Abstract

Current topic models often suffer from discovering topics not matching human intuition, unnatural switching of topics within documents and high computational demands. We address these shortcomings by proposing a topic model and an inference algorithm based on automatically identifying characteristic keywords for topics. Keywords influence the topic assignments of nearby words. Our algorithm learns (key)word-topic scores and self-regulates the number of topics. The inference is simple and easily parallelizable. A qualitative analysis yields comparable results to those of state-of-the-art models, such as LDA, but with different strengths and weaknesses. Quantitative analysis using nine datasets shows gains regarding classification accuracy, PMI score, computational performance, and consistency of topic assignments within documents, while most often using fewer topics.

1 Introduction

Topic modeling deals with the extraction of latent information, i.e., topics, from a collection of text documents. Classical approaches, such as probabilistic latent semantic indexing (PLSA) and more so its generalization, Latent Dirichlet Allocation (LDA) enjoy widespread popularity despite a few shortcomings. They neglect word order within documents, i.e., documents are not treated as a sequence of words but rather as a set or ‘bag’ of words. This might be one reason for unnatural word-topic assignments within documents, where topics change after almost every word (Figure 1). Many attempts have been made to remedy the ‘bag-of-words’ assumption (eg. [7, 8, 22, 24]), but improvement comes usually at a price, a strong increase in computational demands and often model complexity.

The second shortcoming is an unnatural assignment of word-topic probabilities. Better solutions to the LDA model (in terms of its loss function) might result in worse human interpretability of topics [4]. For PLSA (and LDA) the frequency of a word within a topic heavily influences its probability (within the topic) whereas the frequencies of the word in other topics have a lesser impact. Thus, a very common word that occurs equally frequently across all topics might have a large probability in each topic. This makes sense for models based on document “generation”, because frequent words should be generated more often. But using our rationale that a likely word should also be typical for a topic, high scores (or probabilities) should only be assigned to words that occur in a few topics. Although there are techniques that provide a weighing of words, they either do not fully cover the above rationale and perform static weighing e.g., TF-IDF, or the outcome comes at high computational demands with limited improvements [12].

Third, a topic modeling technique should discover an appropriate number of topics for a corpus and a specific task. Hierarchical models based on LDA limit the number of topics automatically, but increase computational costs. PLSA and LDA do not tend to self-regulate the number of topics. They return as many topics as specified by a parameter. We proclaim that this parameter should be seen as an upper bound. Choosing too large a number of topics leads to over-fitting. Furthermore, it is unclear how to choose the number of topics. Manual investigation of topics also benefits from self-regulation, i.e., a reduction of the number of topics to consider.

Other concerns of topic models are the computational performance and the implementation complexity. We provide a novel, holistic model and inference algorithm that addresses all of the above mentioned shortcomings of existing models – at least partially – by introducing a novel topic model based on the following rationale: i) For each word in each topic, we compute a keyword score stating how likely a word belongs to the topic. Roughly speaking, the score depends on how common the word is within the topic and how common it is within other topics. Classical generative models compute a word-topic probability distribution that states how likely a word is generated given a topic. In those models, the probability of a word
given a topic depends less strongly (only implicitly) dependent on the frequency of a word in other topics. ii) We assume that topics of a word in a sequence of words are heavily influenced by words in the same sequence. Therefore, words near a word with high keyword score might be assigned to the topic of that word, even if they themselves are unlikely for the topic. Thus, the order of words has an impact in contrast to bag-of-words models.

iii) The topic-document distribution depends on a simple and highly efficient summation of the keyword scores of the words within a document. Like LDA does, it has a prior for topic-document distributions.

iv) An intrinsic property of our model is to limit the number of distinct topics. Redundant topics are removed explicitly.

After modeling the above ideas and describing an algorithm for inference, we evaluate it on several datasets, showing improvements compared with two other methods (LDA and BTM [24]) on the PMI score, classification accuracy, performance and a new metric denoted by topic-change likelihood. This metric gives the probability that two consecutive words have different topics. From a human perspective, it seems natural that multiple consecutive words in a document should very often belong to the same topic. Models such as LDA tend to assign consecutive words to different topics (Figure 1).

Our model (Equations 2 – 4) maintains the core ideas of the aspect model, but we account for context by taking the position i of a word into account and we use a keyword score \( f(w, t) \). Taking the context of a word into account implies that if a word occurs multiple times in the same document but with different nearby words, it might have different probabilities. To express context, we include the index \( i \), denoting position of the \( i \)th word in the document (see Equation 2). The distribution \( p(d) \) is proportional to \(|d|\). We simply use a uniform distribution. We also assume a uniform distribution for \( p(i|d) \), as we do not consider any position in the document as more likely (or important) than any other. In principle, one could, for example, give higher priority to initially occurring words that might correspond to a summarizing abstract of a text.

### Model Equations

\[
p(d, w, i) := p(d) \cdot p(s|d) \cdot p(w_i = w|d, i) \quad (2)
\]

\[
p(w|d, i) := \max_{t, i \in R_i} \{ (f(w, t) + f(w_{i+1}, t) ) \cdot p(t|d) \} \quad (3)
\]

with \( R_i := [\max(0, i - L), \min(|d| - 1, i + L)] \)

\[
p(t|d) := \frac{\left( \sum_{i \in [0,|d|-1]} f(w_i, t) \right)^{\alpha}}{\left( \sum_{i \in [0,|d|-1]} f(w_i, t) \right)^{\alpha}} \quad (4)
\]

To compute the probability of a word at a certain position in a document (Equation 3), we use latent variables, i.e., topics, as done in the aspect model (Equations 1). Instead of the generative probability \( p(w|t) \) that word \( w \) occurs in topic \( t \), we use a keyword score \( f(w, t) \). A keyword for a topic should be somewhat frequent and also characteristic for that topic only. The keyword scoring function \( f \) computes a keyword score for a particular word and topic that might depend on multiple parameters such as \( p(w|t) \), \( p(t|w) \) and \( p(t) \), whereas the generative probability \( p(w|t) \) is only based on the relative number of occurrences of a word in a topic. We shall discuss such functions in Section 2.1. We use the idea that a word with a high
keyword score for a topic might “enforce” the topic onto a nearby word, even if that word is just weakly associated with the topic. For a word \(w_i\) at position \(i\) in the document, all words within \(L\) words to the left and right, i.e., words \(w_{i+j}\) with \(j \in [-L, L]\), could contain a word (with high keyword score) that determines the topic assignment of \(w_i\). To account for boundary conditions as the beginning and end of a document \(d\), we use \(j \in \{\max(0, i-L)\), \(\min(|d|−1, i+L)\}\). There are multiple options how a nearby keyword might impact the topic of a word. The addition of the scores \(f(w_i, t)\) and \(f(w_{i+j}, t)\) exhibits a linear behavior that is suitable for modeling the assumption that one word might determine the topic of a nearby word, even if the other word is only weakly associated with the topic. Generative models following the bag-of-words model imply the use of multiplication of probabilities, i.e., keyword scores, which does not capture our modeling assumption: A word that is weakly associated with a topic, i.e., has a score close to zero, would have a low score for the topic even in the presence of strong keywords for the topic. Furthermore, each occurrence of a word in a document is assumed to stem from exactly one topic, which is expressed by taking the maximum in Equation 3.

We compute \(p(t|d)\) dependent on keyword scores \(f(w, t)\) of the words in the document \(d\) (Equation 4). We model the idea of looking for keywords in a document and aggregating their score, i.e., \(f(w, t)\). The parameter \(\alpha\) impacts the number of topics per document. The larger \(\alpha\) the more concentrated the topic-document distribution, i.e., the fewer topics per document.

Essentially, these equations allow us to derive an algorithm for inference that is more efficient than LDA, while avoiding overfitting and allowing to model a prior on topic concentration.

### 2.1 Modeling Keywords

Here we state one way how to compute the keyword score \(f(w, t)\) given a word and a topic. Alternative options are discussed in the Related Work Section. A word obtains a high keyword score for a topic if it is assigned often to the topic and the relative number of assignments to the topic is high compared with other topics. The first aspect relates to the frequency of the word \(n(w, t)\) within the topic. The second captures how characteristic a keyword is for the topic relative to others, i.e., \(p(t|w)\). If the topic-word distribution \(p(t|w)\) is uniform, the word is not characteristic of any topic. If it is highly concentrated then it is. This can be captured using the inverse entropy \(H(w)\):

\[
H(w) := - \sum_t p(t|w) \cdot \log(p(t|w))
\]

The entropy \(H(w)\) is maximized for a uniform distribution, i.e., \(p(w|t) = 1/|T|\) giving \(H(w) = \log |T|\), where \(|T|\) is the number of (current) topics, i.e., initially \(k\). Thus, \(1/H(w)\) is a measure that increases the more concentrated the assignment of words to topics is. However, if all occurrences of a word are assigned to a single topic, the entropy is zero and the inverse infinity. Therefore, we add one in the denominator, i.e., \(1/(1 + H(w))\). Furthermore, if the occurrences of a word \(n(w)\) in the entire corpus are fewer than the number of topics \(|T|\), then the word’s entropy can be at most \(\log n(w) < \log |T|\). Ignoring this results in a high keyword score for a rare word even if each occurrence is assigned to a different topic. Thus, we ensure that rare words are not preferred too much by using the factor \(\log \min(|T|, n(w) + 1)\), where the addition of one is to ensure non-zero weights for words that occur once. An optional weight parameter \(\delta\) allows more or less emphasis to be put on the concentration (relative to the frequency within a topic). Overall, our concentration score is:

\[
\text{con}(w) := \left(\frac{\log \min(|T|, n(w) + 1)}{1 + H(w)}\right)^\delta
\]

The second aspect of keyword scores relates to the frequency of the word within a topic, i.e. \(p(w, t) \cdot \sum_{d \in D} |d|\). Damped frequencies, e.g., \(\log(1 + p(w, t) \cdot \sum_{d \in D} |d|)\), work better than using raw frequencies for inference and classification, because classification relies more on concentration, i.e., being certain that a word belongs to a topic. For humans, the words with the highest keyword score are often too specific – they might only be familiar to experts on the topic. Therefore, we propose a second keyword distribution targeted for human understanding that puts more emphasis on frequency using raw counts. Combining the word frequency and the concentration score we get a score \(f(w, t)\) that prefers rather specific keywords and a second one that emphasizes more widely used (known) words \(f_{hu}(w, t)\). Both can be normalized. We add a prior \(\beta\) for \(f(w, t)\), similar to LDA and other models, stating that a word is assumed to occur for each topic at least \(\beta\) times.

\[
\begin{align*}
\text{log}(1 + p(w, t) \cdot \sum_{d \in D} |d|) \cdot \text{con}(w) \\
\text{f}_{hu}(w, t) &\propto (p(w, t) \cdot \sum_{d \in D} |d|) \cdot \text{con}(w)
\end{align*}
\]

### 3 Inference

We want to find parameters that maximize the likelihood of the data, i.e., \(\prod_{i} \prod_{d \in [0, |d|−1]} p(d, w, i)\). A key challenge for inference is the fairly complex model formalized in Equations (2) and (7). Although methods such as Gibbs sampling might be used, they
would be rather inefficient. In particular, their optimizations for faster inference are harder to apply, such as integrating out (collapsing) variables, for a Gibbs Sampler. To derive an efficient inference mechanism, we follow the expectation-maximization (EM) approach combined with standard probabilistic reasoning based on word-topic assignment frequencies. The general idea of EM is to perform two steps. In the E-step latent variables are estimated, i.e., the probability of a word given a topic equals the forward frequentist inference approach, we get that

\[
p(t|w) = \frac{p(w|t) \cdot p(t)}{p(w)}
\]

and

\[
p(w) = \frac{\sum_{w,t} n(w,t)}{\sum_{w,t} n(w,t)}
\]

Thus:

\[
p(t|w) = \frac{\sum_{w',t} n(w',t)}{\sum_{w',t} n(w',t)}
\]

We also need to compute the keyword score \( f(w, t) \). We use that \( \sum_{d \in D} |d| = \sum_{w,t} n(w,t) \), since each word in each document is assigned to one topic.

\[
f(w, t) \propto \log \left( 1 + p(w, t) \cdot \sum_{d \in D} |d| + \beta \right) \cdot \text{con}(w)
\]

We only keep word-topic distributions that are significantly different from each other. Redundant topics can be removed either during inference (as done in Algorithm 1) or after inference. To measure the difference between two word-topic distributions we use the symmetrized Kullback–Leibler divergence.

\[
KL(t_i, t_j) := \sum_w p(w|t_i) \cdot \log \left( \frac{p(w|t_i)}{p(w|t_j)} \right)
\]

\[
SKL(t_i, t_j) := KL(t_i, t_j) + KL(t_j, t_i)
\]

The set of indexes of (significantly) distinct word-topic distributions \( DT \) is such that for any two topics \( i, j \in DT \), it holds that \( SKL(t_i, t_j) \geq \gamma \). We used \( \gamma := 0.25 \).

### 5 Evaluation

TKM and LDA were compared using quantitative (Experiments 1, 2) and qualitative analysis (Experiment 3). We assessed the tendency to self-regulate the number of topics (Experiment 1). We evaluated several metrics, such as the classification accuracy, PMI score and computation time (Experiment 2), where we also included BTM in the evaluation. Finally, topics were assessed qualitatively for one dataset (Experiment 3).
| Symbol | Meaning |
|--------|---------|
| D     | corpus, all documents |
| d     | document from D |
| | number of words in d |
| W     | set of unique words in D |
| w     | word |
| w_i   | i-th word in a document d |
| k     | (maximal) number of topics |
| T     | set (of indexes) of topics |
| DT    | set (of indexes) of distinct topics in T |
| t     | topic t from T |
| t(w, i, d) | topic of word w assuming i-th word w_i = w in d |
| L     | length of sliding wind. to 1 side |
| α, β | topic, word prior |
| δ     | weight for word concentration |
| n(w, t) | number of assignments of word w to topic t |
| n(w) | number of occurrences of word w in D |

Table 1: Notation

| Dataset          | Docs | Unique Words | Avg. Words | Classes |
|------------------|------|--------------|------------|---------|
| BookReviews      | 179541 | 22211       | 33          | 8       |
| WikiBig          | 52024  | 137155      | 346         | 11      |
| Ohsumed          | 23166  | 23068       | 99          | 23      |
| 20Newsgroups     | 18625  | 37150       | 122         | 20      |
| WikiSmall        | 10390  | 60612       | 388         | 82      |
| Reuters21578     | 9091   | 11098       | 69          | 65      |
| CategoryReviews  | 5809   | 6506        | 60          | 6       |
| WebKB4Uni       | 4022   | 7670        | 136         | 4       |
| BrownCorpus      | 500    | 16514       | 1006        | 15      |

Table 2: Datasets

5.1 Algorithms, Datasets and Setup

We compared an implementation of Algorithm 1 in Python, LDA using a collapsed Gibbs sampler available as Python library and BTM [24] available as C++ library. For all algorithms, we used the same convergence criterion, i.e., computation stopped once word-topic distributions no longer changed significantly. For all algorithms, we ran experiments with different parameters for α and β. We chose the best configuration focusing on classification accuracy, i.e., α = 5/k and β = 0.04 for LDA, α = 50/k and β = 0.02 for BTM. For TKM, α = 2.5 and β = 0.05 were chosen for all experiments.

The datasets in Table 2 are public and most have already been used for text classification. For the distinct review datasets (from Amazon) we predicted either the product, i.e., book, based on a review or the product category. The Wiki benchmarks are based on Wikipedia categories. We performed standard preprocessing, e.g., stemming, stopword removal and removal of words that occurred only once in the entire corpus. All experiments ran on a server with a 64bit Ubuntu system, 100 GB RAM and 24 AMD cores operating at 2.6 GHz.

5.2 Experiments

Experiment 1: We empirically analyzed the convergence of the number of distinct topics |DT| (See Section 4) depending on the upper bound k of the number of topics for LDA and TKM.

Experiment 2: We compared various metrics for LDA, TKM and BTM using k = 100. The classification accuracy was measured using a random forest with 100 trees, with 60% of all data for training and 40% for testing. The time to compute the word-topic and topic-document distributions on the training data, the time to compute the topic-document distribution on the test data, the number of distinct topics |DT|
(Equation 12) and the PMI score as proposed in [18] were also compared. PMI measures the co-occurrence of words within topics relative to their co-occurrence within documents (or sequences of words) of a large external corpus, i.e., we used an English Wikipedia dump with about 4M documents as done in [18]. For each pair of words $w_i, w_j$ we calculated the fraction of documents in which both occurred.

$$p(w_i, w_j) = \frac{|\{d|w_i, w_j \in d, d \in D\}|}{|D|}$$

$$p(w_i) = \frac{|\{d|w_i \in d, d \in D\}|}{|D|}$$

$$PMI(w_i, w_j) = \log \frac{p(w_i, w_j)}{p(w_i)p(w_j)}$$

(13)

The PMI score for a topic is the median of the PMI score for all pairs of the ten highest-ranked words according to the word-topic distribution, i.e., for TKM the distribution is $p_{hu}$ as in Equation 7. The PMI score for TKM (and LDA) is the mean of the PMI score of all individual topics. PMI has been reported to correlate better with human judgment than other measures such as perplexity [18].

We introduce a novel measure that captures the consistency of assignments of words to topics within documents as motivated in the end of Section 1. It is given by the probability that two consecutive words stem from different topics. As LDA computes a distribution across all topics, we choose the most likely topic (see Equation 8). The topic change probability $ToC$ for a corpus $D$ is defined as

$$ToC := \frac{\sum_{d \in D} \sum_{i} I(t(i, d) \neq t(i+1, d))}{\sum_{d \in D} |d|}$$

(14)

The indicator variable $I$ is one if two neighboring words have different topics and zero otherwise.

Experiment 3: The topics using $k = 20$ from LDA and TKM for the 20 Newsgroups dataset were assessed qualitatively.

5.3 Results

Experiment 1: The number of distinct topics $|DT|$ of TKM converges when the maximal number of topics $k$ increases (Figure 3). For all datasets LDA showed a significant weaker tendency to limit the number of topics. The fact that topics are typically investigated manually and humans often assign fewer than 100 categories for each dataset (Table 2), suggests that discovering more than a few hundred topics seems less suitable. Furthermore, TKM achieves higher classification accuracy with fewer topics (see results of Figure 2).

Figure 2: Significantly distinct topics given an upper bound.

Experiment 2). LDA’s convergence seems to depend more on the number of documents; that of TKM’s more on the number of unique words. The number of topics $|DT|$ weakly depends on $\alpha$ — more so for TKM. LDA showed no significant changes, even when lowering $\alpha$ to 1/500 of the recommended value of 50/$k$ [6]. Limiting the number of discovered topics, i.e., returning very similar topics, is an intrinsic property of TKM. A word influences the topic of nearby words directly and those of more distant words indirectly. Words with high keyword scores for a topic tend to pull other words to the same topic.

Experiment 2: On average, TKM outperforms LDA and TKM on all metrics as shown in Table 3. For classification TKM dominates or matches LDA except for the Ohsumed dataset. The accuracy for Ohsumed is low for all techniques. It seems that there are too many terms that are shared across different categories, which seems to impact TKM and BTM more strongly than LDA. TKM does much better on the Brown corpus and the WikiSmall corpus than LDA. Both are characterized by limited redundancy, i.e., a relatively large diversity of categories and words given the number of documents. BTM seems somewhat better than LDA but worse than TKM for classification except for the BookReviews dataset, which is characterized by relatively short documents that seem to be better handled by BTM, which was designed for short texts. LDA, BTM and TKM can be made to do better on specific datasets, but at the price of classifying worse on others.

PMI assesses co-occurrence and TKM reports the best results. TKM’s word-topic distribution $p_{hu}$ (Equation 7) is based on weighing words by frequency and
Table 3: Comparison of TKM, BTM and LDA. ‘Bold’ is better at a 99.5% significance level.

| Dataset     | Classification Accuracy | PMI | Distinct Topics | Topic Change Probability | Training Time[Min] | Inference Time[Min] |
|-------------|-------------------------|-----|-----------------|--------------------------|-------------------|--------------------|
|             | TKM | BTM | LDA | TKM | BTM | LDA | TKM | BTM | LDA | TKM | BTM | LDA | TKM | BTM | LDA |
| 20Newsg.    | 0.79 | 0.78 | 0.76 | 1.58 | 0.85 | 1.33 | 99  | 99  | 100 | 0.14 | 0.65 | 0.62 | 3.7  | 66  | 10  |
|             | 0.79 | 0.78 | 0.76 | 1.58 | 0.85 | 1.33 | 99  | 99  | 100 | 0.14 | 0.65 | 0.62 | 3.7  | 66  | 10  |
| Reuters     | 0.9  | 0.87 | 0.83 | 1.36 | 1.26 | 1.28 | 95  | 100 | 100 | 0.14 | 0.63 | 0.59 | 1.1  | 17  | 2.3 |
|             | 0.9  | 0.87 | 0.83 | 1.36 | 1.26 | 1.28 | 95  | 100 | 100 | 0.14 | 0.63 | 0.59 | 1.1  | 17  | 2.3 |
| WebKB4U     | 0.81 | 0.81 | 0.81 | 1.34 | 1.33 | 1.33 | 93  | 100 | 100 | 0.25 | 0.75 | 0.72 | 0.8  | 16  | 1.9 |
|             | 0.81 | 0.81 | 0.81 | 1.34 | 1.33 | 1.33 | 93  | 100 | 100 | 0.25 | 0.75 | 0.72 | 0.8  | 16  | 1.9 |
| WikiBig     | 0.95 | 0.93 | 0.93 | 2.17 | 1.33 | 1.53 | 100 | 100 | 100 | 0.13 | 0.62 | 0.53 | 20   | 516 | 71  |
|             | 0.95 | 0.93 | 0.93 | 2.17 | 1.33 | 1.53 | 100 | 100 | 100 | 0.13 | 0.62 | 0.53 | 20   | 516 | 71  |
| BrownCo.    | 0.45 | 0.42 | 0.3  | 1.34 | 1.21 | 1.37 | 92  | 100 | 100 | 0.38 | 0.88 | 0.79 | 1.0  | 15  | 2.1 |
|             | 0.45 | 0.42 | 0.3  | 1.34 | 1.21 | 1.37 | 92  | 100 | 100 | 0.38 | 0.88 | 0.79 | 1.0  | 15  | 2.1 |
| Ohsumed.    | 0.24 | 0.23 | 0.36 | 2.2  | 2.07 | 2.43 | 99  | 100 | 100 | 0.12 | 0.66 | 0.66 | 2.9  | 66  | 9.1 |
|             | 0.24 | 0.23 | 0.36 | 2.2  | 2.07 | 2.43 | 99  | 100 | 100 | 0.12 | 0.66 | 0.66 | 2.9  | 66  | 9.1 |
| Categor.    | 0.9  | 0.88 | 0.87 | 1.5  | 1.41 | 1.44 | 90  | 100 | 100 | 0.24 | 0.78 | 0.78 | 0.6  | 9.6 | 1.3 |
|             | 0.9  | 0.88 | 0.87 | 1.5  | 1.41 | 1.44 | 90  | 100 | 100 | 0.24 | 0.78 | 0.78 | 0.6  | 9.6 | 1.3 |
| BookRev.    | 0.77 | 0.79 | 0.75 | 1.42 | 1.3  | 1.37 | 73  | 100 | 100 | 0.24 | 0.7  | 0.67 | 8.1  | 146 | 25  |
|             | 0.77 | 0.79 | 0.75 | 1.42 | 1.3  | 1.37 | 73  | 100 | 100 | 0.24 | 0.7  | 0.67 | 8.1  | 146 | 25  |
| WikiSma.    | 0.85 | 0.81 | 0.75 | 1.81 | 1.31 | 1.5  | 100 | 100 | 100 | 0.14 | 0.62 | 0.62 | 5.7  | 127 | 17  |
|             | 0.85 | 0.81 | 0.75 | 1.81 | 1.31 | 1.5  | 100 | 100 | 100 | 0.14 | 0.62 | 0.62 | 5.7  | 127 | 17  |

6 Related Work

Hofmann [19] introduced probabilistic latent semantic analysis (PLSA) as an improvement over latent semantic analysis. Its generalization LDA [23] adds priors with hyperparameters to sample from distributions of topics and words. LDA has been extended and varied in many ways, e.g., [13, 17, 5, 2, 24, 25, 21]. Whereas our model has little in common with PLSA and LDA except its rooting in the aspect model, extensions and
with the exception of \[ 7, 1, 19, 8, 22, 23, 24 \] rely on the bag-of-words assumption. In contrast, we assume that a word influences topics of nearby words. In [7] each sentence is assigned to one topic using a Markov chain to model topic-to-topic transitions after sentences. Multiple works have used bigrams for latent topic models, e.g., \[ 1, 19, 8, 22 \]. \[ 1, 19 \] both multiply probabilities containing conditionals \[ w_i | w_j \] into \[ p(t | w_i, w_j) \propto p(w_i | t) \cdot p(w_j | t) \]. They use distanced n-grams, i.e., for a fixed distance \( d \) between two words they estimate a probability distribution \( p(w_i | t) \) using all word pairs at distance \( d \) from the corpus. N-gram statistics and latent topic variables have been combined in \[ 22 \] and later work, e.g., \[ 23, 24 \]. A key underlying modeling assumption of \[ 22 \] is inferring the probability of one word given its predecessor using smoothed bigram estimators. The sparsity of short texts was the motivation in \[ 24 \] to use bitersms yielding the BTM model. In the BTM model the probability of a biterm equals \( p(w_i, w_j) = \sum_t p(t) \cdot p(w_i | t) \cdot p(w_j | t) \). In \[ 24 \] an increases of the time complexity by about a factor of 3 is reported together with improvements otherwise. We also consider all bitersms within a window and therefore also compare against \[ 24 \]. Aside from that, there are few similarities. Relatively little work has been conducted on limiting the number of topics. The hierarchical topic models \[ 21, 16 \] do not require the specification of the number of topics but come with increased complexity. Keyword extraction often relies on using co-occurrence data, POS tagging \[ 20 \] and external sources or TF-IDF \[ 9 \]. Typically, these methods first extract key-words and then rank them. For example, \[ 20 \] first splits words into phrases using sentence delimiters and stop words. They compute keywords using the ratio of the frequency of a word as well as its degree, i.e., all words that are within a specific distance for any occurrence within a phrase. We do not use any of the typical preprocessing, e.g., POS tagging or phrase extraction, though this might be beneficial. We also tested the metric of \[ 20 \] and found that it gave overall slightly worse result. Topic modeling and keyword extraction are related. For example \[ 14 \] extracts keywords using topic-word distributions obtained from LDA. Key word extraction and clustering are also related. \[ 15 \] uses clustering as a preprocessing step to obtain keyword candidates stemming from

| TKM Topics | LDA Topics | Sim |
|------------|------------|-----|
| ax giz tax myer presid think chz go pl ms | ax giz chz gk pl fij uy fyn ah ei | 1.0 |
| armenian turkish armenia turk turkei azerbaijan azeri | armenian turkish muslim turkei turk armenia peopl war | 0.78 |
| com bike motorcyl dod bmw ride ca write biker articl | car com bike write articl edu engin ride drive new | 0.65 |
| chip clipper encryption wire privaci nsa kei escrow | kei us com chip clipper secur encryption would system | 0.64 |
| game stephanopoulo playoff pitch espn score pitcher pt | game team plai player year edu ca win hockey season | 0.64 |
| monitor simm mh2 mac card centri duo us edu modem | card drive us edu mac driver system work disk problem | 0.64 |
| gun firearm atf weapon stratu fire handgun bd amend edu | gun com edu would fire write articl peopl koresh fbi | 0.64 |
| church christ scriptur bibl kores faith sin god cathol | god christian jesu bibl church as christ sai sin love | 0.64 |
| medic cancer drug patient doctor hiv health newslett | us medic studi disease patient effect drug doctor food | 0.63 |
| edu israel isra arab palestinian articl write adl uci | right israel state isra peopl arab edu war write jew | 0.59 |
| space drive sci orbit shuttl disk nasa id mission hist | space nasa gov launch orbit earth would mission moon edu | 0.55 |
| peopl us would bacteri like jew right know make year | presid would state us mr think go year monei tax | 0.49 |
| imag jpeg gif format file graphic xv color pixel viewer | imag us window graphic avail system server softwar data | 0.47 |
| window mous font server microsoft client xterm us | file us window program imag entri format jpeg need | 0.44 |
| oil kuwait ac write rushdi edu islam uk entri contest | edu write articl com know uic cc would anyone cs | 0.33 |
| homosexu rutger cramer christian sexual gai msg food edu | edu peopl sai write would think com articl moral us | 0.33 |
| insur kei detector radar duke de phone system edu ripem | book post list mail new edu inform send address email | 0.24 |
| team player jesu plai water roger edu laurentian hockei | us would like write edu articl good com look get | 0.23 |
| moral atheist god widget atheism openwindow belief exist | go like get would time know us sai peopl think | 0.14 |
| car brake candida yeast engin militia vitamin steer | us com edu need power work help ca mous anyone | 0.12 |

Table 5: Topics by TKM and LDA for 20Newsgroups dataset for \( k = 20 \)
medoids of these clusters. [15][14] stick to the concept that keywords are extracted based on a preprocessing phase. In contrast, we perform a dynamic iterative approach, in which words are assigned to topics and the distribution of word assignments across topics determines the keyword score. [12] uses term-weighing to enhance topic modeling, i.e., LDA. Their keyword score corresponds to the variance of the word-topic distribution. They do not state a thorough derivation of their Gibbs sampler, i.e., an explicit integration of Equation (4) in [12] to get (6) and (7). Their classification performance reported on the 20Newsgroup dataset for their best algorithm is significantly lower than the performance we observed for LDA (as well as that of our algorithms).

7 Conclusions

We presented a novel topic model based on keywords. Both the model and its inference are simple and performant. Our experiments report improvements to existing techniques across a number of metrics, thus suggesting that the proposed technique aligns closer with human intuition about topics and keywords.

References

[1] M. Bahrani and H. Sameti. A new bigram-plsa language model for speech recognition. EURASIP Journal on Applied Signal Processing, 2010, 2010.
[2] D. M. Blei, T. L. Griffiths, and M. I. Jordan. The nested chinese restaurant process and bayesian nonparametric inference of topic hierarchies. Journal of the ACM (JACM), 57(2):7, 2010.
[3] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. Journal of machine Learning research, 3(Jan):993–1022, 2003.
[4] J. Chang, J. L. Boyd-Graber, S. Gerrish, C. Wang, and D. M. Blei. Reading tea leaves: How humans interpret topic models. In Conf. on Neural Information Processing Systems(NIPS), pages 1–9, 2009.
[5] R. Das, M. Zaheer, and C. Dyer. Gaussian lda for topic models with word embeddings. In Proc. of the Annual Meeting of the Asso. for Computational Linguistics, 2015.
[6] T. L. Griffiths and M. Steyvers. Finding scientific topics. Proceedings of the National academy of sciences, 101(suppl 1):5226–5235, 2004.
[7] A. Gruber, Y. Weiss, and M. Rosen-Zvi. Hidden topic markov models. In AISTATS, pages 163–170, 2007.
[8] M. A. Haidar and D. O’Shaughnessy. PLSA enhanced with a long-distance bigram language model for speech recognition. In European Signal Processing Conference (EUSIPCO), pages 1–5, 2013.
[9] K. S. Hasan and V. Ng. 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, 2014.
[10] T. Hofmann. Probabilistic latent semantic indexing. In Proceedings of Research and development in information retrieval, pages 50–57, 1999.
[11] T. Hofmann. Unsupervised learning by probabilistic latent semantic analysis. Machine learning, 42(1):177–196, 2001.
[12] S. Lee, J. Kim, and S.-H. Myaeng. An extension of topic models for text classification: A term weighting approach. In Conf. on Big Data and Smart Computing (BigComp), pages 217–224, 2015.
[13] W. Li and A. McCallum. Pachinko allocation: Dag-structured mixture models of topic correlations. In Proc. of the int. conference on Machine learning, pages 577–584, 2006.
[14] Z. Liu, W. Huang, Y. Zheng, and M. Sun. Automatic keyphrase extraction via topic decomposition. In Conference on empirical methods in natural language processing, pages 366–376, 2010.
[15] Z. Liu, P. Li, Y. Zheng, and M. Sun. Clustering to find exemplar terms for keyphrase extraction. In Conference on Empirical Methods in Natural Language Processing, pages 257–266, 2009.
[16] D. Mimno, W. Li, and A. McCallum. Mixtures of hierarchical topics with pachinko allocation. In Proceedings of the international conference on Machine learning, pages 633–640, 2007.
[17] D. Newman, E. V. Bonilla, and W. Buntine. Improving topic coherence with regularized topic models. In Advances in neural information processing systems, pages 496–504, 2011.
[18] D. Newman, J. H. Lau, K. Grieser, and T. Baldwin. Automatic evaluation of topic coherence. In Conf. of the North American Chapter of the Asso. for Computational Linguistics, pages 100–108, 2010.
[19] J. Nie, R. Li, D. Luo, and X. Wu. Refine bigram plsa model by assigning latent topics unevenly. In Workshop on Automatic Speech Recognition & Understanding, pages 141–146. IEEE, 2007.
[20] S. Rose, D. Engel, N. Cramer, and W. Cowley. Automatic keyword extraction from individual documents. Text Mining, pages 1–20, 2010.
[21] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei. Sharing clusters among related groups: Hierarchical dirichlet processes. In Advances in neural information processing systems, pages 1385–1392, 2005.
[22] H. M. Wallach. Topic modeling: beyond bag-of-words. In Proc. of the int. conf. on Machine learning, pages 977–984, 2006.
[23] X. Wang, A. McCallum, and X. Wei. Topical n-grams: Phrase and topic discovery, with an application to information retrieval. In International Conference on Data Mining (ICDM), pages 697–702, 2007.
[24] X. Yan, J. Guo, Y. Lan, and X. Cheng. A biterm topic model for short texts. In Proc. of the international conf. on World Wide Web, pages 1445–1456, 2013.
[25] G. Yang, D. Wen, N.-S. Chen, E. Sutinen, et al. A novel contextual topic model for multi-document summarization. Expert Systems with Applications, 42(3):1340–1352, 2015.