, or just a handful of the biggest.

In 2016, Arruda et al. designed a static topic model, topic segmentation (TS), based on an undirected graph with nouns and verbs as nodes, connected by co-occurrence [22]. The authors describe three different ways to compute co-occurrence, based on adjacency of words, paragraph co-occurrence, and an approach that compares co-occurrences to a baseline frequency. The authors employ the Louvain modularity algorithm to detect communities of connected words in their graph [10]. The complexity of the modularity algorithm is $O(nlogn)$. Given a subset of nodes in a graph, the Louvain modularity algorithm compares the number of internal edges (between nodes in the subset) to the number of outgoing edges (between a node in the subset and a node not in the subset). A higher than expected ratio of internal edges to outgoing edges indicates a high modularity score. A high modularity score, in the approach designed by Arruda et al., indicates a probable topic represented in the graph. The authors created their own data set, consisting of 200 documents built from paragraphs extracted from Wikipedia articles revolving around pre-chosen topics. The authors compare their approach to clustering methods instead of topic models.

In 2018, Churchill et al. [21] designed Topic Flow Model (TFM), with the goal of reducing noise in topics in a temporal setting, and tracking the evolution of topics. The authors employed a similar directed graph structure as Cataldi et al. [13]. For the same reasons as [13], the complexity of TFM is $O(|V|^2)$. Instead of performing a local edge pruning method, the authors include an edge only if the weight meets a global threshold. This results in fewer edges in the graph, which allows for faster modeling. The authors also introduce the notion of domain-specific *flood words*. Flood words are words that appear in documents across a domain, and like stopwords, do not add context to the topics. For example, if we were trying to understand COVID-19 relevant topics, 'COVID-19' would be in every topic, making it a flood word. To account for flood words, the authors introduce an upper bound for word frequency when detecting emerging terms. A third parameter captures the individual change in frequency over time for possible emerging terms. While Catadi et al. [13] traversed the graph using a DFS in order to find topics, Churchill et al. [21] used a breadth-first search (BFS), and limited the distance of the search. While this more effectively filters out noisier emerging terms from topics, more parameters need to be tuned.

The authors use the emerging topics detected at each time epoch to track topics through time. In addition to the emerging terms detected at each epoch, TFM seeks to confirm the continuing existence of previous topics. If a previous topic can be recovered in the current epoch, it is included in its new state. In this manner, the

evolution of a topic can be tracked from emergence to disappearance. In their experiments, the authors used a small synthetic data set consisting of 500 documents randomly generated, consisting of seven topics, seven time periods, and varying levels of flood words. The authors also tested their model at larger scale, using a set of 280,000 tweets about the 2016 U.S. Presidential Election, and a set of 14,000 newspaper articles about the same topic. Precision, recall, and signal-to-noise ratio was used to evaluate TFM. Using ground-truth labeled topics, TFM was found to be better at detecting topics in a temporal setting than the baselines.

In 2020, Churchill and Singh proposed another graph-based topic model, Percolation-based Topic Model (PTM) [18], for detecting topics in noisy data sets. Their approach consisted of incrementally stripping noise away from the graph, leaving small topic kernels containing little to no noise. Churchill and Singh relaxed the clique percolation problem[1] to allow for subgraphs (topic kernels in their case) to be grouped together. In their experiments, the authors used large, domain-specific Twitter data sets consisting of more than 750,000 tweets per data set. The authors used topic diversity and topic coherence (NPMI) to evaluate the quantitative quality of their models, showing competitive scores to the best models. A qualitative analysis compares their models to state of the art models in terms of interpretability and noise. Due to having to iterate through all edges in each round of decomposition, with the possibility of having to perform $|V|$ rounds of decomposition, the complexity of PTM is $O(|V|^3)$, where $|V|$ is the number of words in the vocabulary.

Graph-based approaches add a new perspective to topic modeling, taking a different direction from the earlier generative models. They do not assume knowledge of the underlying topic distribution, so they more readily find topics of varying size. In some sense, they can be more adaptive than generative models, allowing for either a global or local topic discovery. However, if flood words are not well managed, the graph structure can be very costly to traverse.

5.3 Mixing Traditional Topic Models with Modern NLP Methods

In recent years, there has been a shift toward incorporating more sophisticated natural language processing techniques into topic models. While topic models can be viewed as NLP models, here we consider how other NLP models augment traditional topic models. In contrast to early models such as LDA, which are statistical models, these new approaches leverage prior knowledge of natural language in the form of other pre-trained NLP models in order to improve the coherence and accuracy of the unsupervised topic model. Figure 3 outlines a few of the most notable contributions discussed in this section.

The most popular form of modern NLP model that has been incorporated into topic models is word embedding spaces. A survey of word embeddings by Almeida and Xexeo [2] gives an in-depth account of their development. Here, we focus on the inception of large-scale word embeddings to provide context for the topic models that follow. In 2003, Bengio et al. described how a neural model could be used to learn a distributed representation for words [5]. They named these distributed representations word feature vectors, which have come to be known as word embeddings. In 2013, Mikolov et al. introduced a model called Word2Vec for creating word embeddings that were both fast to compute and accurate [50]. They showed how their word vectors could be used to find semantically similar words. This simplicity lends a natural hand to topic models, which are tasked with finding groups of semantically similar words that represent a topic.

5.3.1 Biterm Topic Model. In the same year that Mikolov and colleagues introduced improved word embeddings [50], Yan et al. released the biterm topic model (BTM) [88]. Designed for short texts such as tweets and other social media posts, the biterm topic model attempts to model the word co-occurrence patterns present in a data set instead of document-level patterns. This approach makes sense in the context of short texts because document-level patterns are harder to track in documents that contain only a small number of words. The authors

[1]Clique percolation is a way of clustering size-k cliques in a graph. It consists of grouping two size-k cliques if they share k-1 nodes. This process is repeated for all pairs of size-k cliques.

extract word pairs that occur in the same document and perform inference on the word pairs instead of the documents. Biterms, as they call word pairs, are generated based on a topic-word distribution. Their generative process is dependent not on documents, but on these biterms. To approximate the biterm distribution, the authors employ a Gibbs sampling algorithm not unlike that of LDA or DMM, where each biterm is generated by some topic with some probability.

Yan and colleagues show that iterations of BTM using Gibbs sampling take significantly longer than those of LDA since the set of biterms can be significantly larger than the set of documents. On the other hand, BTM requires less memory due to the actual size of each biterm compared to that of each document. The authors performed experiments on the Tweets2011 collection, which contains 4.2 million tweets with a vocabulary of 98,857 words. They also use a data set of questions from Baidu, a Chinese Q&A website, containing over 648,000 questions. The authors show that on both data sets, their model outperforms LDA in terms of coherence and classification accuracy.

5.3.2 Self-Aggregating Topic Model . In 2015, Quan et al. introduced the Self-Aggregating Topic Model (SATM), to attempt to improve topic modeling on short texts [62]. The authors identify the problem of short texts lacking adequate word co-occurrence for use in traditional topic models, and attempt to solve it by aggregating short texts into longer texts that are better suited for use in LDA. In their model, the authors employ a two step process, running LDA on the data set of short texts, and using the output topics to probabilistically generate longer pseudo-texts. A pseudo-text is an aggregation of shorter documents into a single longer document for the purposes of increasing word co-occurrence. For instance, two similar documents, 'Medical Facts: Coronavirus makes your teeth fall out,' and 'Medical Facts: Coronavirus is the number one cause of cavities and root canals,' could be combined into a pseudo-text 'Medical Facts: Coronavirus makes your teeth fall out Medical Facts: Coronavirus is the number one cause of cavities and root canals.' [2] These pseudo-texts are used to generate short text snippets that represent topics, such that each snippet is generated according to one pseudo-text. These text snippets are then output as topics.

The authors test their model on 1,740 NIPS papers and 88,120 questions from Yahoo Answers. They use a PMI score to evaluate the coherence of their model compared to a set of hand-labeled, gold standard, topics for each data set. SATM achieves a higher purity score than LDA, DMM, and BTM, but no comparison in terms of coherence between SATM and the baselines is offered.

5.3.3 Latent Feature LDA and Latent Feature DMM. In 2015, Nguyen et al. [56] proposed using word embedding spaces (latent features) in conjunction with traditional topic models LDA and DMM. They replace the topic-word distribution of LDA and DMM with a two-component mixture of a topic-word distribution and a latent feature component. What this means is that LF-LDA and LF-DMM retain the structure of their traditional counterparts, while adding latent feature vectors for each word in the distribution. In generating a word for a document, the word is either chosen from the drawn topic, or from the latent feature vector of the topic, in essence expanding the pool of words to be chosen by allowing words similar to the topic in the latent feature space to be added as well. Because of the added latent feature space, the authors expect their models to perform well on data sets where there is little data about topic-word distributions, i.e. short texts.

The authors test their model on different variants of the Twenty Newsgroups data set [41] containing 18,820 documents (variants contained 1,794 and 400 documents respectively), the TagMyNews data set [80] containing 32,597 documents, and a Twitter corpus containing 2,520 documents. They use NPMI to evaluate the quality of their models using Word2Vec [50] and GloVe [58] as the latent feature spaces. They show that at least one of their models beat vanilla LDA in terms of NPMI and purity on each data set. A qualitative analysis shows how their model successfully groups together topics over iterations.

[2]These 'facts' are *not* true.

5.3.4 Neural Variational Document Model. While not technically designed to be a topic model, Miao, Yu, and Blunsom, created the Neural Variational Document Model (NVDM) in 2016 to model documents [49]. The goal of NVDM is to create a "continuous semantic latent variable for each document." The NVDM is an unsupervised generative model that uses a neural network to perform a multinomial logistic regression on the document set, resulting in what is essentially a word embedding vector for each document. The authors tested their model against LDA and other neural document classifiers on the Twenty Newsgroups data set, and a set of Reuters newspaper articles consisting of 804,414 articles. The authors use perplexity as their only evaluation metric for topic quality and show that the perplexity of their model is lower than that of the baseline models.

5.3.5 lda2vec. In 2016, Moody created a model called lda2vec, with the goal of directly incorporating the word2vec model into the classic LDA model [51]. lda2vec alters the word2vec model to create document vectors as well as word vectors. Document vectors allow the measurement of similarity between documents, and between documents and words or phrases. Each topic is a vector in the same space as the word and document vectors, calculated by summing the probability of each document belonging to that topic. The resulting topic vector can be compared to word vectors to find the most similar words to the topic. The author shows that this approach works well on Hacker News comments, containing 66,000 short texts, as well as the Twenty Newsgroup data set, containing 11,313 documents. Moody does not offer any comparison of the model to other state of the art topic models, instead opting to show example topics discovered by lda2vec on each data set.

5.3.6 Pseudo-document-based Topic Model (PTM). In 2016, Zuo et al. proposed the Pseudo-document-based Topic Model [97] as an improvement on the Self-Aggregating Topic Model [62]. PTM assumes that short texts are generated from longer pseudo-documents. For each short text observed, a latent pseudo-text is drawn from the distribution of pseudo-documents. A topic is then drawn from the topic distribution of the pseudo-text as opposed to the short text, and that topic is used to generate the next word in the short text. The authors' hypothesis is that by condensing many short texts into a single pseudo-document, the word co-occurrence matrix is condensed, leading to a more accurate approximation of topics. The authors claim that their generative process is significantly faster than the two-step process of SATM owing to the fact that their process is a single step in which a single pseudo-document is used to generate a document. The authors also propose a variant of their model, SPTM, to account for sparsity by adding the "Spike and Slab" prior [31] to the topic distribution of pseudo documents.

The authors test their model on four data sets: a news data set consisting of 29,200 articles, a DBLP data set consisting of 55,290 research paper titles from six research areas, a questions data set consisting of 142,690 questions from a Chinese question and answer website, and a Twitter data set consisting of 182,671 tweets labeled with categories. The authors use precision, recall, and f-score to evaluate their models, as well as a PMI-based coherence score. The authors perform a short qualitative analysis of PTM on DBLP, showing how the most probable topics in a pseudo-topic can be explained by the content of the pseudo-topic. The authors show that PTM or SPTM outperform LDA, SATM, and other baseline models on all data sets except for Twitter.

5.3.7 Embedding-based Topic Model. In 2016, Qiang et al. introduced an embedding-based topic model (ETM), for performing topic modeling on short texts with the help of word embedding vectors [60]. Using the Word2Vec framework created by Mikolov et al. [50], the authors create a new distance metric, called the Word Mover's Distance, to measure the difference between documents given the word embedding vectors of their component words. The Word Mover's Distance (WMD), an adaptation of the Earth Mover's Distance, computes the minimum distance between each word in one document to its closest neighbor in the other document. The authors compute the WMD between documents, and then aggregate short documents into longer pseudo-texts using K-means clustering, with the WMD as the distance metric. The authors run LDA on the pseudo-texts to get topic assignments for each pseudo-text and word in the vocabulary. They create a Markov Random Field (manifested as an undirected graph) and for each pair of words with a sufficiently small WMD, they create an edge in the

graph between the two word nodes, representing their shared topic assignment. Then, for each pseudo-text, there is an undirected graph consisting of the nodes referring to the words in the pseudo-text. Words are drawn from a multinomial distribution given their WMD and pseudo-text. As a result, similar words that appear in the same pseudo-text will have a high probability of being placed in the same topic, while similar words that do not appear in the same pseudo-text will not.

The authors tested their model against multiple state of the art models on two data sets, consisting of 16 million tweets and 6,974 news articles, respectively. They find that their approach improves on the coherence of other models. While the example topics that the authors display show less noise than other models, their comparison is on a small set of only four topics.

5.3.8 Gibbs Sampling DMM. In 2014, in an effort to more effectively model topics on short text, Yin and Wang [94] revived the Dirichlet Multinomial Mixture (DMM) created nearly fifteen years earlier by Nigam et al. [57]. Yin and Wang altered the original DMM by proposing a Gibbs sampling algorithm that improved on the scalability of DMM. DMM naturally lends itself to modeling short texts because of its assumption that every document is generated from a single topic. In short texts such as social media, there is often not enough space to reference multiple topics effectively, so a model such as LDA might end up converging to a point where it finds that most documents are generated in large part by a single topic, or erroneously finding that documents are generated by many topics, leading to noisy and inaccurate topics. Yin and Wang perform experiments on DMM using 11,109 news articles from Google News, as well as 2,472 pre-labeled tweets from TREC. The authors compare their model to clustering algorithms instead of topic models. Despite the small change to the original model, this new look at an old model paved the way for other more advanced variants of DMM.

5.3.9 GPUDMM and GPUPDMM. In 2016, Li et al. introduced GPUDMM, another topic model that incorporated word embeddings into a classic model, although in a completely different manner than Moody [42]. GPUDMM, which stands for Generalized Polya Urn (GPU) Dirichlet Multinomial Mixture (DMM), attempts to bring together semantically related words into the same topic using the GPU model. In the Gibbs sampling model, when a word is sampled, it is added back to the chosen topic. In the GPU model, when a word is sampled, a copy of similar words as well as the original word are added back to the chosen topic. This results in groups of similar words all being pushed to the top of a topic together, producing topics that contain more coherent sets of words. The similarity of words in the GPU model is decided by their word embedding distance. So, in comparison to lda2vec, which relies completely on the word2vec embedding model to create topics, GPUDMM augments the traditional DMM model with better words based on the word vector similarities. In terms of the DMM part of the model, the authors borrow directly from Yin et al.'s GSDMM. To incorporate the GPU model into DMM, the authors employ a probabilistic sampling strategy in an attempt to only reinforce those words which are very similar to the sampled topic.

To account for the possibility of documents being generated by more than a single topic, the authors also introduce the GPUPDMM model. This variant of GPUDMM replaces the DMM with a Poisson-based DMM, which allows for one or more topics to generate a document. The difference between PDMM and LDA is that PDMM limits the number of topics that can generate a document based on a Poisson distribution, whereas LDA allows for all topics to contribute to generation with some probability. Aside from the difference in the number of topics allowed to generate a document, GPUDMM and GPUPDMM are identical.

Li and colleagues show results for two different short text data sets, with 12,265 and 179,042 documents respectively. In their experiments, they show that their topics are more coherent than those generated by DMM and other state of the art models that incorporate word embeddings.

5.3.10 Distributed representation-based expansion (DREx). In 2017, Bicalho et al. proposed a framework for expanding short texts using word embeddings [6]. Given a word embedding space and a document, the document

can be expanded by finding the closest ngrams in the embedding space to each ngram contained in the document. For each candidate ngram, if it is sufficiently close to the ngram in the document, the candidate word is added to the document. This process stops when no candidate words remain, or when the document size limit is reached (60 in the paper).

The authors test their framework on seven short text data sets, including four Twitter data sets, two news data sets, and one web search snippet data set, ranging in size from 1,001 documents to 70,707 documents. They show that by expanding using DREx, documents can increase in size from single digits to between twenty and sixty words (because the limit is 60). The authors use NPMI, precision, and recall to compare the performance of models with and without DREx, and find that DREx with GloVe [58] word embeddings performs best throughout all data sets. Testing DREx with LDA [9], LF-LDA [56], and BTM [88], the authors show that DREx improves all results.

5.3.11 Common Semantics Topic Model. In 2018, Li et al. proposed Common Semantics Topic Model (CSTM) [44] to filter noise from topics in social media data sets. Based on DMM [57] (what the authors call mixture of unigrams model), the authors add 'common topics' to the mix. A document can be generated from a single 'function topic' (a traditional topic), and from the set of common topics. Common topics are designed to capture words that appear across all topics. Given k topics, C are defined to be common topics, and K are function topics. For each document, a function topic z_d is sampled from K, and then for each word, a topic is drawn from the mixture of $C \cup z_d$, and the word is drawn from the chosen topic.

The authors test their models on three data sets, one consisting of 12,340 web search snippets, one consisting of 179,022 questions from a Chinese Q&A website, and the final consisting of 2,000,000 tweets. They tested against baseline models including: SATM [62], BTM [88], DMM [57], and DREx+LDA [6]. A qualitative analysis of the tweets data set shows the identified common words grouped into the common topics. CSTM produces more coherent topics than the baseline models (based on NPMI), and performs competitively on classification accuracy and purity. The authors also show the effect of changing the number of common topics on coherence for each data set. The coherence of topics increases up to about five common topics, and decreases as the number of common topics increases further.

5.3.12 Word Embedding LDA. In 2018, Bunk and Krestel proposed a model called WELDA, a combination of word embeddings (WE) and LDA [12]. In their paper, Bunk and Krestel find that word embeddings and topic models do not have a high natural correlation in terms of which words they find to be similar. They find that in terms of judging word similarity, word embeddings are far superior to LDA. With this in mind, they set out to combine word embeddings with LDA in order to create a more coherent topic model.

The authors use a pre-trained embedding model as their embedding space, and perform a slightly altered version of LDA's generative algorithm to find topics. Instead of just being given the topic-word distributions for each topic, an embedded topic distribution is given as well. The embedded topic distribution is a set of words that are closest in the embedding space to the top words of a given topic. Using this embedding space, for each observed word in a document, a coin is flipped with some success probability λ. If this coin flip is successful, then the observed word is replaced in the document with the nearest neighbor to the observed word in the embedded topic distribution. The replaced word is sampled instead of the actual observed word. This results in the top words for each topic moving closer together in the embedding space, helping topics become more coherent.

For their experiments, the authors tested on the Twenty Newsgroups data set and NIPS data set, which contain 11,295 and 1,740 documents, respectively. The authors note the small size of the data sets, but mention that other attempts at word embedding topic models still struggle to process these data sets. The authors use C_V as defined by Röder et al. [64] and word intrusion as defined by Chang et al. [14]. The authors found that the word intrusion scores for WELDA were higher than LDA and slightly higher than or comparable to other baseline models that

used word embeddings. The authors also found that the coherence of WELDA was much better than LDA and better than or comparable to other baseline models.

5.3.13 Laplacian DMM. In 2019, Li et al. adapted DMM to better suit short texts, and proposed Laplacian DMM (LapDMM) [45]. While they note that the assumption of one topic per document already suits DMM to short texts, they incorporate variational manifold regularization in order to preserve the local neighborhood structure of short texts. Manifold regularization in the context of topic modeling adds the constraint that topic representations of document pairs should be similar to each other if they are nearest neighbors in document manifolds. In order to find the nearest neighbors of documents, a graph is constructed prior to training LapDMM that measures document distances. The Laplacian matrix of the graph can then be used as a constraint on the topic assignment of documents, to ensure that documents assigned to the same topic have words in similar neighborhoods in the graph. The authors use word embeddings (specifically Word2Vec [50]) and the Word Mover's Distance [39] to compute the distance between documents. Using word embeddings instead of direct term distances allows one to compare documents with no words in common.

The authors test their model variants against DMM [94], GPU-DMM [42], BTM [88], and an aggregation model similar to SATM [62]. They use three data sets, a TREC question data set consisting of 5952 documents, a web search snippets data set consisting of 12,340 documents, and a Stack Overflow data set consisting of 20,000 question titles. Using NPMI and accuracy (the topic of each document was labeled), the authors showed that their model performed better or competitively on all data sets. They also showed that LapDMM was more accurate in classifying documents on each data set. The authors note that the construction of the document graph can be time-consuming, especially for large data sets.

5.3.14 CluWords and CluHTM. In 2019, Viegas et al. proposed a topic model that leveraged clusters of words and TF-IDF to generate topics [78]. The model revolves around the notion of a CluWord, which is a cluster of words given an embedding space. Words belong to the same CluWord if their cosine similarity in the embedding space is greater than some threshold α. Once CluWords for each word in the vocabulary have been computed, each word in each document is replaced by its CluWord. The TF-IDF of the CluWords are then computed to filter out very common and rare CluWords. The authors use CluWord representations of documents with NMF to produce a topic set. The authors test their model on twelve small data sets (909-22,384 documents each), against a number of baseline models including LDA [9], BTM [88], GPUDMM [42], and ETM [60], and find that their model's coherence outperforms the baselines.

In 2020, Viegas et al. proposed a model based on CluWords and NMF called CluHTM to perform hierarchical topic modeling [79]. The model is initialized in the same manner as CluWords [78], but once the initial set of topics is approximated, instead of finding CluWords and performing modeling on the entire data set, it recursively focuses only on the documents belonging to a single topic at a time. This repeated process results in a hierarchy of topics where each set of topics one rung down corresponds to one of the original topics and so on. The authors test CluHTM on the same data sets as in the CluWords paper, this time testing against other hierarchical models. They find that in most data sets, CluHTM outperforms the other models in terms of coherence.

5.3.15 Embedded Topic Model and Dynamic Embedded Topic Model. In 2019, Dieng, Ruiz, and Blei introduced another version of a topic model assisted by word embeddings [25]. In their model, also named ETM, words and topics are both represented by a vector in an embedding space. Because topics and words are projected onto the same embedding space, words can be placed into topics probabilistically based on how close a word vector is to the topic vector. The generative process described is similar to that of LDA, where for each document, a topic is drawn according to a probability distribution. In the case of LDA, that distribution is the Dirichlet. In the case of ETM, that distribution is the logistic-normal distribution, in order to facilitate easier reparameterization in the inference algorithm. Then, for each word, a topic assignment is drawn. Finally, the observed word is drawn, in

embedding form, from the assigned topic. In this way, the words are drawn from their context (the embeddings), instead of from the words that are close to them in the given document. The ETM model is fit using a neural network with the goal of maximizing the log marginal likelihood of observed documents.

ETM can be used with or without pretrained embeddings, and in the case of pretrained embeddings, will assign words in the embedding space to topics even if they do not occur in the corpus. This could be particularly useful in sparse data sets like domain-specific Twitter data, where there is a limited vocabulary and little word co-occurrence reinforcement.

The authors compare their model to LDA and NVDM. They test their model on the Twenty Newsgroup data set, and a data set of over 1.8 million news articles from the New York Times. They use a hybrid evaluation metric, topic quality, which is the normalized product of the topic coherence (normalized PMI) and topic diversity. The authors find that ETM with pretrained embeddings performed best on Twenty Newsgroups, consistently outperforming other methods. On the New York Times data set, other models catch up to it, with a much closer spread in terms of interpretability and predictive power. However, it still outperforms the other models. The authors also show that ETM is robust to stopwords. If stopwords are left in the documents, it finds a few stopword topics, but stopwords do not infiltrate other topics as is the case with many other topic models.

Also in 2019, Dieng, Ruiz, and Blei designed the Dynamic Embedded Topic Model for temporal topic modeling assisted by word embeddings [26]. A variant of the ETM that they first designed, D-ETM adds a time-varying aspect to the model. The difference in the model's generative process is that the generation is run for each time step, such that there are k topics in each time step, still all projected onto an embedding space. Word embeddings are not time dependent in this model, so it is possible that topics from an earlier time step could pick up words that appear only in documents in later time steps. This helps for continuity of topics through time steps, but may be a problem when judging the accuracy of a topic at a given time.

The authors test their model on three temporal data sets, including ACL abstracts (8,936 documents), articles from *Science Magazine* (13,894 documents), and United Nations (UN) general debates (196,290 documents). They tested against two dynamic versions of LDA (D-LDA and D-LDA-REP). The authors use topic coherence, topic diversity, and topic quality, and find that in the UN and *Science* data sets, D-ETM beats D-LDA on topic diversity and quality, but loses in terms of coherence. In the smallest data set, ACL, D-ETM performs the best across all metrics. The authors also find that D-ETM beats D-LDA on two of three data sets in terms of perplexity.

5.3.16 Topic Modeling with BERT. In 2020, Thompson and Mimno [76] proposed using the language model BERT [24] to produce topics. The authors use k-means to cluster tokens observed in the data set based on their contextual vectors drawn from BERT. BERT is a language model originally proposed in 2018 that is a contextual word embedding space. It is a bidirectional model, meaning that its word embeddings consider both the left and right side context [24]. This differs from previous models such as GloVe and Word2Vec which are context-free embedding spaces with a single embedding representation for each word. It excels across the board on traditional NLP tasks. The authors propose different model variants based on different variants of the BERT model. In their experiments, the authors employ three data sets, including a 1,000 document Wikipedia data set, a 5,300 document data set consisting of U.S. Supreme Court opinions, and a 25,000 document data set consisting of Amazon product reviews. The authors test their model variants against LDA using PMI-based topic coherence, and diversity (which they call exclusivity). The authors show that for any given metric, at least one of their model variants outperforms LDA; however, no model consistently outperforms LDA on every metric. In a qualitative analysis, the authors show how their models are more syntactically-aware in their clusterings than LDA.

5.3.17 Topic-Noise Models. In 2021, Churchill and Singh proposed a new type of topic model, topic-noise models, that jointly approximates topic and noise distributions [20]. Their model, Topic-Noise Discriminator (TND) uses a Beta distribution to decide whether an observed word belongs in a topic or in the noise distribution. If a word is determined to be a noise word, TND samples the nearest words in an embedding space and adds them to the

noise distribution as well. This fortifies the noise distribution, resulting in stronger and more accurate noise filtering. The authors combine TND and LDA to create Noiseless LDA (NLDA), an ensembled topic-noise model. NLDA uses the noise distribution of TND and the topic distribution of LDA to generate topics, again using the Beta distribution to determine whether a word is noise or a topic word.

The authors test TND and NLDA on the Twenty Newsgroups dataset [41], as well as a data set containing one million tweets about the Covid-19 pandemic, and a data set containing 1.4 million tweets about the 2020 United States Elections. Comparing to LDA, DMM, GPUDMM, and CSTM using topic coherence, diversity, and noise penetration metrics, the authors show that NLDA performs better on social media data, especially in terms of coherence. The authors also conduct a qualitative analysis that shows the interpretability of topics generated by topic-noise models using example topics and human judgments.

NLP-aided topic models have evolved in the past eight years, mostly due to the innovation of Word2Vec in 2013. The best of these models incorporate word embeddings in some smart way, not to wholly replace classic topic model frameworks, but to help reduce the sparsity of the word co-occurrence space. Word embeddings and deep-learning language models like BERT are not suitable to topic modeling themselves, due to their difference in objective from topic models. Instead of trying to produce a set of words related to a topic, word embeddings attempt to cluster words based on their contextual similarities. Language models like BERT are trained to produce coherent sentences and syntax, which is not required in topics, and can sometimes lead to less coherent topics. Despite their difference in objective from topic models, these models have been shown to be useful for topic augmentation. As shown by Dieng et al. in 2019 and by Churchill and Singh in 2021, these models continue to get more accurate, and the evaluation metrics continue to evolve, honing in on important aspects of topic models in different scenarios.

5.4 Meta-data Augmented, Supervised, and Reinforcement Learning based Models

While the scope of this survey is focused specifically on unsupervised topic models, we describe a few seminal works for meta-data augmented, supervised, and reinforcement learning based topic models that have begun to appear in recent years. This section is not exhaustive, but rather an introduction into the main approaches in these areas.

In 2011, Zhao et al. proposed Twitter-LDA [95] as a means of producing better topics on Twitter data. Their model worked by aggregating tweets into larger documents by user. These pseudo-documents are then fed into an LDA-based model that assumes a different topic distribution for each user. This approach has shown promising quality on Twitter data, but it requires meta-data (namely, the user who sent each tweet). It also requires the ability to go back and recover the entire tweet set of each user in the data set, which can be impractical for larger data sets. In 2014, Sasaki et al. proposed a hybrid model of Twitter-LDA [95] and Topic Tracking Model (TTM) [32] called Twitter-TTM [67]. It essentially inserted Twitter-LDA's generative model into the TTM temporal framework to produce a temporal model that works well on Twitter data, given the meta-data and extra user tweet sets required by Twitter-LDA. The authors show that Twitter-TTM is much better on their Twitter data set than LDA, Twitter-LDA, and TTM in terms of perplexity.

The main difference between supervised and unsupervised models is that the former relies on documents pre-labeled with topics, or pre-labeled topics, in order to detect topics and label unseen documents. As a result, supervised topic models are less popular because of the reliance on labeled data, which is time-consuming to create and difficult to find.

An early supervised topic model is MDK-LDA [17]. It used user-defined sets of related words drawn from topics in other domains in order to generate topics on a new data set. Each topic is a mixture of these pre-defined sets of words, and documents are drawn from topics, as well as the most related words in the vocabulary to those topics. A supervised Hierarchical Topic Modeling approach uses supervision in the form of a labeled topic

hierarchy to allow for a more accurate hierarchical topic structure [47] (this approach uses a graph structure as opposed to neural networks). In 2019, Adversarial-neural Topic Model (ATM) [85] used a generative adversarial network (GAN) to approximate topics through reinforcement learning. ATM translates each document into TF-IDF vectors and asks the GAN to approximate Dirichlet priors such that it can recreate the document with high accuracy. Generative Adversarial Networks are neural networks that contain a generator neural network and a discriminator neural network. The generator is tasked with creating an acceptable fake version of the document, and presented with both the true and fake document, the discriminator is tasked with guessing which is the fake document. The authors believe that the GAN can learn which are the most important words in the data set and learn patterns in their appearance in similar documents. The authors test ATM on a data set containing just under 100,000 New York Times articles, an encyclopedia data set containing 29,762 documents, and an events data set containing 20,199 news articles from May 2014. The authors use coherence and a qualitative analysis to show that ATM outperforms LDA and other baselines in these data sets.

In 2020, another reinforcement learning algorithm, Bidirectional Adversarial Topic Model (BAT), [84] was proposed to build on ATM. Instead of just the generator network and discriminator network, a third component of the framework was added: the encoder. The encoder takes a V-dimensional document representation (where V is the size of the document), and converts it to a K-dimensional topic distribution (where K is the number of topics). Now, instead of just having to verify that the document is similar to the real document, the discriminator is tasked with evaluating the pair of the generated document *and* its simultaneously generated topic distribution. The data sets that they test on are relatively small and do not contain short texts, but their coherence results on those data sets are strong.

In 2021, Gui et al. introduced another neural topic model with reinforcement learning [28] (VTMRL). This new model leverages topic coherence within the model itself, using the coherence score and what the authors call topic overlap (similar to the inverse of topic diversity) as the reward for reinforcement learning. A topic with high coherence and low topic overlap will be reinforced as a good topic, while the opposite will be punished with a low reward. This in effect allows the model to optimize for topic coherence and diversity. The authors test on the Twenty Newsgroups and a NIPS data set. Also in 2021, Zhao et al. introduced another neural topic model that incorporates word embedding vectors and entity vectors into the model [96] (VAETM). Word vectors have been incorporated in both neural and unsupervised topic models. However, the inclusion of entity vectors (which map a word to a set of known entities that represent topics, concepts, or objects) along with word vectors is another example of the added value of meta-data and outside knowledge bases for generating topic models. The authors test on the Twenty Newsgroups data set, as well as two other data sets (25,000 and 96,000 documents, respectively). While both VTMRL and VAETM test on data sets that are not short texts or social media data, their approaches show a trend toward incorporating outside knowledge like known entity vectors and improving topic generation by employing NLP techniques and models that have been successful for other learning tasks.

5.5 Model-Agnostic Improvements to Topic Modeling

While our focus has been on improvements to topic modeling algorithms in particular, we pause to mention that there are other important developments with regards to topic modeling more generally that can significantly impact the performance of topic models. There has been significant effort in understanding the effects of text preprocessing on topic modeling performance. Schofield et al. analyzed the effects on topics of removing stopwords, finding a signficant improvement in LDA's performance when stopwords were removed [68]. Churchill and Singh created a preprocessing pipeline (textPrep) designed for topic modeling in an effort to standardize preprocessing and provide a single way to access both basic and more sophisticated preprocessing methods [19].

There have been concerted efforts toward making current topic models more efficient. MALLET LDA [48] was originally implemented by McCallum in 2002 but has since been maintained, optimized and parallelized by David

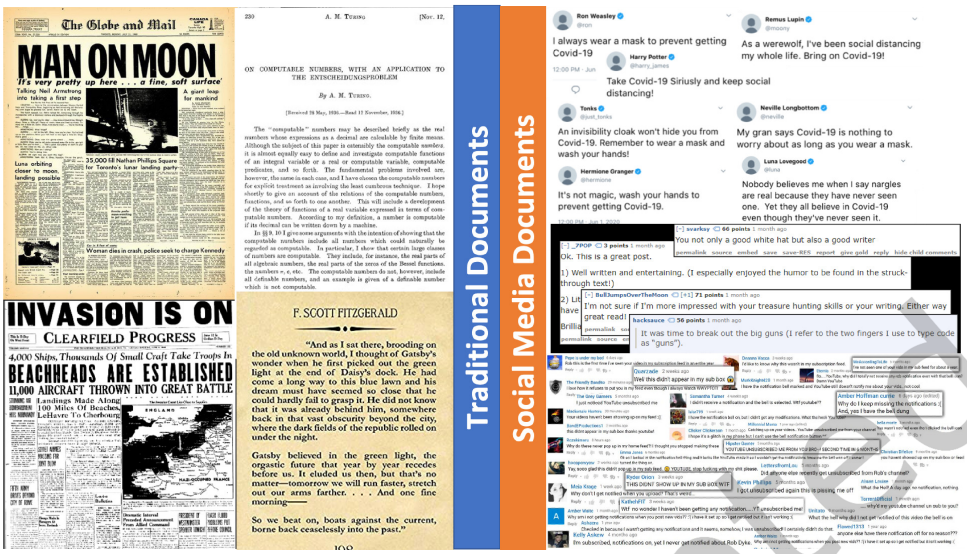


Fig. 5. An example of the variety in documents

Mimno and many others. Smola and Narayanamurthy proposed another parallel architecture for generative topic models in 2010 [72]. Other parallel architectures have been proposed for generative models as well [16, 75]. Researchers have adapted LDA to run on GPUs in order to model larger data sets [43, 87]. Řehůřek and Sojka created Gensim in 2010, a python library containing the most extensive set of topic modeling tools, increasing access to topic modeling for users of all levels of proficiency in coding and topic modeling [63].

6 TOPIC MODELING TODAY AND TOMORROW

In this survey, we have traced the evolution of topic models from their inception up to the present. In this section, we discuss the current state of topic modeling, and our intuition about the criteria researchers should consider when selecting a topic model for different settings.

Thanks to the proliferation of the internet and social media, there are far more types and styles of text data in existence today than when topic models were invented. Social media websites such as Facebook, Twitter, and Reddit produce millions of short text documents every day. Figure 5 shows the variety in modern day documents. Document length and quality, as well as data set volume, vocabulary size, and sparsity are all significant factors that contribute to this variability. It is this variability of factors that has led to the creation of so many different types of topic models.

Given all of the topic models that exist, it is difficult to say that any one model would perform best on a hypothetical data set. Figure 6 provides the main factors to consider when choosing a topic model. The overarching theme of these factors focuses on understanding the properties of the data. The most important factor is the data source. When we want a model that is going to produce good topics out of the box, it is important to compare the data set we are using and the data sets that were used in the original experiments on the specific model. For example, if we are attempting to infer topics on a Twitter data set and we are considering two short text topic models, we would lean toward the one whose authors test on Twitter data sets in their paper. We must also consider the size of our data set. It's not uncommon to see a promising model fail because it cannot scale from the smaller data sets it was tested on to larger data sets, containing millions of documents. An NLP-aided topic model

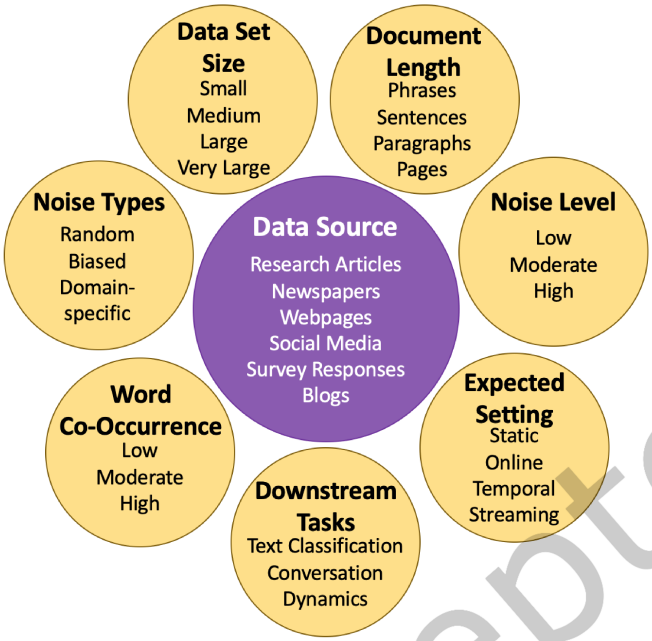


Fig. 6. Factors to consider when choosing a topic model

may perform very well on smaller data sets, but may be too complex to scale to larger data sets. Other unforeseen problems can be avoided if researchers take into account the length, noise levels, and sparsity of documents. The variety of document types has resulted in a field of topic models optimized for different lengths, noise levels, and word co-occurrence densities. Whereas these were not always factors of great concern due to the homogeneity of documents, this new variety has made them far more important. If researchers have a specific setting in mind such as temporal topic modeling or online topic modeling, the number of options is much smaller, and it is likely that an iterative process will be necessary to determine the best model.

Still, given all the empirical investigation, there are a few takeaways with regards to specific methods. LDA [9] and NMF [69] are good general purpose topic models. LDA (specifically the MALLET implementation [48]) can struggle with noise filtration, but can be used in a semi-supervised iterative manner to produce a good ground truth set of topics in those cases [66]. NMF [69] is still a popular choice of topic model as well. It is common to see it used as an alternative to LDA when researchers are uncertain about the characteristics of their data.

When data sets consist of short texts, consider models such as SATM [62], ETM [60], or others that were designed specifically for short texts. When noise is an issue, such as in social media data sets, there are a number of options including TND and NLDA [20], PTM [18], CSTM [44], WELDA [12], and other models designed with noise in mind. Graph-based models such as those proposed by Cataldi et al. [13] and Arruda et al. [22] perform well in their respective niches.

Finally, another overlooked question when choosing a topic model is the task that we wish to use our topic model for. In the past, text classification was by and large the most common downstream task assigned to topic models. Now, we see use of topic models in all sorts of areas, including social science [11, 66] and other fields where humans are directly interpreting topics or using these topics as explanatory variables within more traditional statistical analyzes. Given the new role that topic models are playing across disciplines, as a field,

we need to continue to improve our intuition about the strengths and weaknesses of different topic models in different settings.

A good approach is to run a suite of topic models on a new data set, taking the results of the best model. Qiang et al. [61] published a library of short text topic models that can easily be used in such a way. It includes implementations of ten of the topic models in this survey, many of which we have used in our own research when conducting experiments on our own models. In terms of evaluating topic modeling results, we believe that having at least one method from each of the larger categories of evaluation is essential. Coverage, coherence, and qualitative methods are all important, but no one topic model evaluation is truly complete by itself.

Although topic models have come far and improved greatly over the past three decades, they are still not perfect. Many newer topic models still suffer from noise pollution in topics, and many more are not scalable to the size of modern data sets. Furthermore, there are very few topic models designed to model on a temporal aspect. All of these issues are important facets of a modern topic model, given the noise present in and large scale of social media data sets. The issue of temporal topic modeling is also pertinent given the speed with which topics on a platform can change. Solving these issues, which come down to interpretability and speed, would provide a great benefit to the users of topic models. Online and temporal topic models, and topic models robust to the noise of social media, are what we should strive to produce going forward. Finally, unsupervised topic models have limitations. We touched on meta-data augmented, supervised, and reinforcement learning topic models at the end of Section 5. As machine learning becomes more and more powerful and better understood, it is likely that we will see more and more of these types of topic models. However, due to the growing size of data sets, it may not always be feasible to train computationally expensive models such as supervised and reinforcement learning models. Semi-supervised models may be an important future direction that balances the computational cost and the cost of labeling training data. Topic models are also likely to become useful for generating features for different machine learning and NLP tasks. We are already beginning to see this for learning of complex dynamics like forced migration [70]. Other possible applications include understanding conversation dynamics to detect malicious groups of users in social media networks, identification of different types of misinformation, and comparisons of different types of document collections (newspaper and social media as an example). These are all promising directions that can further increase the impact of topic models.

7 CONCLUSION

This survey describes how topic models have evolved in many directions since their inception. The evolution has been driven in large part by the evolution of the data that researchers want to understand. As the structure and volume of data has changed, the limitations of classic topic models have been exposed. New topic models have been designed to account for the pervasiveness of informal, conversational text that is inherently noisy, short, and unstructured. These models are not always new, as much as they are rediscoveries of older models, such as in the case of DMM. Even today, many of the best models are adaptations of LDA to account for these modern problems. We have seen how graph-based approaches have been used in certain settings in order to reduce noise and find subsets of topics (like emerging topics), but these models do not find the full topic set and cannot perform the same document classification as generative models. We have seen how non-negative matrix factorization can play the part of a topic model, efficiently approximating factor matrices that represent topics. We have seen how relatively simple natural language tools such as word embedding vector spaces can significantly improve the performance of old topic models on different types of text, with little additional computational overhead.

Despite all of these innovations, most of the research in topic models has been in the direction of static models. Even in the case of many temporal models, such as Dynamic Topic Model and Dynamic Embedded Topic Model, there is an assumption that the entire data set is known beforehand. Online learning has been incorporated into some models in order to facilitate faster inference, but the current state of the art is still not designed for building

models using millions of streaming documents on a temporal dimension. Topic models are also not designed for multi-lingual situations. It is not uncommon for social media posts, reviews, emails, and other informal texts to contain multiple languages. Understanding how to identify and incorporate these multi-lingual cues is also an important future direction. In general, as text becomes more informal, multi-lingual, multi-modal, and noisy, future topic models must be optimized for these constraints. Finally, given the new emphasis on algorithmic bias, it is also time to look at which models generate topics that perpetuate bias that exist in the language and which ones compensate for these differences. Toward that end, topic modeling fairness is another area that needs more attention. While the history of topic modeling is rich, a need exists for continued advances in this research area.

Acknowledgements. This work was supported by the Massive Data Institute (MDI) at Georgetown University, and through grants provided by the National Science Foundation (#1934925 and #1934494).

REFERENCES

[1] Deepak Agarwal and Bee-Chung Chen. 2010. fLDA: matrix factorization through latent dirichlet allocation. In *International Conference on Web Search and Data Mining (WSDM)*. 91–100.
[2] Felipe Almeida and Geraldo XexÃľo. 2019. Word Embeddings: A Survey. arXiv:1901.09069 [cs.CL]
[3] Daniel Backenroth, Zihuai He, Krzysztof Kiryluk, Valentina Boeva, Lynn Pethukova, Ekta Khurana, Angela Christiano, Joseph D Buxbaum, and Iuliana Ionita-Laza. 2018. FUN-LDA: a latent dirichlet allocation model for predicting tissue-specific functional effects of noncoding variation: methods and applications. *The American Journal of Human Genetics* 102, 5 (2018), 920–942.
[4] Arindam Banerjee and Sugato Basu. 2007. Topic models over text streams: A study of batch and online unsupervised learning. In *SIAM International Conference on Data Mining (SDM)*. 431–436.
[5] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A neural probabilistic language model. *Journal of Machine Learning Research* 3, Feb (2003), 1137–1155.
[6] Paulo Bicalho, Marcelo Pita, Gabriel Pedrosa, Anisio Lacerda, and Gisele L Pappa. 2017. A general framework to expand short text for topic modeling. *Information Sciences* 393 (2017), 66–81.
[7] David M Blei and John D Lafferty. 2006. Dynamic topic models. In *International Conference on Machine Learning (ICML)*. 113–120.
[8] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2002. Latent Dirichlet Allocation. In *Advances in Neural Information Processing Systems (NIPS)*, T. G. Dietterich, S. Becker, and Z. Ghahramani (Eds.). 601–608.
[9] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet Allocation. *Journal of Machine Learning Research* 3 (3 2003), 993–1022.
[10] Vincent D Blondel, Jean loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. 2008. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment* 10 (2008).
[11] Leticia Bode, Ceren Budak, Jonathan M Ladd, Frank Newport, Josh Pasek, Lisa O Singh, Stuart N Soroka, and Michael W Traugott. 2020. *Words that matter: How the news and social media shaped the 2016 Presidential campaign.* Brookings Institution Press.
[12] Stefan Bunk and Ralf Krestel. 2018. Welda: Enhancing topic models by incorporating local word context. In *ACM/IEEE on Joint Conference on Digital Libraries*. 293–302.
[13] Mario Cataldi, Luigi Di Caro, and Claudio Schifanella. 2010. Emerging Topic Detection on Twitter Based on Temporal and Social Terms Evaluation. In *KDD Workshop on Multimedia Data Mining*.
[14] Jonathan Chang, Jordan Boyd-Graber, Chong Wang, Sean Gerrish, and David M. Blei. 2009. Reading Tea Leaves: How Humans Interpret Topic Models. In *Advances in Neural Information Processing Systems (NIPS)*.
[15] Chaitanya Chemudugunta, Padhraic Smyth, and Mark Steyvers. 2007. Modeling general and specific aspects of documents with a probabilistic topic model. In *Advances in Neural Information Processing Systems (NIPS)*.
[16] Jianfei Chen, Jun Zhu, Jie Lu, and Shixia Liu. 2018. Scalable training of hierarchical topic models. *VLDB Endowment* (2018), 826–839.
[17] Zhiyuan Chen, Arjun Mukherjee, Bing Liu, Meichun Hsu, Malu Castellanos, and Riddhiman Ghosh. 2013. Leveraging multi-domain prior knowledge in topic models. In *International Joint Conference on Artificial Intelligence*.
[18] Rob Churchill and Lisa Singh. 2020. Percolation-based topic modeling for tweets. In *KDD Workshop on Issues of Sentiment Discovery and Opinion Mining (WISDOM)*.
[19] Rob Churchill and Lisa Singh. 2021. textPrep: A Text Preprocessing Toolkit for Topic Modeling on Social Media Data. In *The DATA Conference*.
[20] Rob Churchill and Lisa Singh. 2021. Topic-Noise Models: Modeling Topic and Noise Distributions in Social Media Post Collections. In *International Conference on Data Mining (ICDM)*.
[21] Rob Churchill, Lisa Singh, and Christo Kirov. 2018. A Temporal Topic Model for Noisy Mediums. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD)*.

[22] Henrique Ferraz de Arruda, Luciano da Fontoura Costa, and Diego R. Amancio. 2016. Topic segmentation via community detection in complex networks. *Chaos* 26 (2016).
[23] Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American Society for Information Science* 41, 6 (1990), 391–407.
[24] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).
[25] Adji B Dieng, Francisco JR Ruiz, and David M Blei. 2019. Topic modeling in embedding spaces. *arXiv preprint arXiv:1907.04907* (2019).
[26] Adji B. Dieng, Francisco J. R. Ruiz, and David M. Blei. 2019. The Dynamic Embedded Topic Model. *CoRR* (2019).
[27] Chris Ding, Tao Li, and Wei Peng. 2008. On the equivalence between non-negative matrix factorization and probabilistic latent semantic indexing. *Computational Statistics & Data Analysis* 52, 8 (2008), 3913–3927.
[28] Lin Gui, Jia Leng, Gabriele Pergola, Yu Zhou, Ruifeng Xu, and Yulan He. 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)*. 3478–3483.
[29] Matthew Hoffman, Francis R. Bach, and David M. Blei. 2010. Online Learning for Latent Dirichlet Allocation. In *Advances in Neural Information Processing Systems (NIPS)*. 856–864.
[30] Thomas Hofmann. 1999. Probabilistic latent semantic indexing. In *SIGIR Conference on Research and Development in Information Retrieval*. 50–57.
[31] Hemant Ishwaran, J Sunil Rao, et al. 2005. Spike and slab variable selection: frequentist and Bayesian strategies. *The Annals of Statistics* 33, 2 (2005), 730–773.
[32] Tomoharu Iwata, Shinji Watanabe, Takeshi Yamada, and Naonori Ueda. 2009. Topic tracking model for analyzing consumer purchase behavior. In *International Joint Conference on Artificial Intelligence*. Citeseer.
[33] Hamed Jelodar, Yongli Wang, Chi Yuan, Xia Feng, Xiahui Jiang, Yanchao Li, and Liang Zhao. 2019. Latent Dirichlet Allocation (LDA) and Topic modeling: models, applications, a survey. *Multimedia Tools and Applications* 78, 11 (2019), 15169–15211.
[34] Matthew L Jockers and David Mimno. 2013. Significant themes in 19th-century literature. *Poetics* 41, 6 (2013), 750–769.
[35] G Kaminka et al. 2016. A joint model for sentiment-aware topic detection on social media. In *European Conference on Artificial Intelligence*. 338.
[36] Shiva Prasad Kasiviswanathan, Prem Melville, Arindam Banerjee, and Vikas Sindhwani. 2011. Emerging Topic Detection Using Dictionary Learning. In *International Conference on Information and Knowledge Management (CIKM)*.
[37] Ralf Krestel, Peter Fankhauser, and Wolfgang Nejdl. 2009. Latent dirichlet allocation for tag recommendation. In *ACM Conference on Recommender Systems*. 61–68.
[38] Solomon Kullback and Richard A Leibler. 1951. On information and sufficiency. *The annals of mathematical statistics* 22, 1 (1951), 79–86.
[39] Matt Kusner, Yu Sun, Nicholas Kolkin, and Kilian Weinberger. 2015. From word embeddings to document distances. In *International Conference on Machine Learning (ICML)*. 957–966.
[40] John D Lafferty and David M Blei. 2006. Correlated topic models. In *Advances in Neural Information Processing Systems (NIPS)*. 147–154.
[41] Ken Lang. 1995. 20 Newsgroups Dataset. http://people.csail.mit.edu/jrennie/20Newsgroups/
[42] Chenliang Li, Haoran Wang, Zhiqian Zhang, Aixin Sun, and Zongyang Ma. 2016. Topic Modeling for Short Texts with Auxiliary Word Embeddings. In *SIGIR Conference on Research and Development in Information Retrieval*. 165–174.
[43] Kaiwei Li, Jianfei Chen, Wenguang Chen, and Jun Zhu. 2017. Saberlda: Sparsity-aware learning of topic models on gpus. *ACM SIGPLAN Notices* 52, 4 (2017).
[44] Ximing Li, Yue Wang, Ang Zhang, Changchun Li, Jinjin Chi, and Jihong Ouyang. 2018. Filtering out the noise in short text topic modeling. *Information Sciences* 456 (2018), 83–96.
[45] Ximing Li, Jiaojiao Zhang, and Jihong Ouyang. 2019. Dirichlet Multinomial Mixture with Variational Manifold Regularization: Topic Modeling over Short Texts. In *AAAI Conference on Artificial Intelligence*, Vol. 33. 7884–7891.
[46] Chenghua Lin and Yulan He. 2009. Joint sentiment/topic model for sentiment analysis. In *International Conference on Information and Knowledge Management (CIKM)*. 375–384.
[47] Qingqing Long, Yilun Jin, Guojie Song, Yi Li, and Wei Lin. 2020. Graph Structural-topic Neural Network. In *International Conference on Knowledge Discovery & Data Mining (KDD)*. 1065–1073.
[48] Andrew Kachites McCallum. 2002. MALLET: A Machine Learning for Language Toolkit. (2002).
[49] Yishu Miao, Lei Yu, and Phil Blunsom. 2016. Neural Variational Inference for Text Processing. In *International Conference on Machine Learning (ICML)*, Vol. 48. 1727–1736.
[50] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems (NIPS)*. 3111–3119.
[51] Christopher E. Moody. 2016. Mixing Dirichlet Topic Models and Word Embeddings to Make lda2vec. *CoRR* (2016).
[52] Ramesh M. Nallapati, Susan Ditmore, John D. Lafferty, and Kin Ung. 2007. Multiscale Topic Tomography. In *International Conference on Knowledge Discovery & Data Mining (KDD)*. 520âĂŞ529.

[53] Feng Nan, Ran Ding, Ramesh Nallapati, and Bing Xiang. 2019. Topic modeling with Wasserstein autoencoders. *arXiv preprint arXiv:1907.12374* (2019).
[54] David Newman, Youn Noh, Edmund Talley, Sarvnaz Karimi, and Timothy Baldwin. 2010. Evaluating topic models for digital libraries. In *Joint Conference on Digital Libraries*. 215–224.
[55] David J Newman and Sharon Block. 2006. Probabilistic topic decomposition of an eighteenth-century American newspaper. *Journal of the American Society for Information Science and Technology* 57, 6 (2006), 753–767.
[56] Dat Quoc Nguyen, Richard Billingsley, Lan Du, and Mark Johnson. 2015. Improving topic models with latent feature word representations. *Association for Computational Linguistics (ACL)* 3 (2015), 299–313.
[57] Kamal Nigam, Andrew Kachites McCallum, Sebastian Thrun, and Tom Mitchell. 2000. Text classification from labeled and unlabeled documents using EM. *Machine Learning* 39, 2-3 (2000), 103–134.
[58] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global Vectors for Word Representation. In *Empirical Methods in Natural Language Processing (EMNLP)*. 1532–1543.
[59] Ian Porteous, David Newman, Alexander Ihler, Arthur Asuncion, Padhraic Smyth, and Max Welling. 2008. Fast collapsed gibbs sampling for latent dirichlet allocation. In *International Conference on Knowledge Discovery & Data Mining (KDD)*. 569–577.
[60] Jipeng Qiang, Ping Chen, Tong Wang, and Xindong Wu. 2016. Topic Modeling over Short Texts by Incorporating Word Embeddings. *CoRR* (2016).
[61] Jipeng Qiang, Qian Zhenyu, Yun Li, Yunhao Yuan, and Xindong Wu. 2019. Short Text Topic Modeling Techniques, Applications, and Performance: A Survey. *arXiv preprint arXiv:1904.07695* (2019).
[62] Xiaojun Quan, Chunyu Kit, Yong Ge, and Sinno Jialin Pan. 2015. Short and sparse text topic modeling via self-aggregation. In *International Joint Conference on Artificial Intelligence*.
[63] Radim Řehůřek and Petr Sojka. 2010. Software Framework for Topic Modelling with Large Corpora. In *LREC 2010 Workshop on New Challenges for NLP Frameworks*. 45–50.
[64] Michael Röder, Andreas Both, and Alexander Hinneburg. 2015. Exploring the space of topic coherence measures. In *International Conference on Web Search and Data Mining (WSDM)*. 399–408.
[65] Marco Rossetti, Fabio Stella, and Markus Zanker. 2013. Towards Explaining Latent Factors with Topic Models in Collaborative Recommender Systems. In *International Workshop on Database and Expert Systems Applications*. 162–167.
[66] R Ryan, P. E. Davis-Kean, L. Bode, J Kruger, Z. Mneimneh, and L. Singh. 2020. The new Dr. Spock: Analyzing information provided by parenting-focused Twitter accounts. In *[Unpublished paper under review]*.
[67] Kentaro Sasaki, Tomohiro Yoshikawa, and Takeshi Furuhashi. 2014. Online topic model for twitter considering dynamics of user interests and topic trends. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 1977–1985.
[68] Alexandra Schofield, Måns Magnusson, and David Mimno. 2017. Pulling out the stops: Rethinking stopword removal for topic models. In *Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, Vol. 2. 432–436.
[69] Farial Shahnaz, Michael W. Berry, V.Paul Pauca, and Robert J. Plemmons. 2006. Document Clustering Using Nonnegative Matrix Factorization. *Information Processing Management* (3 2006), 373–386.
[70] Lisa Singh, Laila Wahedi, Yanchen Wang, Yifang Wei, Christo Kirov, Susan Martin, Katharine Donato, Yaguang Liu, and Kornraphop Kawintiranon. 2019. Blending noisy social media signals with traditional movement variables to predict forced migration. In *International Conference on Knowledge Discovery & Data Mining (KDD)*. 1975–1983.
[71] Jennifer Sleeman, Milton Halem, Tim Finin, Mark Cane, et al. 2016. Modeling the Evolution of Climate Change Assessment Research Using Dynamic Topic Models and Cross-Domain Divergence Maps. In *AAAI Spring Symposium on AI for Social Good*.
[72] Alexander Smola and Shravan Narayanamurthy. 2010. An architecture for parallel topic models. *VLDB Endowment* 3, 1-2 (2010), 703–710.
[73] David Sontag and Daniel M Roy. 2009. Complexity of inference in topic models. In *Advances in Neural Information Processing: Workshop on Applications for Topic Models: Text and Beyond*.
[74] Yee Whye Teh, Michael I Jordan, Matthew J Beal, and David M Blei. 2006. Hierarchical Dirichlet Processes. *J. Amer. Statist. Assoc.* 101, 476 (2006), 1566–1581.
[75] Alexander Terenin, Måns Magnusson, Leif Jonsson, and David Draper. 2018. Polya Urn Latent Dirichlet Allocation: a doubly sparse massively parallel sampler. *IEEE transactions on pattern analysis and machine intelligence* 41, 7 (2018).
[76] Laure Thompson and David Mimno. 2020. Topic Modeling with Contextualized Word Representation Clusters. *arXiv preprint arXiv:2010.12626* (2020).
[77] Stephen A Vavasis. 2010. On the complexity of nonnegative matrix factorization. *SIAM Journal on Optimization* 20, 3 (2010), 1364–1377.
[78] Felipe Viegas, Sérgio Canuto, Christian Gomes, Washington Luiz, Thierson Rosa, Sabir Ribas, Leonardo Rocha, and Marcos André Gonçalves. 2019. CluWords: exploiting semantic word clustering representation for enhanced topic modeling. In *International Conference on Web Search and Data Mining (WSDM)*. 753–761.
[79] Felipe Viegas, Washington Cunha, Christian Gomes, Antônio Pereira, Leonardo Rocha, and Marcos Goncalves. 2020. Cluhtm-semantic hierarchical topic modeling based on cluwords. In *Association for Computational Linguistics (ACL)*. 8138–8150.

[80] Daniele Vitale, Paolo Ferragina, and Ugo Scaiella. 2012. Classification of short texts by deploying topical annotations. In *European Conference on Information Retrieval (ECIR)*. 376–387.
[81] Hanna M Wallach, Iain Murray, Ruslan Salakhutdinov, and David Mimno. 2009. Evaluation methods for topic models. In *International Conference on Machine Learning (ICML)*. 1105–1112.
[82] Chong Wang, David M. Blei, and David Heckerman. 2008. Continuous Time Dynamic Topic Models. In *UAI*.
[83] Chong Wang, John Paisley, and David Blei. 2011. Online variational inference for the hierarchical Dirichlet process. In *International Conference on Artificial Intelligence and Statistics*. 752–760.
[84] Rui Wang, Xuemeng Hu, Deyu Zhou, Yulan He, Yuxuan Xiong, Chenchen Ye, and Haiyang Xu. 2020. Neural Topic Modeling with Bidirectional Adversarial Training. *arXiv preprint arXiv:2004.12331* (2020).
[85] Rui Wang, Deyu Zhou, and Yulan He. 2019. Atm: Adversarial-neural topic model. *Information Processing & Management* 56, 6 (2019), 102098.
[86] Xuerui Wang and Andrew McCallum. 2006. Topics over Time: A non-Markov Continuous-time Model of Topical Trends. In *International Conference on Knowledge Discovery & Data Mining (KDD)*.
[87] Xiaolong Xie, Yun Liang, Xiuhong Li, and Wei Tan. 2019. CuLDA_CGS: solving large-scale LDA problems on GPUs. In *Symposium on Principles and Practice of Parallel Programming*. 435–436.
[88] Xiaohui Yan, Jiafeng Guo, Yanyan Lan, and Xueqi Cheng. 2013. A Biterm Topic Model for Short Texts. In *The Web Conference (WWW)*. 1445–1456.
[89] Xiaohui Yan, Jiafeng Guo, Shenghua Liu, Xueqi Cheng, and Yanfeng Wang. 2013. Learning topics in short texts by non-negative matrix factorization on term correlation matrix. In *SIAM International Conference on Data Mining (SDM)*.
[90] Tze-I Yang, Andrew Torget, and Rada Mihalcea. 2011. Topic modeling on historical newspapers. In *ACL-HLT Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities*. 96–104.
[91] Limin Yao, David Mimno, and Andrew McCallum. 2009. Efficient Methods for Topic Model Inference on Streaming Document Collections. In *International Conference on Knowledge Discovery & Data Mining (KDD)*.
[92] Xing Yi and James Allan. 2008. Evaluating topic models for information retrieval. In *International Conference on Information and Knowledge Management (CIKM)*. 1431–1432.
[93] Xing Yi and James Allan. 2009. A comparative study of utilizing topic models for information retrieval. In *European Conference on Information Retrieval (ECIR)*. 29–41.
[94] Jianhua Yin and Jianyong Wang. 2014. A dirichlet multinomial mixture model-based approach for short text clustering. In *International Conference on Knowledge Discovery & Data Mining (KDD)*. ACM, 233–242.
[95] Wayne Xin Zhao, Jing Jiang, Jianshu Weng, Jing He, Ee-Peng Lim, Hongfei Yan, and Xiaoming Li. 2011. Comparing twitter and traditional media using topic models. In *European Conference on Information Retrieval (ECIR)*.
[96] Xiaowei Zhao, Deqing Wang, Zhengyang Zhao, Wei Liu, Chenwei Lu, and Fuzhen Zhuang. 2021. A neural topic model with word vectors and entity vectors for short texts. *Information Processing & Management* 58, 2 (2021), 102455.
[97] Yuan Zuo, Junjie Wu, Hui Zhang, Hao Lin, Fei Wang, Ke Xu, and Hui Xiong. 2016. Topic modeling of short texts: A pseudo-document view. In *International Conference on Knowledge Discovery & Data Mining (KDD)*. 2105–2114.