Efficient and Flexible Topic Modeling Using Pretrained Embeddings and Bag of Sentences

Johannes Schneider

University of Liechtenstein, Vaduz, Liechtenstein

Keywords: Topic Modeling, Sentence Embeddings, Bag of Sentences.

Abstract: Pre-trained language models have led to a new state-of-the-art in many NLP tasks. However, for topic modeling, statistical generative models such as LDA are still prevalent, which do not easily allow incorporating contextual word vectors. They might yield topics that do not align well with human judgment. In this work, we propose a novel topic modeling and inference algorithm. We suggest a bag of sentences (BoS) approach using sentences as the unit of analysis. We leverage pre-trained sentence embeddings by combining generative process models and clustering. We derive a fast inference algorithm based on expectation maximization, hard assignments, and an annealing process. The evaluation shows that our method yields state-of-the art results with relatively little computational demands. Our method is also more flexible compared to prior works leveraging word embeddings, since it provides the possibility to customize topic-document distributions using priors. Code and data is at https://github.com/JohnTailor/BertSenClu.

1 INTRODUCTION

Topic modeling is an actively research text-mining technique (Abdelrazek et al., 2023; Churchill and Singh, 2022) that analyzes a corpus to extract latent topics which describe a semantic concept. It dates back to the last millennium with early techniques such as PLSA (Hofmann, 1999), which compute word-topic and document-topic distributions. Back in 2003 Latent Dirichlet Allocation (LDA) (Blei et al., 2003) marked a milestone in topic modeling. It is still the de-facto standard, i.e., many researchers and practitioners alike still rely on this technique or variants thereof for analyzing diverse text corpora, such as short (Hacker et al., 2020; Schneider et al., 2021), mid-sized (Handali et al., 2020), and long texts (Sbalchiero and Eder, 2020). This is surprising for two reasons: (i) Topic models such as LDA fall short in many ways, and (ii) the field of natural language processing has rapidly advanced since the introduction of LDA in 2003. While LDA can be derived in a mathematically sound and elegant way from a few basic assumptions, many topics identified by LDA are meaningless and removed during manual investigation. Topics might change on a word by word basis resulting in unnatural topic assignments. One cause is the optimization objective of LDA, i.e., perplexity, which tends to assign high probability to frequent words not indicative of any particular topic such as 'a','the','is','are', etc. Perplexity is said to be a measure that does not agree as well with human judgment as other measures like PMI (Newman et al., 2010). Thus, an issue of LDA is that the resulting topic models are not well aligned with human-defined topics. Furthermore, the bag of words(BoW) assumption of LDA combined with the generative modeling approach, leads to words within a single sentence being assigned to multiple different topics, e.g., topics can change after almost every word as also reported in (Gruber et al., 2007; Schneider and Vlachos, 2018). Neglecting word order is convenient from a mathematical point of view since it leads to independence of words given a topic. But it is inaccurate from a semantical perspective, e.g., it makes a big difference, if "You eat a fish" or "A fish eats you". While numerous works have attempted to address issues in LDA (Zhao et al., 2021), the overall success has been limited, especially considering the fact that natural language processing(NLP) has rapidly advanced since 2003. For once, (static) word vectors were popularized in 2013 (Mikolov et al., 2013) in NLP and later further developed into more dynamic, contextual vectors (Devlin et al., 2018). These word vectors led to major improvements in many NLP tasks. They allow computing similarities between individual words or even documents in a fine-grained

407

manner. This is an important step forward in NLP that priorly performed only binary comparisons, i.e., comparing if two words are identical or not. Despite the advantages of word vectors, for topic modeling only relatively few attempts have been undertaken to either include pre-trained static word vectors (Das et al., 2015; Miao et al., 2016; Nguyen et al., 2015), contextualized vectors (Meng et al., 2022; Grootendorst, 2022) or learn them from the corpus (Niu et al., 2015; Dieng et al., 2020). If so, typically a BoW model (Grootendorst, 2022; Miao et al., 2016; Nguyen et al., 2015; Das et al., 2015; Niu et al., 2015) was used often by extending LDA or building on variational inference (Miao et al., 2016). Our approach is to deviate from the classical BoW model, which is too fine-grained and leads to topic changes for potentially any word. We also refrain from an analysis of documents as a whole (Meng et al., 2022), which neglects the possibility that words have multiple topics. We introduce a bag of sentence model (BoS) that splits the document into sequences of a few sentences. That is our elementary unit of analysis is a sequence of typically one to five sentences (rather than a single word), which is sufficient to identify a topic but not too large to contain multiple topics. A sequence is assigned to a single topic, i.e., we perform hard assignments rather than computing soft assignments, which has computational advantages. It is also simpler to understand for the end-user. In contrast to (Meng et al., 2022; Grootendorst, 2022) we also forego any dimensionality reduction and show that this leads to favorable outcomes. Our inference algorithm leverages ideas from expectation maximization to compute topic-document probabilities. It can be understood with the help of the aspect model. To compute assignments of sentences to topics, we rely on extending K-Means from clustering a single set of data points to a set of sets, i.e., we use a cluster prior given by topic-document probabilities. In turn, our inference is orders of magnitude faster than complex deep learning models and more flexible for users than (Meng et al., 2022). However, it is slower than (Grootendorst, 2022) which fails to discover multiple topics within a document and lacks in topic quality. Thus, in summary, our method yields state-of-the-art performance with modest computational costs, while being flexible to adjust to user needs. Our contributions are as follows:

1. We introduce a novel topic model and the bag of sentence model (BoS) that leverages pretrained sentence embeddings. It resembles a compromise between prior approaches relying on words, i.e.,

the bag of words, and documents as a whole, i.e., clustering of documents as a basis for topic modeling.

2. We provide a novel inference mechanism using a form of "annealing" rooted in clustering and the aspect model. To extract topics, i.e., a ranking of words within a topic from sentence to topic assignments, we propose a simple and efficient measure combining frequency of a word within a topic and relative relevance of a word within a topic. This also leads to automatic exclusion of rare words and words not relevant for a topic.

3. Through evaluation on multiple datasets and comparing against multiple methods, we show that our method enjoys modest computational demands while outperforming existing models based on multiple topic measures, i.e., topic coherence and topic coverage as measured using a downstream task. We also perform a holistic evaluation comparing other relevant aspects such as functionality of methods from an end-user perspective.

2 SenClu MODEL

Our model can be viewed from two persepctives. First, it can be seen as performing similarity-based clustering based on sentence embeddings and centroids being the mean of cluster points similar to k-Means++ (Arthur and Vassilvitskii, 2007). However, we also have cluster priors, i.e., topic-document probabilities.

Second, our model can also be viewed from the aspect model (Hofmann, 2001) which defines a joint probability of documents and words $D \times W$ to derive the topic-document probabilities. A key conceptual difference is that we deviate from the word frequency driven generative process (with its flaws) and that we treat (short sequences of) sentences as the smallest unit of analysis rather than single words. That is, a sequence of a few sentences is assigned to a topic. Furthermore, the probability of a sentence given a document is not the typical generative probability (as in LDA). Technically, it is a binary distribution, i.e., 0 or 1, that originates from cluster assignments. Like the standard aspect model we assume conditional independence of a sequence of sentences g and a document d given a topic t:

$$p(g,d) := p(d) \cdot p(g|d) \tag{1}$$
$$p(g|d) := \sum_t p(g|t) \cdot p(t|d) \tag{2}$$

A document d is a sequence of sequences of sentences $d = (g_0, g_1, ..., g_{n-1})$ with each $g_i =$

$(s_j, s_{j+1}, ..., s_{j+n_s})$ being a sequence of n_s consecutive sentences and $j \in [i \cdot n_s, (i+1) \cdot n_s - 1]$. Furthermore, s_i is the i-th sentence in document d. In turn, each sentence $s_i = (w_0, w_1, ...)$ is a sequence of words.[1]. The same word can occur multiple times in a sentence, i.e., it can happen that $w_i = w_j$ for $i \neq j$. Analogously, the same sentence can occur multiple times in a document. The distribution $p(d)$ indicates the likelihood of a document. We do not estimate it, i.e., we simply use a uniform distribution as in other works, e.g.,(Schneider and Vlachos, 2018). This is motivated by the assumption that all documents are considered roughly equally likely. We define the remaining missing terms $p(g|t)$ and $p(t|d)$ of the aspect model (Equation 2) as follows:

$$p(g|d) = \frac{\max_t\{h(g,t) \cdot p(t|d)\}}{\sum_t\{h(g,t) \cdot p(t|d)\}} \propto \max_t\{h(g,t) \cdot p(t|d)\} \quad (3)$$

$$h(g,t) := cos(v_g, v_t) \quad (4)$$

$$p(t|d) := \frac{\alpha + \sum_{i<|d|} 1_{t=\arg\max_{t'}\{h(g_i,t') \cdot p(t'|d)\}}}{|d| \cdot (1+\alpha)} \quad (5)$$

First, observe that both $p(g|d)$ and $p(t|d)$ are probability distributions. But they differ from the classical models (like LDA) that define the probabilities merely as a count based measure (of word occurrences). The term $p(g|d)$ is based on the similarity of the sequence and the topic vector $h(g,t)$ (Eq. 4). To compute the probability of a sentence sequence in a document (Eq. 3), we also use $p(t|d)$ and the concept of latent variables, i.e., topics, as done in the aspect model (Equation 2).

More concretely, we use the idea that a topic t as well as a (consecutive) sentences g can be represented by a vector capturing semantic properties. We use a sentence vector v_g from a pretrained sentence transformer, i.e., we use sentenceBERT (Reimers and Gurevych, 2019) in our evaluation and a topic vector v_t computed based on the assigned sentence vectors v_g to a topic. We average the sentence vectors v_g assigned to topic t to obtain the topic vector v_g. The more similar the two vectors v_g and v_t are, the more related is topic t to sentences in g. As similarity measure we use cosine similarity, which is common for high dimensional vectors, but the dot-product yields almost identical results. Equation 5 states that the probability of a topic t given a document is proportional to the number of sentence sequences g_i assigned to t. We also added a smoothing constant α that gives a user some control over whether to prefer documents being assigned to few or many topics

---

[1]Typically, in topic modeling words also include numbers, but not any punctuation.

(similarly to the topic prior α in LDA). We discuss it in more depth later.

3 COMPUTING WORD-TOPIC SCORES

The established norm for presenting topic model outcomes is listing the most likely words of the word-topic distribution $p(w|t)$. Since our method relies on sentences assignments, i.e., outputs $p(g|t)$, an analogous way would be to report the most likely sentences. However, this bears two disadvantages. First, users are not familiar with such outputs since it differs from all existing topic models. Second, sentences are lengthy, and some words in the sentence might be unrelated to the topic, possibly creating confusion.

Obtaining the empirical word-topic distribution $p(w|t)$ from $p(g|t)$ is straightforward since we assign each sentence to a topic and, in turn, each word to a topic. Thus, we can compute the relative frequency of a word within a topic. A shortcoming of this approach is that common words like "is","are","a","the", etc. obtain high probabilities within all topics since these words can occur in almost any sentence. Words occurring in all topics cannot be used to discriminate among topics. Words are more meaningful to a topic if most occurrences of a word are assigned to that topic only. Thus, we compute a score $score(w|t)$ for a word within a topic using a combination of frequency of a word within a topic and a measure capturing the relevance of the word to topic t relative to other topics. The higher the score, the more relevant the word is for a topic.

The frequency $n(w|t)$ is the number of words assigned to topic t, $n(w|d)$ the occurrences of w in document d and the frequency $n(w)$ of w in the corpus is $n(w) = \sum_t n(w|t)$. The frequency score is computed based on a damped frequency, i.e., $\sqrt{\max(n(w|t) - n_{min}, 0)}$. Without damping, frequency tends to get too much importance.[2] The term n_{min} can be seen as the frequency of words that originates by chance, i.e., from a uniform distribution, or from artificial distributions, e.g., a word occurs mostly just in one document. The term gives a minimum threshold based on the expectation of a word $n(w)/|T|$, if we assigned words uniformly at random to each topic, the standard deviation of $std(n(w|t))$ of w across all topics t and the maximum occurrences of $\max_d n(w,d)$ in a document. It is defined as

---

[2]Damping is common in NLP, e.g., taking the logarithm or square root of frequencies is also done in the computation of tf-idf

$n_{min} := n(w)/|T| + std(n(w|t)) + \max_d n(w,d)$. That is, words should occur more than their expectation (for random assignments) plus the observed empirical standard deviation of assignments. If this does not hold, words are not indicative of a topic. Furthermore, the term $\max_d n(w,d)$ for the maximum occurrence of a word expresses the need that a word must occur in multiple documents. The idea is that topics should be prevalent in many documents rather than originating from one or very few documents. In addition to a frequency measure, we use a measure for "relative" relevance. It is the probability beyond expectation, i.e., $p(t|w) - 1/|T|$. The probability $p(t|w)$ is approximated by the fraction of assignments of the word to the topic relative to its total occurrences. Thus, if a word occurs only in one topic, i.e., we have $p(t|w) = 1$, then the word is highly relevant for the topic, even if it is a very rare word. If the word occurs with the same counts in every topic, we have $p(t|w) = 1/|T|$. Such a word is not indicative of a topic and, therefore, it should have zero relevance. Indeed, the relative relevance is zero, i.e., $p(t|w) - 1/|T| = 0$. Overall, $score(w|t)$ is then the product of the frequency and relevance term, i.e., $\sqrt{n(w|t) - n_{min}} \cdot (p(t|w) - 1/|T|)$. Words that have zero or negative scores are not reported. It can happen that a topic consists only of few words or even no word with positive scores. Commonly, this is the case if a cluster contains a few, very specialized words that might occur only in one or two documents, words that are fairly common across all topics (like "the","a") and a few topical words that occur only infrequently for the topic, i.e., there might be a sports cluster where the word sports occurs 1000 times and a cluster, where it occurs only say 5 times. In this case, both the relevance and frequency score are small. Formulas for all terms are stated also in Algorithm 1.

4 INFERENCE

We maximize the likelihood of the data $\prod_d \prod_{g \in d} p(g,d)$ to estimate model parameters. To this end, we rely on our model definitions (Equations 1, 3, and 5). Generally, inference methods such as Gibbs sampling or variational inference are slow. Speeding up such methods, e.g., by integrating out (collapsing) variables of a Gibbs sampler, are difficult. We utilize expectation-maximization(EM) and clustering ideas to obtain a fast inference mechanism. In a similar fashion as (Schneider and Vlachos, 2018) we derive an EM algorithm based on standard probabilistic reasoning using sentence-topic assignment frequencies. EM performs two steps. The purpose

1:    $p(t|d) := 1/k$; $T := [1,k]$; $epochs := 10$; $c(\alpha) := \max(8,\alpha)$, $n_s := 3$
2:    Tokenize docs $d \in D$ into a sequence of sentences $d := (g_0, g_1, ...)$. Each sentence group $g_i = (s_0, s_1, ..., s_{n_s-1})$ is disjoint and consists of n_s sentences. Each sentence s_i is tokenized into words.
3:    Pretrained sentence embedder model $M : s \to v_s$ with dimension $d_s := |v_s|$ (Reimers and Gurevych, 2019)
4:    Initialize topic vectors $v_t \in \{s | s \in d, d \in D\}$ using k-means++ and cosine similarity
5:    **for** $i \in [1, epochs]$ **do**
6:      $A_{t,d} = \{\}$ $\{\forall t \in T, d \in D\}$ {Begin of E-step}
7:      **for** $d \in D$ **do**
8:        $r :=$ random number in $[0,1]$
9:        $i := 1$ if $r < 0.5 + i/(2 \cdot epochs)$ else 2
10:        **for** $g \in d$ **do**
11:          $t_{g,d} = \arg\max_t^i \{cos(v_g, v_t) \cdot p(t|d)\}$ {$\arg\max^i$ gives the argument that yields the i-th largest value, i.e., 1 gives the largest}
12:          $p(t|g,d) = 1$ if $t_{g,d} = t$ else 0
13:          $A_{t_{g,d},d} := A_{t_{g,d},d} \cup g$
14:        **end for**
15:      **end for**
16:      $v_t := \frac{\sum_{d \in D} \sum_{s \in A_{t,d}} v_s}{\sum_{d \in D} |A_{t,d}|}$ {Begin of M-step}
17:      $p(t|d) := \frac{|A_{t,d}| + c}{|d| + c}$
18:      $c(\alpha) := \max(c(\alpha)/2, \alpha)$
19:    **end for**
20:    *Compute word-topic scores*
21:    $n(w|t) = \sum_{d \in D} \sum_{g \in A_{t,d}} \sum_{s \in g} \sum_{w_i \in s} 1_{w_i = w}$
22:    $p(t|w) := \frac{n(w|t)}{\sum_t n(w|t)}$
23:    $n_{min} := n(w)/|T| + std(n(w|t)) + \max_d n(w,d)$
24:    $score(w|t) := \sqrt{n(w|t) - n_{min}} \cdot (p(t|w) - 1/|T|)$

Algorithm 1: SenClu(docs D, nTopics k, Prior α).

of the E-step is to estimate latent variables, i.e., the probability $p(t|g,d)$ of a topic given sentence(s) g in document d. In the M-step the topic distribution $p(t|g,d)$ is kept fixed and used to maximize the loss function with respect to the parameters. We perform hard assignments. That is, we assume that a sentence in a document has exactly one topic with probability one, and the probability of all other topics is zero. Thus, the topic $t_{g,d}$ of a sentence group g in a document d is the most probable topic:

$$t_{g,d} := \arg\max_t \{h(g,t) \cdot p(t|d)\} \tag{6}$$

$$p(t|g,d) = \begin{cases} 1 & t_{g,d} = t \\ 0 & t_{g,d} \neq t \end{cases} \tag{7}$$

$$A_{t',d} := \{g | t_{g,d} = t', g \in d\} \tag{8}$$

The term $A_{t,d}$ are the sentence groups g assigned to topic t in document d. Our definitions (Equation 6 and following) differ from PLSA and LDA, where each word within a document is assigned a distribution typically with non-zero probabilities for all topics. This has computational advantages. But it also leads to challenges, e.g., getting stuck in local minima that require, for example, an annealing process using randomness to escape such minima that is continuously decreased. In the M-Step, we want to optimize parameters. Analogously to Equations (9.30) and (9.31) in (Bishop, 2006) we define the function

$Q(\Theta, \Theta^{old})$ for the complete data log likelihood depending on parameters Θ:

$$\Theta^{new} = \arg\max_{\Theta} Q(\Theta, \Theta^{old}) \qquad (9)$$

$$\text{with } Q(\Theta, \Theta^{old}) := \sum_{d,t} p(t|D, \Theta^{old}) \log p(D, t|\Theta) \qquad (10)$$

The optimization problem in Equation (10) might be tackled using various methods, e.g., using Lagrange multipliers. Unfortunately, simple analytical solutions based on these approaches are intractable given the complexity of the model Equations (3 and following). However, one might also look at the inference of parameters $p(g|t)$ and $p(t|d)$ differently. Assume that we are given all sentences $A_{t,d}$ assigned to topic t from document d for each document $d \in D$ in a collection of documents D. Then, we define the topic vector v_t to be simply the mean:

$$v_t := \frac{\sum_{d \in D} \sum_{g \in A_{t,d}} v_g}{\sum_{d \in D} |A_{t,d}|} \qquad (11)$$

To obtain the other parameters, our frequentist inference approach uses the empirical distribution: The probability of a topic given a document equals the fraction of sentences of the document assigned to the topic. Under mild assumptions, the maximum likelihood distribution equals the empirical distribution (see, e.g., 9.2.2 in (Barber, 2012)):

$$p(t|d) := \prod_{i < |d|} p(g_i|d) \text{ (Using Equation 5)} \approx \frac{|A_{t,d}|}{|d|} \qquad (12)$$

We also add a value $c(\alpha) := c' \cdot \alpha$ for a user-defined value α and a value c' that is decayed throughout optimization. Thus, ultimately, we use

$$p(t|d) \approx \frac{|A_{t,d}| + c(\alpha)}{|d| + k \cdot c(\alpha)} \qquad (13)$$

The value $c(\alpha)$ serves a dual purpose. The first purpose is to improve the optimization process and avoid local minima and poor initialization. The value c is non-fixed. We start from a large value $c_0 := 8$ and decay it in each epoch i to a new value $c_i := \max(c_{i-1}/2, \alpha)$. Thus, the mininum value is given by a user-defined value α explained later. The decaying process ensures that early non-optimal topic-document distributions do not have too much influence. Technically, it ensures that in the early epochs, all topics remain feasible by ensuring a non-zero probability $p(t|d)$ for each topic. Otherwise, it can happen for small $p(t|d)$ and short documents d that early in the optimization all or most sentences of a document d are assigned to a not yet well-determined

topic t that will significantly evolve. Initially, we use a uniform distribution for $p(t|d)$. But if most or all sentences of a document are assigned to one topic (after the first epoch) then $p(t|d)$ gets close to 1 or even one and, in turn, $p(t'|d)$, of other topics t' gets close to 0 or even 0. Thus, as expressed in Equation 6 the assignment to the not yet well-formed topic t is likely not changed in later epochs, and some topics might not be considered any more since their probability has become zero. Thus, conceptually, early in the optimization, we do not yet fully trust the resulting topic to document assignments since they are subject to change. In turn, we smoothen the resulting probability $p(t|d)$. One might also view the decaying process as a form of (simulated) annealing since it fosters changes of topic sentence assignments initially but slowly makes them less likely by making the distribution $p(t|d)$ more concentrated. The second purpose of $c(\alpha)$, i.e., the user-defined value α, is to control whether in the final assignment of sentences in a document should consist of a few or many topics. A large α leads to a smoother, more uniform topic-document distribution, while a value of α close to 0 leads to a concentrated distribution. From a user's perspective, this is similar to the hyperparameter α in LDA that serves as a prior and impacts the concentration of the topic-document distribution. Generally, for short texts few topics are desirable per document, while longer texts should have more. However, the choice is user and data-dependent. The interpretation of α is that "hypothetically" α sentence groups are assigned to each topic. Thus, α ranges between 0 and ∞. Typical, values are 0 to the average number of sentence groups of a document. We initialize topic vectors v_t with randomly chosen sentence group vectors v_s, very similiar to k-means++[3]. One relevant aspect for our method (as well as for k-means) is that it can get stuck in a local minimum. A topic vector, i.e., a centroid, might be based on an outlier, e.g., if a sentence differs very much from all others and the topic vector is set to that sentence. This is highly undesirable, as it would be better to just assign the outlier to the nearest (large) cluster or remove it. We deal with the problem by randomly assigning sentence groups from one cluster to its nearest neighboring cluster. Of course, this leads to many incorrect assignments. Thus, we also perform annealling, i.e., initally we assign a document to the second most likely cluster with probability about 0.5 and then decrease the probability to 0 within half of the epochs. We must stop the random assignments significantly before the last epoch to ensure that there is also time for convergence. Our method called SenClu is sum-

---

[3]Full details are in code with the link in the abstract

Table 1: Datasets, classes are human pre-defined categories. The Gutenberg dataset contains parts of books of multiple languages.

Dataset	Docs	#Words/doc	Vocabulary	Classes
New York Times (NYT)	31,997	690	25,903	10
20Newsgroups	18,625	122	37,150	20
Gutenberg	9,417	1,768	290,215	50
Yelp	29,820	191	75,791	1

Table 2: The 20 Newsgroups.

comp.graphics, comp.os.ms-windows.misc, comp.sys.ibm.pc.hardware, comp.sys.mac.hardware, comp.windows.x, rec.autos, rec.motorcycles, rec.sport.baseball, rec.sport.hockey, sci.crypt, sci.electronics, sci.med, sci.space, misc.forsale,talk.politics.misc,talk.politics.guns, talk.politics.mideast, talk.religion.misc, alt.atheism, soc.religion.christian

Table 3: Quantitative comparison for number of topics k for SenClu.

Dataset	nTopics k	NMI	PMI
20News	25	$0.46_{\pm 0.01}$	$0.8_{\pm 0.02}$
	50	$0.47_{\pm .003}$	$0.79_{\pm .037}$
	100	$0.47_{\pm 0.0}$	$0.73_{\pm 0.02}$
Guten	25	$0.2_{\pm 0.01}$	$0.83_{\pm 0.07}$
	50	$0.2_{\pm .003}$	$0.67_{\pm .03}$
	100	$0.2_{\pm 0.01}$	$0.75_{\pm 0.03}$
NYT	25	$0.29_{\pm 0.01}$	$0.77_{\pm 0.03}$
	50	$0.28_{\pm .021}$	$0.78_{\pm .025}$
	100	$0.28_{\pm 0.01}$	$0.75_{\pm 0.05}$
Yelp	25	-	$0.65_{\pm 0.03}$
	50	-	$0.62_{\pm .007}$
	100	-	$0.6_{\pm 0.01}$

Table 4: Quantitative comparison for number of sentences n_s for SenClu.

Dataset	#Sen. n_s	NMI	PMI
20News	1	$0.45_{\pm 0.02}$	$0.76_{\pm 0.02}$
	3	$0.47_{\pm .003}$	$0.79_{\pm .037}$
	9	$0.47_{\pm 0.0}$	$0.79_{\pm 0.03}$
Guten	1	$0.19_{\pm 0.01}$	$0.66_{\pm 0.07}$
	3	$0.2_{\pm .003}$	$0.67_{\pm .03}$
	9	$0.21_{\pm 0.02}$	$0.76_{\pm 0.04}$
NYT	1	$0.28_{\pm 0.02}$	$0.75_{\pm 0.03}$
	3	$0.28_{\pm .021}$	$0.78_{\pm .025}$
	9	$0.31_{\pm 0.03}$	$0.79_{\pm 0.03}$
Yelp	1	-	$0.65_{\pm 0.04}$
	3	-	$0.62_{\pm .007}$
	9	-	$0.65_{\pm 0.04}$

marized in Algorithm 1. It takes as input a corpus D, the number of desired topics k, and a prior $\alpha \geq 0$ given the preference for few or many topics per document. The initial value of $c(\alpha)$ has limited impacted as long as it is above 2. The group size for sentences could be altered by the user but we based on the rationale that usually a topic extends a few sentences. That is, any value from 1 to 5 is reasonable and also leads to comparable results. A larger value tends to smoothen the impact of rare words or single sentences that cannot be assigned to any topic without additional context. A very large value, i.e., if the value is about the average number of sentences in a document, leads to the situation that topics are treated as a single unit.

Table 5: Quantitative comparison for prior α for SenClu.

Dataset	α	NMI	PMI
20News	0.25	$0.43_{\pm 0.01}$	$0.69_{\pm 0.03}$
	2	$0.47_{\pm .003}$	$0.79_{\pm .037}$
	8	$0.49_{\pm 0.01}$	$0.81_{\pm 0.02}$
Guten	0.25	$0.16_{\pm 0.01}$	$0.73_{\pm 0.05}$
	2	$0.2_{\pm .003}$	$0.67_{\pm .03}$
	8	$0.24_{\pm 0.01}$	$0.78_{\pm 0.03}$
NYT	0.25	$0.24_{\pm 0.0}$	$0.67_{\pm 0.0}$
	2	$0.28_{\pm .021}$	$0.78_{\pm .025}$
	8	$0.33_{\pm 0.02}$	$0.82_{\pm 0.03}$
Yelp	0.25	-	$0.58_{\pm 0.02}$
	2	-	$0.62_{\pm .007}$
	8	-	$0.66_{\pm 0.02}$

5 EVALUATION

We perform a qualitative and quantitative evaluation on four benchmark datasets, four methods, and three metrics. We also assess the impact of parameters such as the number of topics n_t, the number of sentences n_s (per group), and the prior α.

Settings. We ran our experiments on an Ubuntu 20.4 system running Python 3.9 and Pytorch 1.13 running on a server with 64 GB of RAM, 16 cores of an AMD Threadripper 2950X CPU, and an NVIDIA RTX TI 2080 GPU. If not specified differently, we used $k = 50$ topics, $\alpha = 2$, and the top 10 words of each topic. For each configuration we performed three runs and report average and standard deviation.

Table 6: Quantitative comparison between methods. Times are in minutes.

Dataset	Method	NMI	PMI	Time
20News	BerTopic	0.27 ±.011	0.2 ±.003	0.81 ±.012
	LDA	0.24 ±.007	0.35 ±.002	0.31 ±.003
	TopClus	0.38 ±.012	0.39 ±.016	> 150
	SenClu	0.47±.003	0.79±.037	2.26±.031
Guten	BerTopic	0.09 ±.0	0.44 ±.02	1.6 ±.122
	LDA	0.25 ±.007	0.36 ±.022	0.83 ±.001
	TopClus	0.24 ±.004	0.35 ±.014	> 150
	SenClu	0.2±.003	0.67±.03	5.62±.37
NYT	BerTopic	0.07 ±.009	0.2 ±.002	2.91 ±.05
	LDA	0.21 ±.014	0.36 ±.008	1.24 ±.015
	TopClus	0.25 ±.021	0.42 ±.009	> 150
	SenClu	0.28±.021	0.78±.025	6.47±.158
Yelp	BerTopic	-	0.15 ±.008	0.96 ±.115
	LDA	-	0.32 ±.011	0.37 ±.006
	TopClus	-	0.36 ±.008	> 150
	SenClu	-	0.62±.007	2.91±.045

Methods. LDA (Blei et al., 2003) implemented in Python's Gensim 4.3 (Rehurek and Sojka, 2011) served as a baseline since it is the most well-established topic modeling technique. BERTopic (Grootendorst, 2022) and TopClus (Meng et al., 2022) both use fixed pre-trained contextualized embeddings and some form of clustering, as explained in detail in related work. They represent the state-of-the-art with similar methodology.

Datasets and Pre-Processing. For SenClu we tokenize documents into sentences using a simple rule-based tokenizer (Sadvilkar and Neumann, 2020). The sentences are then transformed to contextual sentences embeddings using sentence transformers (Reimers and Gurevych, 2019). For LDA and to tokenize sentences, we used the gensim default tokenizer, while for BERTopic and TopClus preprocessing is included (as part of the library/repo). As post-processing for all methods we lemmatized topical words and removed duplicates before extracting the top 10 words. The datasets in Table 1 have already been used in the context of topic modeling (Schneider and Vlachos, 2018; Meng et al., 2022) except for the Gutenberg dataset, which consists of books from different categories and languages from the public Gutenberg library (see upload). NYT comes with two human categorizations into 10 classes. We used locations, i.e., 10 pre-defined countries.

5.1 Quantitative Evaluation

We focused topic coherence and topic coverage. Coherent topics are more sensical. For topic coherence, we compute the normalized PMI score (Newman et al., 2010) on a document level as defined in (Schneider and Vlachos, 2018) using the En-

glish Wikipedia dump from 2022/10/01 as an external source. PMI score aligns better with human judgment than measures such as perplexity (Newman et al., 2010). For topic coverage, we utilize a downstream task, i.e., clustering based on topic models compared against predefined human categories as in (Meng et al., 2022) given us the normalized mutual information score(NMI). We also report computation time since energy consumption is important in times of climate crises, and topic models are used by a diverse set of researchers and practitioners, where computation time matters. Computation time includes both training and inference of topics for a corpus. But not computing evaluation metrics for a given set of topics such as PMI, since it is the same for all methods. We include topic specific data pre-processing such as tokenization and computing embeddings, and the running time of the topic modeling, i.e., all steps needed to obtain word-topic scores and topic-document distributions.

Results. Results for the sensitivity analysis of the (hyper)paraemeters are shown in Tables 3, 4, and 5. These parameters have some though limited impact, indicating that at least on a quantitative level, the algorithm behaves fairly insensitive to the parameters. Results for comparing methods are shown in Table 6. The quantitative comparison shows that in terms of topic outcomes, SenClu and TopClus are best. The fact that we outperform on topic coverage on two out of three datasets (Yelp lacks clustering information), i.e. a clustering task, against TopClus is surprising, since the method was explicitly optimized for clustering and evaluated on the same clustering task. We see this is as evidence that optimizing for clustering paired with dimensionality reduction is not needed for high quality topic modeling. On the contrary, it might hemper performance, since fundamental assumptions that would justify dimensionality reduction such as that embeddings are fairly noisy are not satisfied. Our method clearly outperforms on PMI computation, meaning that topic coherence is much larger. This can also be seen when looking at the actual topic words: Other models often have topics with common, non-topical words. A striking disadvantage of TopClus are the long computation times of several hours already for modest sized corpora. Our method SenClu is much faster but still requires considerable time, i.e., a few minutes, compared to LDA and BerTopic that often complete within about a minute. However, quantitative metrics are not sufficient to holistically assess the compared models as shown in the overall evaluation in Section 5.3 and the following qualitative evaluation.

Table 7: Top 7 words of topics by SenClu and TopClus for first 15 of 50 topics.

To.	20Newsgroups Dataset	New York Times Dataset
		Method: SenClu
0	hepatitis, biopsy, cph, chronic, hypoglycemia, pituitary, persistent	banquette, sauce, rum, cucumber, entree, menu, patronize
1	infringe, participle, amendment, verb, indulge, infringing, constitution	tyson, boxing, heavyweight, bout, evander, knockout, holyfield
2	pirating, protection, copy, cracked, pirated, cracker, disassemble	emission, dioxide, carbon, environmentalist, environmental, logging, landfill
3	scsi, ide, drive, controller, bus, modem, mhz	japan, japanese, tokyo, nippon, mitsubishi, nomura, takeshita
4	gld	prosecutor, trial, jury, defendant, judge, lawyer, juror
5	doctor, medication, pain, hernia, diet, migraine, crohn	drug, patient, cancer, doctor, disease, health, dr
6	satan, angel, heaven, enoch, god, eternal, poem	detective, police, arrested, stabbed, murder, arrest, graner
7	wheelie, bike, aerobraking, landing, ride, bdi, riding	mir, astronaut, shuttle, nasa, module, atlantis, spacecraft
8	window, graphic, microsoft, cica, adobe, rendering, shading	germany, german, deutsche, ackermann, frankfurt, dresdner, daimler
9	solvent, bakelite, phenolic, wax, drying, adhesive, soldering	rate, economist, index, nikkei, bond, inflation, economy
10	ei, ax, mq, pl, max, lj, gk	bedroom, apartment, bath, building, square, developer, ft
11	xterm, motif, widget, server, mit, sunos, window	kerry, bush, mccain, clinton, presidential, president, poll
12	israel, israeli, arab, palestinian, lebanese, palestine, gaza	cloning, gene, chromosome, genetic, cloned
13	antenna, frequency, transmitter, radio, receiver, detector, khz	ounce, bullion, dollar, cent, mercantile, settled, crude
14	airmail, mcwilliams, mcelwaine, dublin, expiration, dftsrv, albert	editor, circulation, magazine, reader, tabloid, publishing, journalism
		Method: TopClus
0	please, thanks, thank, appreciate, sorry, appreciated, gladly	student, educator, grader, pupil, teenager, adolescent, school
1	saint, biblical, messiah, missionary, apostle, church, evangelist	surname, mustache, syllable, corps, sob, nickname, forehead
2	iranian, korean, hut, child, algeria, vegetable, lebanese	participation, involvement, effectiveness, supremacy, prowess, responsibility
3	considerable, tremendous, immense, plenty, countless, immensely, various	garage, dwelling, viaduct, hotel, residence, bungalow, building
4	expression, phrase, symbol, terminology, prefix, meaning, coordinate	clit, lough, bros, kunst, mcc, quay, lund
5	memoir, publication, hardcover, encyclopedia, bibliography, paperback	moth, taxa, una, imp, null, def, une
6	anyone, somebody, anybody, someone, anything, everybody, something	many, everybody, anything, everyone, several, much, dozen
7	individual, people, populace, human, being, inhabitant, peer	mister, iraqi, hussein, iraq, iranian, iran, kurdish
8	disturbance, difficulty, complication, danger, annoyance, susceptible, problem	iraqi, iraq, baghdad, saddam, hussein, kuwait, iran
9	beforehand, time, sooner, moment, waist, farther, halfway	dilemma, uncertainty, agitation, reality, dissatisfaction, implication, disagre.
10	upgrade, availability, replacement, sale, modification, repository, compatibility	nominate, terminate, establish, stimulate, locate, replace, protect
11	buy, get, install, spend, sell, keep, build	withstand, hesitate, imagine, explain, apologize, happen, translate
12	appropriated, reverted, wore, abolished, rescued, exercised, poured	forefront, accordance, extent, instance, way, precedence, behalf
13	government, diplomat, fbi, ceo, parliament, officer, parliamentary	privy, continual, outstretched, purposely, systematically, unused, unfinished
14	graduation, university, rural, upstairs, overseas, basement, undergraduate	cautious, goofy, arrogant, painful, cocky, hasty, risky

5.2 Qualitative Evaluation

Table 7 shows top words for the first 15 topics comparing against the method that performed best according to our evaluation and prior work (Meng et al., 2022). We did not attempt to label topics, since during actual topic modeling users face exactly such outputs. However, for the 20Newsgroups dataset we listed the ground truth classes 2 to support understanding of topics and the dataset. Our assignment shows that TopClus somewhat suffers from the same issue as LDA: Common words might form topics that have little meaning and must be eliminated. For example, topic 0, 3, and 6 of the 20Newsgroups dataset consist of frequent non-topical words, while, for instance, topic 4,8 and 9 are not easy to assign to any topic. This is a common issue (as also observed for LDA) nd rooted in the Bag of Words assumption. Other topics are well interpretable, e.g., topic 1 can be easily associated with the ground truth label 'religion' (see Table 2 for ground truth labels) and topic 11 with 'forsale'. For SenClu most topics are easy to interpret, e.g., Topic 1 and 5 discuss a medical topic,

Topic 3 hardware, etc. But it also contains a few topics, which make limited sense. For example, Topic 4 consists of just 1 token and topic 10 can also not well be interpreted.

5.3 Overall Evaluation

Table 8 summarizes the comparison of all methods including quantitative evaluation but also offered functionality by the evaluated methods. Though LDA is very fast and a conceptually elegant approach, it suffers in terms of topic quality, which is the most important aspect of a topic model. Therefore, it is not the method of choice compared to methods relying on pretrained contextual embeddings. This finding is aligned with prior works (Meng et al., 2022; Grootendorst, 2022). BerTopic is very fast but topic quality is often not top notch and it treats documents as just having one topic. This is against the key idea of topic models that documents can have multiple topics. It is extremely problematic for long, diverse texts, where this is almost certainly the case. TopClus yields high quality topics but suffers from chal-

Table 8: Summarized Assessment of Methods.

Method	Multiple topics per doc?	Topics per doc controllable?	Speed	Topic quality	Method complexity
LDA	Y	Y	Fast	Medium	Low
BerTopic	N	N	Fast	Medium-High	Low
TopClus	Y	N	Very Slow	High	Medium
SenClu(ours)	Y	Y	Medium	High	Low

lenges in interpretability, i.e., it uses a neural network with multiple loss functions and thousands of neurons. Neural networks are known to be challenging to interpret (Schneider and Handali, 2019; Meske et al., 2022; Longo et al., 2023). More concerning is that its computational overhead makes TopClus tedious to apply in practice. Furthermore, users cannot express their intentions of how many topics are desirable per document. That is, if a user explicitly wants that documents should have a few topisc (or, on the contrary, many topics), the user cannot specify this. LDA and our method come with a hyperparameter (α) guiding the algorithm to prefer few or many topics per document, which is a clear advantage. Thus, in summary, SenClu achieves state-of-the-art topic quality in reasonable time and it offers all functionality desirable by a user.

6 RELATED WORK

Early, Discrete Topic Models. Already in the last millennium probabilistic latent semantic analysis (PLSA) (Hofmann, 1999) was introduced as an improvement over latent semantic analysis using discrete word representations, i.e., one-hot word encodings. Latent dirichlect allocation (LDA) (Blei et al., 2003) adds priors with hyperparameters to sample from distributions of topics and words as a generalization over LDA. LDA has been extended and varied in many ways. The generative probability of a word within a topic $p(w|t)$ is based on the relative number of occurrences of a word in a topic. In contrast, we use the idea that semantic similarity determines the probability $p(s|d)$ that a sentence, which is our unit of analysis rather than words as in LDA, belongs to a topic. While most works rely on words, especially, a bag of words model, few models also investigated sentence assignments. For example, in (Gruber et al., 2007), each sentence is assigned to a topic using a Markov chain to model topic-to-topic transitions after sentences. Though technically very different, the work also shares the idea that sentences are an important unit of analysis. Such an approach avoids that nearby words being

assigned many different topics. (Balikas et al., 2016) is an extension of LDA that adds the extra "plate" of a sentence. All words of a sentence are assigned to the same topic. However, the model does not resolve the fundamental issue of LDA that frequency determines likelihood, e.g., short sentences of frequent words (and, in turn, frequent words) are most likely for a topic. (Schneider and Vlachos, 2018) used the idea to identify keywords that can influence the topic of nearby words, which effectively also leads to the situation that chunks of texts are assigned the same topic.

Topic Models with Continuous Word Representations. The aforementioned works treat words as one-hot encodings. With the success of (static) word vectors (Mikolov et al., 2013) based on neural networks, the idea to use continuous representations also emerged in topic modeling with early approaches such as (Niu et al., 2015; Das et al., 2015; Miao et al., 2016). Although the idea to use external knowledge beyond the corpus to improve topic models has been known (Newman et al., 2011), most works estimated vectors from the corpus used for topic modeling. Since then neural topic models that leverage deep learning networks for topic modeling have gained in popularity (Zhao et al., 2021; dan Guo et al., 2022). They have been used to address various issues such as accounting for correlated and structured topics (Xun et al., 2017), including meta-data and faster inference (Srivastava and Sutton, 2017). (Bianchi et al., 2020) specifically aimed to combine ProdLDA (Srivastava and Sutton, 2017) variant based on auto-encoders with document embeddings (Reimers and Gurevych, 2019). The implementation is simple, i.e., adding document embeddings as part of the input to the autoencoder. The evaluation showed improvements compared to LDA (Blei et al., 2003) and ProdLDA (Srivastava and Sutton, 2017), but improvements were not consistent for other models. (Hoyle et al., 2020) employed knowledge distillation. They computed two distributions over words: One based on a standard topic model such as a variant of LDA (Srivastava and Sutton, 2017) and one based on a pretrained model, i.e., BERT. These two distributions serve as

ground truth for a student network that is trained to reconstruct documents. Most similar to our work are BERTopic (Grootendorst, 2022) and TopClus (Meng et al., 2022). Both use pre-trained contextualized embeddings without any fine-tuning and some form of dimensionality reduction and clustering. Both works argue that dimensionality reduction is beneficial since distance measures tend to differ little when dimensions approach infinity (Grootendorst, 2022). This is true, but it should be kept in mind that word embeddings (Reimers and Gurevych, 2019) only have about 300-800 dimensions and they were intentionally trained using these number of dimensions on large-scale data. Thus, in contrary, dimensionality reduction could lead to an information loss rather than be beneficial. (Meng et al., 2022) argues that training a dimensionality reduction layer optimized towards clustering leads to better outcomes since it facilitates clustering. Thus, in some sense, the burden of separating clusters is placed at least partially on the dimensionality reduction. Here, also dimensionality reduction might lead to an information loss. Furthermore, the proposed architecture for dimensionality reduction in (Meng et al., 2022) requires many parameters, which could lead to overfitting given that clustering corpora are often relatively small, e.g., existing of less than 100k documents. Thus, while dimensionality reduction is often a valuable technique in machine learning, we do not use dimensionality reduction of pre-trained embeddings due to the risk of information loss and the additional complexity. In (Meng et al., 2022) a word is represented as the product of the pre-trained word embedding and an attention weight. Document embeddings are summed attention-weighted word embeddings. An objective is to optimally reconstruct these document sums by summing the topic embeddings of a document. The probability of a word belonging to a topic is given by a soft assignment using cosine similarity of the word embedding and the topic vector. We differ by performing hard assignments, it enforces that a sentence is only assigned to one topic, which we deem more natural from a human topic modeling perspective. It also lessens the computational burden. In (Meng et al., 2022), it was also realized that soft assignments might not be ideal. This issue dealt with squaring of the topic-word distribution, which amplifies the relative difference between the most likely and the second most likely topic. However, this is ad-hoc, since the power of 2 is arbitrary and constitutes a tuning parameter. The training in (Meng et al., 2022) requires three distinct losses, each loss must be weighed, which is more complex than our method conceptually and computationally more

expensive. BERTopic (Grootendorst, 2022) simply computes embeddings of entire documents based on contextual word vectors (Reimers and Gurevych, 2019) and then clusters them using a density-based clustering technique, i.e., HDBScan. Thus, they lack the idea that a document can have multiple topics. Our and other models account for this option. In turn, our clustering is more similar to K-Means though as we also jointly compute topic-sentence probabilities, our clustering is more intricate.

Pretrained Language Models. First works on learning word vectors, a form of distributed representation, date back to the early 2000s (Bengio et al., 2000). They were popularized about 10 years later using a simple neural network architecture that yielded static word vectors derived from large corpora, which allows doing arithmetics on words (Mikolov et al., 2013). Contextual word embeddings (Devlin et al., 2018) replaced static word vectors using much more intricate transformer architectures. They allowed deriving a vector based on contextual words, e.g. a word and nearby words in a text document. Since the first proposal, many possible improvements have been suggested, such as models tailored towards embedding sentences (Reimers and Gurevych, 2019), robustness (Liu et al., 2019), and performance (Sanh et al., 2019). While using a sentence embedder is the obvious choice for a bag of sentence model, other models could also be used and could yield benefits, e.g., for faster inference words within a sentence might be aggregated using fast BERT version (Sanh et al., 2019).

Topic labeling discusses the problem of finding appropriate descriptions for identified topics or paragraphs. Typically, external knowledge such as Wikipedia is used to perform this task (Lau et al., 2011). Our approach uses only knowledge within the corpus as (Schneider and Vlachos, 2018; Grootendorst, 2022) to identify topic words. On a high level, all follow a similar approach, i.e., weigh words based on a measure accounting for frequency of a term and distribution of a term (across topics) as done in classical measures such as term frequency and inverse document frequency (TF-IDF) (Hasan and Ng, 2014).

7 DISCUSSION AND FUTURE WORK

Contextual word embeddings computed using transformers have pushed the state-of-the-art in NLP. However, their usage in topic modeling is still understudied. Our work introducing a bag of sentence

model might also benefit from priorly derived insights for the bag of words, e.g., typically, nearby sentences are more related than more distance sentences. Leveraging such insights could improve the resulting topic models at the expense of a more complex algorithm. Complex and slow algorithms can be major obstacles for users since users might not have powerful machines but still desire to run the model with multiple hyperparameter settings. While our model is fast, in future work, we might further optimize its computation. For example, rather than using all documents to perform an EM step, we might rely on a subset. This could lead to faster convergence. Furthermore, our model could benefit from better sentence tokenizers and improvements in word embeddings. We presented and derived our work from the perspective of the early but very foundational aspect model. An alternative approach is to view it from a clustering perspective, i.e., from k-Means. K-Means clustering also uses expectation maximization and computes cluster centers as the average of all points assigned to a cluster. However, k-Means considers a set of points, whereas we consider sets of sets of points. This hierarchical representation influences the clustering process and requires extensions such as an "annealing" process to obtain high quality topics.

8 CONCLUSIONS

Topic modeling is challenging. While many other NLP tasks have rapidly advanced in the last years, wildly used topic models still date back decades despite striking weaknesses. Our work has shown how to utilize external knowledge in the form of contextual word embeddings in an efficient manner to improve on prior topic models. It not only overcomes major shortcomings of prior works such as extremely long computation times or the inability to extract multiple topics per document, but it also improves on a variety of other measures such as downstream tasks. To do so, we utilize and introduce a novel topic model including inference mechanisms based on multiple ideas such as Bag-of-Sentences, hard assignments, and simulated annealing. Despite our promising experimental evaluation, we have elaborated in our discussion that further improvements might be possible and encourage fellow researchers to engage in this challenge.

REFERENCES

Abdelrazek, A., Eid, Y., Gawish, E., Medhat, W., and Hassan, A. (2023). Topic modeling algorithms and applications: A survey. *Information Systems*, 112:102131.

Arthur, D. and Vassilvitskii, S. (2007). K-means++ the advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1027–1035.

Balikas, G., Amini, M.-R., and Clausel, M. (2016). On a topic model for sentences. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 921–924.

Barber, D. (2012). *Bayesian reasoning and machine learning*. Cambridge University Press.

Bengio, Y., Ducharme, R., and Vincent, P. (2000). A neural probabilistic language model. *Advances in neural information processing systems*, 13.

Bianchi, F., Terragni, S., and Hovy, D. (2020). Pre-training is a hot topic: Contextualized document embeddings improve topic coherence. *arXiv preprint arXiv:2004.03974*.

Bishop, C. M. (2006). Pattern recognition. *Machine Learning*, 128:1–58.

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022.

Churchill, R. and Singh, L. (2022). The evolution of topic modeling. *ACM Computing Surveys*, 54(10s):1–35.

dan Guo, D., Zhao, H., Zheng, H., Tanwisuth, K., Chen, B., Zhou, M., et al. (2022). Representing Mixtures of Word Embeddings with Mixtures of Topic Embeddings. In *International Conference on Learning Representations*.

Das, R., Zaheer, M., and Dyer, C. (2015). Gaussian lda for topic models with word embeddings. In *Proc. of the Asso. for Computational Linguistics(ACL)*.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Dieng, A. B., Ruiz, F. J., and Blei, D. M. (2020). Topic modeling in embedding spaces. *Transactions of the Association for Computational Linguistics*, 8:439–453.

Grootendorst, M. (2022). Bertopic: Neural topic modeling with a class-based tf-idf procedure. *arXiv preprint arXiv:2203.05794*.

Gruber, A., Weiss, Y., and Rosen-Zvi, M. (2007). Hidden topic markov models. In *AISTATS*, pages 163–170.

Hacker, J., vom Brocke, J., Handali, J., Otto, M., and Schneider, J. (2020). Virtually in this together–how web-conferencing systems enabled a new virtual togetherness during the COVID-19 crisis. *European Journal of Information Systems*, 29(5):563–584.

Handali, J. P., Schneider, J., Dennehy, D., Hoffmeister, B., Conboy, K., and Becker, J. (2020). Industry demand for analytics: A longitudinal study.

Hasan, K. S. and Ng, V. (2014). Automatic keyphrase extraction: A survey of the state of the art. In *Proc. of the Annual Meeting of the Association for Computational Linguistics*, pages 1262–1273.

Hofmann, T. (1999). Probabilistic latent semantic indexing. In *Proceedings of Research and development in information retrieval*, pages 50–57.

Hofmann, T. (2001). Unsupervised learning by probabilistic latent semantic analysis. *Machine learning*, 42(1):177–196.

Hoyle, A., Goel, P., and Resnik, P. (2020). Improving neural topic models using knowledge distillation. *arXiv preprint arXiv:2010.02377*.

Lau, J. H., Grieser, K., Newman, D., and Baldwin, T. (2011). Automatic labelling of topic models. In *Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies*, pages 1536–1545.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Longo, L., Brcic, M., Cabitza, F., Choi, J., Confalonieri, R., Del Ser, J., Guidotti, R., Hayashi, Y., Herrera, F., Holzinger, A., et al. (2023). Explainable Artificial Intelligence (XAI) 2.0: A Manifesto of Open Challenges and Interdisciplinary Research Directions. *arXiv preprint arXiv:2310.19775*.

Meng, Y., Zhang, Y., Huang, J., Zhang, Y., and Han, J. (2022). Topic discovery via latent space clustering of pretrained language model representations. In *Proceedings of the ACM Web Conference 2022*, pages 3143–3152.

Meske, C., Bunde, E., Schneider, J., and Gersch, M. (2022). Explainable artificial intelligence: objectives, stakeholders, and future research opportunities. *Information Systems Management*, 39(1):53–63.

Miao, Y., Yu, L., and Blunsom, P. (2016). Neural variational inference for text processing. In *International conference on machine learning*, pages 1727–1736. PMLR.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Newman, D., Bonilla, E. V., and Buntine, W. (2011). Improving topic coherence with regularized topic models. In *Adv. in neural information processing systems*, pages 496–504.

Newman, D., Lau, J. H., Grieser, K., and Baldwin, T. (2010). Automatic evaluation of topic coherence. In *Human language technologies: The 2010 annual conference of the North American chapter of the association for computational linguistics*, pages 100–108.

Nguyen, D. Q., Billingsley, R., Du, L., and Johnson, M. (2015). Improving topic models with latent feature word representations. *Transactions of the Association for Computational Linguistics*, 3:299–313.

Niu, L., Dai, X., Zhang, J., and Chen, J. (2015). Topic2Vec: learning distributed representations of topics. In *2015 International conference on asian language processing (IALP)*, pages 193–196. IEEE.

Rehurek, R. and Sojka, P. (2011). Gensim–python framework for vector space modelling. *NLP Centre, Faculty of Informatics, Masaryk University, Brno, Czech Republic*, 3(2).

Reimers, N. and Gurevych, I. (2019). Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Sadvilkar, N. and Neumann, M. (2020). PySBD: Pragmatic sentence boundary disambiguation. In *Proceedings of Second Workshop for NLP Open Source Software (NLP-OSS)*, pages 110–114, Online. Association for Computational Linguistics.

Sanh, V., Debut, L., Chaumond, J., and Wolf, T. (2019). DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Sbalchiero, S. and Eder, M. (2020). Topic modeling, long texts and the best number of topics. Some Problems and solutions. *Quality & Quantity*, 54(4):1095–1108.

Schneider, J., Hacker, J., Litvinova, Y., Handali, J., and vom Brocke, J. (2021). Exploring the Use of Backgrounds in Web-conferencing with Image and Text Analysis. In *International Conference of Information Systems (ICIS)*.

Schneider, J. and Handali, J. (2019). Personalized explanation in machine learning: A conceptualization. *arXiv preprint arXiv:1901.00770*.

Schneider, J. and Vlachos, M. (2018). Topic modeling based on keywords and context. In *Proceedings of the 2018 SIAM international conference on data mining*, pages 369–377. SIAM.

Srivastava, A. and Sutton, C. (2017). Autoencoding variational inference for topic models. *arXiv preprint arXiv:1703.01488*.

Xun, G., Li, Y., Zhao, W. X., Gao, J., and Zhang, A. (2017). A correlated topic model using word embeddings. In *IJCAI*, volume 17, pages 4207–4213.

Zhao, H., Phung, D., Huynh, V., Jin, Y., Du, L., and Buntine, W. (2021). Topic modelling meets deep neural networks: A survey. *arXiv preprint arXiv:2103.00498*.