Effective Document Labeling with Very Few Seed Words: A Topic Modeling Approach

Chenliang Li[1], Jian Xing[1], Aixin Sun[2], Zongyang Ma[2]
[1]State Key Lab of Software Engineering, Computer School, Wuhan University, China
{cllee,xingj}@whu.edu.cn
[2]School of Computer Science and Engineering, Nanyang Technological University, Singapore
{axsun,zyma}@ntu.edu.sg

ABSTRACT

Developing text classifiers often requires a large number of labeled documents as training examples. However, manually labeling documents is costly and time-consuming. Recently, a few methods have been proposed to label documents by using a small set of relevant keywords for each category, known as *dataless text classification*. In this paper, we propose a **S**eed-Guided **T**opic **M**odel (named STM) for the dataless text classification task. Given a collection of unlabeled documents, and for each category a small set of seed words that are relevant to the semantic meaning of the category, the STM predicts the category labels of the documents through topic influence. STM models two kinds of topics: *category-topics* and *general-topics*. Each category-topic is associated with one specific category, representing its semantic meaning. The general-topics capture the global semantic information underlying the whole document collection. STM assumes that each document is associated with a single category-topic and a mixture of general-topics. A novelty of the model is that STM learns the topics by exploiting the explicit word co-occurrence patterns between the seed words and regular words (*i.e.,* non-seed words) in the document collection. A document is then labeled, or classified, based on its posterior category-topic assignment. Experiments on two widely used datasets show that STM consistently outperforms the state-of-the-art dataless text classifiers. In some tasks, STM can also achieve comparable or even better classification accuracy than the state-of-the-art supervised learning solutions. Our experimental results further show that STM is insensitive to the tuning parameters. Stable performance with little variation can be achieved in a broad range of parameter settings, making it a desired choice for real applications.

Keywords

Topic Modeling, Dataless Text Classification, Text Analysis

1. INTRODUCTION

Text classification refers to the task of assigning category labels to documents based on their semantics. Given the explosive growth of documents, text classifiers have become important tools in managing and analyzing large document collections. Due to its wide usage, text classification has been studied intensively for many years. Existing solutions are mainly based on supervised learning techniques which require tremendous human effort in annotating documents as labeled examples, as shown in the upper part of Figure 1. To reduce the labeling effort, many semi-supervised algorithms have been proposed for text classification [5, 27]. Considering the diversity of the documents in many applications, constructing relatively small training set required by the semi-supervised algorithms remains very expensive.

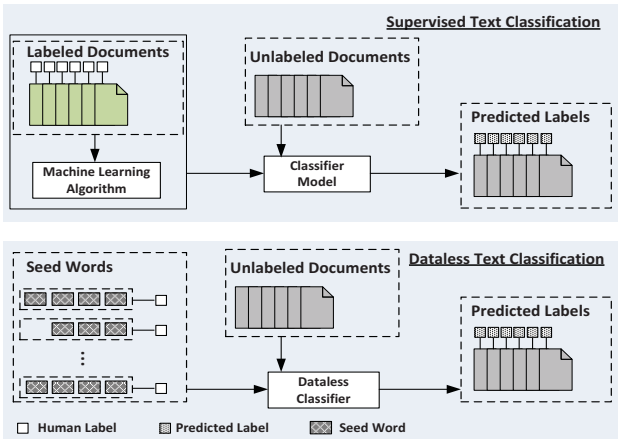


Figure 1: Supervised vs dataless text classification.

Recently, a number of *dataless text classification* methods have been proposed [5, 7, 11, 12, 14, 17, 18, 22, 29]. Instead of using labeled documents as training examples, dataless methods only require a small set of relevant words for each category or labeling the topics learned from a standard LDA model [3], to build text classifiers. As illustrated in Figure 1, dataless classifiers do not require labeled documents, which saves a lot of human efforts. It has been reported that a speed-up of up to 5 times can be achieved to build a dataless text classifier with indistinguishable performance to a supervised classifier, by assuming that labeling a word is 5 times faster than labeling a document [12]. These promising results suggest that dataless text classification is a practical alternative to the supervised approaches, when constructing the training documents is not an easy task. More importantly, the labeled documents produced by a dataless classifier can also be used as training examples to learn supervised text classifiers if necessary.

Human beings can quickly learn to distinguish whether a document belongs to a category, based on several relevant keywords about a category. This is because that people can learn to build

CIKM'16 , October 24-28, 2016, Indianapolis, IN, USA

DOI: http://dx.doi.org/10.1145/2983323.2983721

the relevance among the representative words of the category. For example, a human being can successfully identify a relevant word "wheel" to category *automobile*, after browsing several documents in category *automobile*, even if she does not know the meaning of word "wheel". The underlying reason is the high co-occurrence between "wheel" and other relevant words like "cars" and "engines". This relevance learning process is analogous to the unsupervised topic inference process of the standard LDA [3], a probabilistic topic model (PTM) that implicitly infers the hidden topics from the documents based on the higher-order word co-occurrence pattern [31]. However, conventional PTMs like PLSA and LDA are unsupervised techniques that implicitly infer the hidden topics based on word co-occurrences [3, 19]. It is difficult or even infeasible to classify or label documents in such a purely unsupervised manner.

Inspired by the recent success of the PTM-based dataless text classification techniques [7, 17, 18], in this paper, we propose a **S**eed-guided **T**opic **M**odel, named STM, for dataless text classification. Given a collection of unlabeled documents, STM is able to classify documents by taking only a few semantically relevant words for each category (called "seed words"). Compared to existing PTM-based dataless text classification models, the novelty of STM is two-fold:

- First, instead of simply associating a hidden topic with one or more categories directly, STM models two sets of topics: *category-topics* and *general-topics*. Each category-topic is associated with one specific category, and is assumed to represent the meaning of that category. The general-topics cover the general semantics of the whole document collection. In STM, each document is associated with a single category-topic and a mixture of general-topics. The posterior category-topic assignment is used to label the document. Although the modeling of general and specific aspects of documents was studied previously for information retrieval [6], modeling two sets of topics for dataless text classification has not been studied [7, 17, 18].

- Second, STM does not solely rely on the implicit word co-occurrence pattern to guide the category inference process. We estimate the probability of a word being generated by a category-topic by measuring its correlations to the seed words of the category. The estimation is based on the explicit word co-occurrence patterns derived from the document collection. We call the words that are generated by a category-topic *category words*. We also leverage the seed words to estimate the initial category distribution of each document for model initialization.

In summary, STM learns the category labels of documents in an unsupervised manner, just as what humans do in learning to classify documents with just few words: (i) first to identify the highly relevant documents based on the given seed words of a category; (ii) then based on these highly relevant documents, to collectively identify the category words in addition to the seed words; (iii) next to use both the seed words and category words to find new relevant documents and new category words; the last step repeats until a global equilibrium is optimized.

We conduct extensive experiments on two datasets Reuters-10 and 20-Newsgroup, and compare STM with state-of-the-art dataless text classifiers and supervised learning solutions. In terms of classification accuracy measured by F_1, our experimental results show that STM outperforms all the dataless competitors in all tasks and performs better than the supervised classifiers sLDA and SVM in a few tasks. We also conduct a comprehensive performance evaluation to analyze the impact of parameter settings in STM. The results show that the proposed STM is reliable to a broad range of parameter values, indicating its superiority in real scenarios.

2. RELATED WORK

Here, we review related work on dataless text classification and topic modeling with auxiliary knowledge.

Dataless Text Classifiers. As being the seminal work of dataless text classification, Liu *et al.* investigated the possibility of building a text classifier by simply employing few words relevant to each category in a semi-supervised manner, where these relevant words are used to bootstrap an initial set of training instances [22]. Then a semi-supervised naive Bayes classifier based on the Expectation Maximization algorithm (NB-EM) [27] is built based on the training instances. Similarly, Gliozzo *et al.* [14] proposed to build an initial set of training instances by using the Latent Semantic Analysis [10]. Then a support vector machine (SVM) classifier is trained based on these bootstrapped instances. Downey and Etzioni provided a theoretical analysis about the possibility of achieving accurate classification in the absence of training data [11]. Their analysis and empirical studies showed that the accurate text classification without the training data is possible under certain assumptions. Druck *et al.* proposed a maximum entropy based dataless text classifier which uses only the labeled words of each category, named GE-FL [12]. GE-FL was designed by assuming that the documents containing the seed words of a category are more likely to belong to this category. Hence, the parameters of GE-FL are optimized by minimizing the distance between the expected category distribution of the documents containing a labeled word under GE-FL and the corresponding reference category distribution of the labeled word. They showed that a speed-up of 5 times can be achieved by GE-FL with indistinguishable performance to an entropy regularization based semi-supervised (ER) method [15], given labeling a word is 5 times faster than labeling a document [28].

Chang *et al.* proposed a dataless text classification method by projecting each word and document into the same semantic space of Wikipedia concepts [5]. They represent each category with the words used in the category label. The similarity between a document and a category is measure by using Explicit Semantic Analysis (ESA) [13]. Recently, Song and Roth [29] studied the task of dataless hierarchical text classification by applying the work of [5]. Their experimental results showed that Wikipedia-based ESA still performs the best for this task. Since the large-scale knowledge base like Wikipedia is not always available for many languages or domains, this method may not be applicable in these cases. Note that the proposed STM does not rely on the external knowledge base at all. Instead, STM learns the discriminative category information by exploiting the semantic relevance of the seed words to the dataset itself, which can be applied in a much broader range of scenarios.

Several methods based on the standard LDA have been proposed for dataless text classification [7,17,18]. Hingmire *et al.* proposed a dataless text classifier model based on the LDA, named ClassifyLDA [18]. ClassifyLDA first infers the hidden topics by using LDA. Then, an annotator assigns a category to each topic. ClassifyLDA continues the topic inference process by aggregating the topics with the same category label as a single topic. The corresponding category of the topic with the maximum posterior topic proportion is used as the prediction. They showed that ClassifyLDA achieves almost comparable performance with a semi-supervised naive Bayes classifier (NB-EM) proposed in [27]. Hingmire and Chakraborti [17] proposed a new model (TLC) by extending ClassifyLDA, which allows to assign more than one category to a topic. Then, TLC was further enhanced to incorporate the relevant words of each category, called TLC++. TLC++ selects the most informative words by using

the information gain metric based on the initial category predictions from TLC. They found that TLC++ consistently outperforms ClassifyLDA, TLC and GE-FL by a comprehensive evaluation.

Chen *et al.* proposed a LDA based dataless classification model, called DescLDA [7]. DescLDA assumes that each category is associated with a fixed number of topics, and each topic is only associated with a single category. The selected semantically related words (called descriptive words) for each category are used to constrain the topic-word distribution such that these words have a higher probability under the associated topics. DescLDA has a tuning parameter, *i.e.,* the topic number for a category. However, DescLDA is very sensitive to this parameter. According to their experimental results, a significant performance degradation is experienced when a suboptimal number is used across the both datasets used in their work. Our proposed STM differs significantly from the above PTM-based solutions. While these methods associate all the topics to the categories, STM uses two separate sets of topics to represent the documents. While category-topics are responsible for extracting the discriminative category information, general-topics are used to organize the general semantic information underlying the whole dataset. Although STM also has a similar parameter T to specify the number of general topics. Our experimental results show that STM is very robust to this parameter setting. That is, little performance variations are observed for different T values across different datasets.

Topic Models with Auxiliary Knowledge. Different kinds of prior domain knowledge have been incorporated into PTMs to achieve better performance of different tasks. Mimno *et al.* exploited the corpus-specific word co-occurrence information to enhance the topic coherence of the standard LDA [25]. Besides exploiting the corpus-specific knowledge, many works have proposed to incorporate the semantical relations between word pairs into the topic model [1, 8, 9, 23]. The semantic relatedness information based on the learnt word embeddings over the large external corpus is incorporated for better short text topic modeling in [23]. A seeded topic model was proposed to extract the aspects and sentiments from the customer comments in [26], named SAS. SAS takes the seed words related to a specific aspect as a seed set, *e.g.,* words related to the aspect *room service*. Then an aspect is considered as a multinomial distribution over the non-seed words and the seed sets. Based on the implicit word co-occurrence information regarding these seed words, SAS can obtain a significant improvement in terms of aspect extraction accuracy. Similarly, Jagarlamudi *et al.* proposed a SeededLDA model to learn better topic-word and document-topic distributions with the seed words selected by using information gain from the labeled documents [20]. These works exploit the semantic guidance provided by the seed words in an implicit way, *i.e.,* the word co-occurrence information. Differing from these works, we employ an explicit strategy to estimate the initial document relevance and discriminate relevant words of a specific category. These prior knowledge extracted based on the seed words are then used directly to guide the topic inference process, leading to a promising classification performance. Later in Section 4.4, we will show that this strategy indeed brings significant improvement to the classification performance.

3. SEED-GUIDED TOPIC MODEL

In this section, we present the proposed STM model for dataless text classification in detail.

3.1 Model Overview

Given a collection of unlabeled documents and a few seed words for each category, the goal of STM is to label the documents through

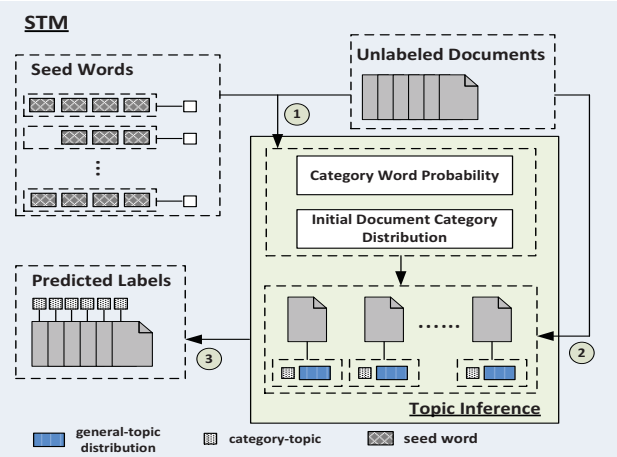


Figure 2: The architecture of the seed-word guided topic model.

topic influence. As mentioned in Section 1, one of the key differences between STM and existing models for dataless text classification is STM's capability in exploiting prior knowledge derived directly from the document collection. Illustrated in Figure 2, before we conduct topic influence, we first estimate the *category word probability* and the *initial document category distribution*, solely based on the given collection of documents and the seeds words.

Estimating Category Word Probability. In STM, we assume that each category is associated with one category-topic and a mixture of general-topics. A category word w for a category c is a word that is generated by c's category-topic.

Similar to the given seed words of c, category words are expected to represent the semantic meaning of category c. For this reason, we believe that category words could be semantically or statistically related to the seed words of the same category. However, without training documents labeled for a category, the statistically informative words could not be easily derived. Although semantically relevant words can be extracted based on the seed words and an external thesauri or knowledge base, such prior knowledge bases may not always be available. Here, we simply use word co-occurrences to estimate the probability of category words. If a word has high word co-occurrences with the seed words of a category, this word is more likely to be a category word. The degree of co-occurrences between a word w and a seed word s is measured by the conditional probability $p(w|s)$:

$$p(w|s) = \frac{df(w,s)}{df(s)} \tag{1}$$

where $df(s)$ is the number of the documents containing seed word s, and $df(w,s)$ is the number of the documents containing both word w and seed word s. Then, we calculate the relevance score $rel(w,c)$ and weight $\tau_{w,c}$ for each word w and category c as follows:

$$rel(w,c) = \frac{1}{|\mathbb{S}_c|}\sum_{s\in\mathbb{S}_c} p(w|s) \tag{2}$$

$$\nu(w,c) = \max\left(\frac{rel(w,c)}{\sum_c rel(w,c)} - \frac{1}{C}, 0\right) \tag{3}$$

$$\nu_c(w,c) = \frac{\nu(w,c)}{\sum_w \nu(w,c)} \tag{4}$$

$$\tau_{w,c} = \max\left(\frac{\nu_c(w,c)}{\sum_c \nu_c(w,c)}, \epsilon\right) \tag{5}$$

In Equation 2, $\mathbb{S}_c$ is the collection of seed words of category c. In Equation 3, we normalize the relevance score $rel(w, c)$ and subtract it by the average relevance score $1\backslash C$ for each category. It is expected that word w is a category word for category c only if w and c have a high $rel(w, c)$ value. Therefore subtracting the relevance scores by the average is necessary to filter out irrelevant categories using Equation 3. At this stage, we can simply take $\nu(w, c)$ as the final weight indicating the relevance between w and c. However, we observe that the absolute value of $\nu(w, c)$ does not really reflect the true relevance between word w and category c, because the values are largely affected by the statistical properties of the seed words as well.

Given a seed word s with very high document frequency (*i.e.,* $df(s)$ is large), $p(w|s)$ would be relatively smaller than other seed words with small document frequencies based on Equation 1. This results in relatively smaller $\nu(w, c)$ values for all words under category c. Hence, we take the impact of the seed words into account by normalizing $\nu(w, c)$ with respect to c by using Equation 4. Similarly, the $\nu_c(w, c)$ values for different word w could be in different scales. We further normalize $\nu_c(w, c)$ with respect to word w and take it as the final relevance weight $\tau_{w,c}$ between word w and category c, by using Equation 5.

A higher $\tau_{w,c}$ means that word w is more likely to be a category word of category c. A small constant ϵ is assigned for the words that have relatively low $rel(w, c)$ values to avoid being zero, and $\epsilon = 0.01$ in this work.

Initial Document Category Distribution Estimation. Since the seed words represent the meaning of a category, it is reasonable to assume that a document containing the seed words of a category is likely to belong to this category. Specifically, given a document d, we derive the initial category distribution η_d as follows:

$$f(d, c) = \sum_{s \in \mathbb{S}_c} tf(s, d) \quad (6)$$

$$\eta_d(c) = \frac{\ln(1 + f(d, c)) + \gamma}{\sum_c (ln(1 + f(d, c))) + C\gamma} \quad (7)$$

where $tf(s, d)$ is the term frequency of seed word s in document d, and γ is a prior parameter for the Dirichlet smoothing, $\gamma = 0.01$ in this work.

Category-topic and General-topic. Unlike the existing related works that directly use a topic to represent a distinct category [7, 17, 18], we assume that there are two kinds of topics underlying the document collection: category-topics and general-topics. As in PLSA and LDA, general-topics in STM are shared by all the documents and capture the global semantic information of the whole collection. In contrast, each category-topic is associated with a single category and be comprised by the relevant words of the category. Also, each category has only one category-topic. That is, the category and its related category-topic have an one-to-one correspondence in STM.[1] In this sense, each document is generated by a category-topic and all general-topics together.

Since general-topics are used in STM to model the semantic information besides the category information encoded by the category topics, it is expected that the documents of the same category could share similar document general-topic distributions to some extent. As mentioned above, each category in STM is associated with one category-topic and a mixture of general-topics. Let φ_c denotes the general-topic distribution associated with category c, θ_d denotes the general-topic distribution of document d. We assume the general-topic distribution θ_d is sampled from a Dirichlet prior with the concentration parameter α_1 and the category's φ_c as the base measure. This hierarchical Dirichlet prior setting allows the flexibility that the deviation of general-topic distributions of the documents under the same category is controlled by the concentration parameter α_1 [32]. Given C categories and T general topics, a word in a document can be either generated from category-topic c or a general-topic t, where category-topic c indicates that the document belongs to category c. The notations used in the rest of the paper are summarized in Table 1.

[1] Category and category-topic are considered equivalent and exchangable in this work when the context has no ambiguity.

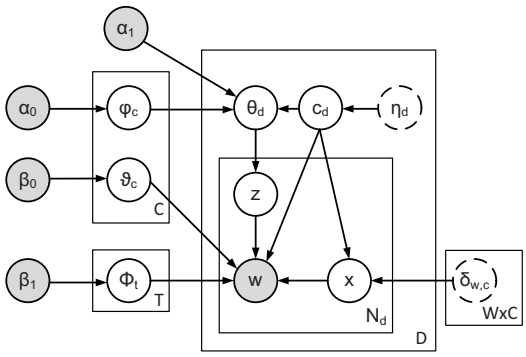


Figure 3: Graphical representation of STM

The graphical representation of STM is shown in Figure 3 and the generative process is described as follows:

1. For each category $c \in \{1...C\}$:
   (a) draw a general-topic distribution $\varphi_c \sim Dir(\alpha_0)$;
   (b) draw a word distribution $\vartheta_c \sim$ Dir(β_0);
2. For each general-topic $t \in \{1...T\}$:
   (a) draw $\phi_t \sim$ Dir(β_1);
3. For each document $d \in \{1...D\}$;
   (a) generate an initial category distribution η_d;
   (b) draw a category $c_d \sim$ Multi(η_d);
   (c) draw a general-topic distribution $\theta_d \sim$ Dir($\alpha_1 \cdot \varphi_{c_d}$);
   (d) for each word $i \in \{1...|d|\}$:
      i. draw $x_{d,i} \sim$ Bernoulli($\delta_{w_{d,i},c_d}$);
      ii. if $x_{d,i} = 0$ draw $w_{d,i} \sim$ Multi(ϑ_{c_d});
      if $x_{d,i} = 1$
      (1) draw $z_{d,i} \sim$ Multi(θ_d);
      (2) draw $w_{d,i} \sim$ Multi($\phi_{z_{d,i}}$);

The binary variable $x_{d,i} = 0$ indicates that the associated word $w_{d,i}$ is generated from category-topic c_d , otherwise word $w_{d,i}$ is generated from a specific general-topic $z_{d,i}$ (*i.e.,* $x_{d,i} = 1$). The discriminative power carried by the category words essentially determines the classification accuracy of STM. In STM, the chance of choosing an occurrence of word w as a category word for category c relies on the prior knowledge constructed by using the seed words, *i.e.,* a category word probability $\delta_{w,c}$ that word w is a category word ($x = 0$) under category c.

Initially, it is simple to assume that each word has the same probability to be a category word for each category. Hence, we can refer a global parameter ρ as the probability that a word is picked as a category word for any category and $1 - \rho$ as the probability that a

Table 1: List of Notations

D	the total number of documents in the dataset
C	the total number of categories/category-topics in the dataset
T	the total number of general-topics
W	the size of the vocabulary
s	a seed word of a category
$\mathbb{S}_c$	a set of seed words of category c
c_d	the category topic assignment for document d
$w_{d,i}$	the observed word at position i in document d
$z_{d,i}$	the general-topic assignment for word $w_{d,i}$
$x_{d,i}$	indicator about whether $w_{d,i}$ is generated by a category-topic
η_d	the initial category distribution of document d
θ_d	the general-topic distribution of document d
φ_c	the prior general-topic distribution of all documents of category c
ϕ_t	the word distribution of general-topic t
ϑ_c	the word distribution of category-topic c
α_1	the concentration parameter of φ_c for document's θ_d of category c
$\delta_{w,c}$	the probability of word w being a category word for category c
$\tau_{w,c}$	the relevance weight between word w and category c
ρ	the tuning parameter for category word probability $\delta_{w,c}$
$\alpha_0, \beta_0, \beta_1$	Dirichlet Priors

word is picked as being from a general-topic, *i.e.,* $\delta_{w,c} = \rho$. Note that the category-topics are mainly governed by the higher-order word co-occurrence patterns. Therefore, the words that frequently co-occur with each other under the documents of the same category can be grouped together in the corresponding category-topic. Because STM is a probabilistic topic model, a wrong category-topic may be sampled for some documents. Given the underlying collection is severely imbalanced, the documents of the largest category could be allocated with a wrong category-topic. The smaller categories will be dominated by these documents. The resultant incorrect category-topics of the smaller categories in turn will hurt the classification performance very much.

By considering the weight $\tau_{w,c}$ given in Equation 5 based on the word co-occurrences between word w and the seed words of category c , we can refine the category word probability $\delta_{w,c}$ of word w and category c as follows:

$$\delta_{w,c} = \frac{\tau_{w,c}\rho}{1 - \rho + \tau_{w,c}\rho} \tag{8}$$

In Equation 8, ρ becomes a tuning parameter within $[0, 1]$, specifying the importance of $\tau_{w,c}$ for $\delta_{w,c}$. When $\rho = 0$ (*i.e.,* $\delta_{w,c} = 0$), STM is downgraded to the standard LDA and classify each document based on the general-topic distribution of the document only. When $\rho = 1$ (*i.e.,* $\delta_{w,c} = 1$), STM consists of only category-topics, and all word occurrences are assigned to some category, which is equivalent to TLC++ model proposed in [17] with the exception that the association between the topic and category is made by the seed words here instead of manual labeling as in the TLC++. Since the variables η_d and $\delta_{w,c}$ are estimated based on the seed words of STM, we therefore plot these two sets of variables in dotted circle in Figure 3.

3.2 Inference and Parameter Estimation

In this section, we describe the algorithm to infer the hidden parameters $\{\varphi_c, \vartheta_c, \phi_t, \theta_d, z_{d,i}, x_{d,i}, c_d\}$ by using Gibbs Sampling.

Inference by using Gibbs sampling. As with LDA and other PTMs, the exact inference of STM is intractable. We therefore utilize the Gibbs Sampling to perform the approximate inference and parameter learning [31]. Specifically, we construct a Markov chain on latent parameters. At each step, a latent parameter or a set of latent parameters are sampled based on the conditional probability given the values of other parameters. In STM, because parameters $z_{d,i}$ and $x_{d,i}$ are correlated, we jointly sample their values as follows:

$$p(z_{d,i}, x_{d,i} | \mathbf{z}_{\neg(d,i)}, \mathbf{x}_{\neg(d,i)}, \mathbf{c}, \mathbf{w}) \propto \begin{cases} \phi^{w_{d,i}}_{t,\neg(d,i)} \times \theta^{t}_{d,\neg i} \times (1 - \delta_{w_{d,i},c_d}) & z_{d,i} = t, x_{d,i} = 1 \\ \vartheta^{w_{d,i}}_{c_d,\neg(d,i)} \times \delta_{w_{d,i},c_d} & x_{d,i} = 0 \end{cases} \tag{9}$$

where $\phi^{w_{d,i}}_{t,\neg(d,i)}$ is the probability of seeing $w_{d,i}$ under general-topic t excluding the current assignment, $\theta^{t}_{d,\neg i}$ is the probability of seeing topic t in document d excluding the current assignment, and $\vartheta^{w_{d,i}}_{c_d,\neg(d,i)}$ is the probability of seeing word $w_{d,i}$ under category-topic c_d excluding the current assignment.

The associated category-topic is sampled based on the conditional probability presented in Equation 10, where n^w_t is the number of times word w is assigned to general-topic t, n^t_d is the number of words that are assigned to general-topic t within document d, n^w_c is the number of times word w is assigned to category-topic c. Symbol $\neg d$ means that document d is excluded from the count.

$$\begin{aligned} p(c_d = c | \mathbf{z}, \mathbf{x}, \mathbf{c}_{\neg d}, \mathbf{w}) \propto & \prod_{t=1}^{T} \frac{\prod_{w \in d}[(n^w_t + \beta_1 - 1) \cdots (n^w_{t,\neg d} + \beta_1)]}{[\sum_{w=1}^{W}(n^w_t + \beta_1) - 1] \cdots [\sum_{w=1}^{W}(n^w_{t,\neg d} + \beta_1)]} \\ & \times \prod_{t=1}^{T} \frac{(n^t_d + \alpha_1 \cdot \varphi^t_c - 1) \cdots (\alpha_1 \cdot \varphi^t_c)}{[\sum_{k=1}^{T}(n^k_d + \alpha_1 \cdot \varphi^k_c) - 1] \cdots [\sum_{k=1}^{T} \alpha_1 \cdot \varphi^k_c]} \\ & \times \frac{\prod_{w \in d}[(n^w_c + \beta_0 - 1) \cdots (n^w_{c,\neg d} + \beta_0)]}{[\sum_{w=1}^{W}(n^w_c + \beta_0) - 1] \cdots [\sum_{w=1}^{W}(n^w_{c,\neg d} + \beta_0)]} \eta_d(c) \end{aligned} \tag{10}$$

The category general-topic distribution φ_c, document general-topic distribution θ_d, word-distribution ϑ_c of category c and topic-word distribution ϕ_t of general-topic t can be computed as follows:

$$\varphi^t_c = \frac{n^t_c + \alpha_0}{\sum_{k=1}^{T}(n^k_c + \alpha_0)} \tag{11}$$

$$\theta^t_d = \frac{n^t_d + \alpha_1 \cdot \varphi^t_{c_d}}{\sum_{k=1}^{T}(n^k_d + \alpha_1 \cdot \varphi^t_{c_d})} \tag{12}$$

$$\vartheta^w_c = \frac{n^w_c + \beta_0}{\sum_{w'=1}^{W}(n^{w'}_c + \beta_0)} \tag{13}$$

$$\phi^w_t = \frac{n^w_t + \beta_1}{\sum_{w'=1}^{W}(n^{w'}_t + \beta_1)} \tag{14}$$

where n^t_c is the total number of words that are assigned to general-topic t within the documents of category-topic c.

Because all the pairs of $z_{d,i}$ and $x_{d,i}$ of document d are affected by the choice of category-topic c_d and vice versa (Equations 9 and 10), the sampling order of $z_{d,i}, x_{d,i}$ and c_d becomes a critical factor. Simply sampling a new c_d based on Equation 10 with all $z_{d,i}, x_{d,i}$ values conditioned on the previous c_d could not converge to the true posterior distribution. This is a common issue of Markov

Chain Monte Carlo (MCMC) methods, called autocorrelation phenomenon [30]. Here, to avoid the autocorrelation, at the first step we sample each pair of $z_{d,i}, x_{d,i}$ conditioned on each possible c. Then, the likely c_d is sampled conditioned on all the corresponding $z_{d,i}, x_{d,i}$ values with Equation 10. Afterwards, all the values of $z_{d,i}, x_{d,i}$ are set to the updated c_d's corresponding values sampled in the first step. For document classification, the category corresponding to the sampled c_d of document d at the last iteration is assigned to the document as the category prediction by STM.

4. EXPERIMENT

In this section, we evaluate the classification performance of the proposed STM[2] against other state-of-the-art dataless text classification methods. For the completeness of the comparison, we also report the results obtained from supervised learning methods, as well as the results obtained through supervised learning methods trained using the documents labeled by STM. We then analyze the impact of parameter settings in STM. Our experimental results show that STM outperforms all existing state-of-the-art competitors and is very robust to the parameter settings and the seed words provided.

4.1 Datasets

20 Newsgroup (20NG): The 20NG is a widely used dataset[3] for document classification research [5, 7, 16, 21, 33]. It contains approximately $20,000$ newsgroup documents, evenly distributed across 20 different newsgroups/categories. We use the *bydate* version of the 20NG dataset, where a total of $18,846$ documents are divided into a training set (60%) and a test set (40%). These 20 categories can be further aggregated into 6 major categories. For example, the major category sci consists of 4 categories: sci.crypt, sci.electronics, sci.med, sci.space. This dataset has been previously used in the related works [5, 7, 12, 17, 18, 29]. When parsing the documents, we keep the text contained in the "Subject", "Keywords", and "Content" fields. The information in the other fields and email addresses are filtered out.

Reuters-10: Reuters-21578 is also a widely used dataset for document classification. It contains $21,578$ documents in 135 categories. Among them, $13,625$ and $6,188$ documents are in the training set and test set respectively. This dataset is very imbalanced and the variation of category size is quite large. We used the 10 largest categories (hence denoted by Reuters-10) in the dataset with Aptè split[4]. We further discard the documents belonging to more than one category. This left us with a total of $7,285$ documents: $5,228$ documents in train and $2,057$ documents in test set. The same subset, Reuters-10, has been previously used in the related works as well [7, 33].

For both datasets, we further remove the stop words, the words shorter than 2 characters, and the words appear in fewer than 5 documents. The data statistics after preprocessing are reported in Tables 2 and 3, respectively. The statistics of the 5 major categories of 20NG dataset used in the experiments are reported in the last 5 rows in Table 2. Observe from Table 3 that the Reuters-10 is very imbalanced. While most categories have around $100 - 300$ documents, the two largest categories have $3,713$ and $2,055$ documents respectively.

4.2 Experimental Setup

Parameter Setting. In STM, there are several hyper-parameters. They are set to typical values: $\alpha_0 = 50/T$, $\beta_0 = \beta_1 = 0.01$, as used in PTM studies [31]. We set $\gamma = \epsilon = 0.01$ in our experiments. As to the tunable parameters in STM, we use the setting: $\alpha_1 = 100, \rho = 0.85, T = 3 \cdot C$ in the evaluation. We run STM for 50 iterations, and then the category-topic assigned to a document during the last iteration is taken as its predicted label. The reported classification results of STM is the average results over 10 runs with random model initialization, the same setting used in TLC++ [17].

Table 2: Statistics of 20NG dataset. #train/#test: the number of training/test documents; Avg($|d|$): the average length of the documents; $|\mathbb{S}^L|$ / $|\mathbb{S}^D|$: the number of seed words obtained from the category label $\mathbb{S}^L$, and from category description $\mathbb{S}^D$.

Category label	#train	#test	Avg($\|d\|$)	$\|\mathbb{S}^L\|$ / $\|\mathbb{S}^D\|$
alt.atheism	480	319	157.30	1/5
comp.graphics	584	389	135.58	2/5
comp.os.ms-windows.misc	591	394	226.08	4/8
comp.sys.ibm.pc.hardware	590	392	95.10	5/7
comp.sys.mac.hardware	578	385	86.98	4/3
comp.windows.x	593	395	146.81	3/5
misc.forsale	585	390	77.27	1/8
rec.autos	594	396	103.10	1/7
rec.motorcycles	598	398	95.41	1/3
rec.sport.baseball	597	397	113.07	2/3
rec.sport.hockey	600	399	144.714	2/3
sci.crypt	595	396	164.34	2/5
sci.electronics	591	393	97.60	2/5
sci.med	594	396	141.16	2/5
sci.space	593	394	148.34	2/6
soc.religion.christian	599	398	175.13	3/6
talk.politics.guns	546	364	166.24	2/5
talk.politics.mideast	564	376	243.53	2/5
talk.politics.misc	465	310	210.61	1/3
talk.religion.misc	377	251	165.61	1/6
politics	1575	1050	207.02	3/13
religion	1456	968	166.79	4/13
sci	2373	1579	137.92	5/31
comp	2936	1955	138.38	13/27
rec	2389	1590	114.10	4/17

Table 3: Statistics of the Reuters-10 dataset.

Category label	#train	#test	Avg($\|d\|$)	$\|\mathbb{S}^L\|$ / $\|\mathbb{S}^D\|$
acq	1,435	620	68.13	1/9
coffee	89	21	109.54	1/4
crude	223	98	118.10	1/9
earn	2,637	1,040	57.97	1/7
gold	70	20	78.36	1/4
interest	140	57	89.23	1/8
money-fx	176	69	99.70	2/9
ship	107	35	77.94	1/8
sugar	90	24	104.08	1/2
trade	225	73	130.02	1/7

Methods in Comparison. The proposed STM is compared against the following state-of-the-art *dataless text classification* methods and *supervised classification* methods:

Topic Labeled Classification with Labeled Words (TLC++) This method learns to classify documents based on the posterior topic proportions and the category labels of the topics [17]. The labels of the topics are annotated manually based on the highly probable words in each topic. We present the 30 most probable words in each topic for topic labeling, the same setting recommended by its authors.

[2] Our implementation is at https://github.com/ly233/Seed-Guided-Topic-Model

[3] http://qwone.com/~jason/20Newsgroups/

[4] http://kdd.ics.uci.edu/database/reuters21578/reuters21578.html

Generalized Expectation with Feature Labels (GE-FL) It learns a maximum entropy based text classifier by using the labeled words in each category as soft constraints [12]. Here, we use the same seed words that are used for STM as labeled words of each category for fair comparison. We used the implementation of GE-FL that is provided in the MALLET toolkit.[5]

Descriptive LDA (DescLDA) This method learns the category label of a document by applying document clustering over the learned hidden topics [7]. The hidden topic distributions are inferred based on the seed words. DescLDA has a tunable parameter: the number of topics. We report the best results with the optimal setting obtained in our experiments.

Seed-based NB-EM (SNB-EM) It learns dataless text classifier in a semi-supervised manner [22], where NB-EM method [27] is used for model building. We report the best performance with the optimal parameter settings obtained in our experiments. For fair comparison, we use the same seed words that are used for STM to build the initial training instances.

Support Vector Machines (SVM) This is a state-of-the-art supervised learning technique for text classification. We train a linear SVM classifier by using LIBSVM with the default parameter settings and TF-IDF weighting scheme.[6]

sLDA is a supervised text classifier based on the LDA model [2]. We train the model by using the implementation provided by the authors.[7] The best results obtained in our experiments are reported.

Note that, Chang *et al.* learns the category label of a document by projecting the document and the category into the same semantic space of Wikipedia concepts [5]. The nearest-neighbor based explicit semantic analysis (NN-ESA) is then used for the classification. Since NN-ESA involves parsing the whole Wikipedia, we choose not to include this method for comparison. Nevertheless, it was reported that DescLDA significantly outperform NN-ESA in earlier study [7].

Performance Metric. In the experiments, we use the standard training/test partitions of the two datasets for the evaluation. For all the dataless classifiers, the classifiers run over both the training and test documents as a single collection of unlabeled documents (*i.e.,not using their labels*). The classification accuracy are evaluated based on the labels of documents in test set, for fair comparison with the supervised methods. For supervised methods, the classifiers are developed using the training documents and then evaluated on the test set, as per normal.

For performance comparison, we report macro-averaged F_1 scores (Macro-F_1) [24]. Macro-F_1 is the averaged F_1 scores of all categories. We report the average results over 10 runs for all the methods (excluding SNB-EM and SVM). The statistical significance is conducted by using the student *t-test*.

Seed Words Selection. The quality of the seed words is a critical factor for all dataless classifiers: STM, GE-FL, DescLDA, and SNB-EM. Here, we exploit two sets of the seed words selected from the category label (denoted by $\mathbb{S}^L$) and description (denoted by $\mathbb{S}^D$) respectively. Category label means that the seed words are extracted from the label in the given dataset directly. For example, from the category label comp.sys.ibm.pc.hardware in 20NG, five seed words "computer, systems, ibm, pc, hardware" are extracted as $\mathbb{S}^L$. Note that the semantically irrelevant words in the label are excluded here. For example, "talk" is excluded from category talk.politics.guns. The seed words in $\mathbb{S}^D$ are compiled manually with the domain knowledge. For example, the authors of DescLDA followed the labeling procedure used in TLC++ [17] (*i.e.,* assisted by standard LDA) to compile the $\mathbb{S}^D$. The two sets of seed words used here are both used in earlier studies [5, 7, 29]. The details about these seed words can be found in the work of DescLDA [7]. The number of seed words in $\mathbb{S}^L$ and $\mathbb{S}^D$ in each category are listed in Tables 2 and 3.

[5] http://mallet.cs.umass.edu

[6] www.csie.ntu.edu.tw/~cjlin/liblinear

[7] http://www.cs.cmu.edu/~chongw/slda/

4.3 Performance Comparison

We evaluate all the methods on both datasets: 20 categories in the 20NG dataset and 10 categories in the Reuters-10 dataset. We further create 7 classification tasks based on the 20NG dataset, by selecting the documents in subsets of categories. For example, one of the tasks is to classify documents in categories pc and mac, denoted by pc-mac. These 7 classification tasks were used in the work of TLC++ for the evaluation [17]. In total, we have 9 classifications tasks involving different number of categories on the two datasets. Table 4 reports the Macro-F_1 scores of these methods on the 9 tasks. We make the following observations:

First, among the dataless classifiers, STM +$\mathbb{S}^D$ significantly outperforms other state-of-the-art alternatives on 8 out of 9 classification tasks. SNB-EM+$\mathbb{S}^D$ performs the second best on 5 classification tasks, followed by STM +$\mathbb{S}^L$ on the other 3 tasks. The superiority of SNB-EM+$\mathbb{S}^D$ is attributed to its semi-supervised nature. After an initial NB-EM classifier is built based on the seed words, SNB-EM retrains itself by using the classification results of high confidence, and this procedure repeats until the probability parameters stabilize. TLC++ outperforms GE-FL on most tasks. This is reasonable since each topic in TLC++ is manually examined and labeled based on its 30 most probable words, which is equivalent to labeling 30 relevant words for each topic. It is expected that more human efforts leads to better classification accuracy, all else being equal. In [17], TLC++ is proven to be superior to GE-FL, which is consistent with our finding.

Second, all the dataless classifiers obtain relatively poorer results when all the categories are used in the classification tasks, *i.e.,* 20 categories on 20NG and 10 categories on Reuters-10. STM +$\mathbb{S}^D$ achieves the best performance over the other alternatives in these two tasks. Although TLC++ outperforms GE-FL in the other tasks, GE-FL performs much better here. We observe that when the number of categories is larger, the resultant topics often carry mixed semantic information. It even becomes difficult for annotators to manually associate topics to the relevant categories for TLC++. In this sense, TLC++ experiences a significant performance deterioration. Over all on these 9 classification tasks, STM +$\mathbb{S}^D$ outperforms GE-FL+$\mathbb{S}^D$, DescLDA+$\mathbb{S}^D$, SNB-EM+$\mathbb{S}^D$, and TLC++ by around $3.3-45.2\%$, $4.3-86.1\%$, $0.9-9.0\%$ and $2.7-64.8\%$ respectively.

Third, the supervised learning methods SVM and sLDA may not always perform better than the dataless counterparts. Dataless classifier outperforms SVM in 3 out of 9 classification tasks: DescLDA+$\mathbb{S}^D$ on med-space, STM +$\mathbb{S}^D$ on both pc-mac and autos-motorcycles-baseball-hockey. Moreover, STM +$\mathbb{S}^D$ consistently outperform sLDA on all the classification tasks. Even STM +$\mathbb{S}^L$ performs significantly better than sLDA on 7 out of 9 classification tasks, although sLDA is a supervised classification method. Specifically, sLDA only achieves a Macro-F_1 of 0.735 on *pc-mac*, where a large number of words are shared by the two categories [4]. However, STM +$\mathbb{S}^D$ outperforms SVM on this task by 0.018, given that 1,168 labeled documents are used to train the SVM classifier. Further, STM +$\mathbb{S}^D$ achieves very close performance with SVM on 4 clas-

Table 4: Macro-F_1 of the 7 methods on all tasks. The best and second best results by dataless classifiers are highlighted in boldface and underlined respectively, on each task. † indicates that the difference to the best result is statistically significant at 0.05 level.

Dataset	Classification task	STM		GE-FL		DescLDA		SNB-EM		TLC++	sLDA	SVM	STM+$\mathbb{S}^D$ →SVM
		$\mathbb{S}^L$	$\mathbb{S}^D$	$\mathbb{S}^L$	$\mathbb{S}^D$	$\mathbb{S}^L$	$\mathbb{S}^D$	$\mathbb{S}^L$	$\mathbb{S}^D$				
20NG	med-space	0.964	0.966	0.712†	0.935†	0.877†	**0.977**	0.897	0.967	0.938†	0.910†	0.976	0.953
	pc-mac	0.898†	**0.943**	0.491†	0.705†	0.688†	0.694†	0.895	0.876	0.685†	0.735†	0.925	0.905
	politics-religion	0.907†	**0.952**	0.684†	0.883†	0.888†	0.900†	0.894	0.939	0.911†	0.925†	0.954	0.942
	politics-sci	0.960	**0.962**	0.750†	0.889†	0.624†	0.912†	0.846	0.941	0.906†	0.930†	0.971	0.956
	comp-religion-sci	0.918	**0.927**	0.709†	0.828†	0.559†	0.498†	0.907	0.919	0.817†	0.900†	0.936	0.918
	politics-rec-religion-sci	0.919†	**0.941**	0.719†	0.827†	0.514†	0.782†	0.768	0.917	0.834†	0.823†	0.941	0.928
	autos-motorcycles-baseball-hockey	0.916†	**0.977**	0.849†	0.673†	0.531†	0.713†	0.715	0.938	0.734†	0.894†	0.957	0.952
	All 20 categories	0.662†	**0.739**	0.320†	0.590†	0.632†	0.663†	0.461	0.678	0.510†	0.633†	0.820	0.710
Reuters-10	All 10 categories	0.694†	**0.822**	0.667†	0.776†	0.317†	0.800†	0.529	0.778	0.506†	0.754†	0.932	0.909

sificaiton tasks: politics-religion, politics-sci, comp-religion-sci, politics-rec-religion-sci. Although SVM performs better than STM +$\mathbb{S}^D$ when all the categories are used in the classification tasks on both datasets, the performance gap is not very large. Shown in Tables 2 and 3, only a small number of seed words are used to "train" the STM classifiers. Compared with the much larger number of labeled documents used to train SVM, minimum human efforts are necessary to learn dataless classifiers like STM. In this sense, the proposed STM can work as an important complement to the existing supervised solutions.

Fourth, with $\mathbb{S}^D$, STM, GE-FL, DescLDA and SNB-EM all perform better than their counterparts with $\mathbb{S}^L$. This is expected because more semantic information could be exploited by providing more seed words. The performance gain by using $\mathbb{S}^D$ over $\mathbb{S}^L$ is about $0.1 - 20.2\%$, $15.0 - 84.4\%$, $0.9 - 152.4\%$, and $1.3 - 47.1\%$ for STM, GE-FL, DescLDA, and SNB-EM respectively. Observe that the performance gap between STM +$\mathbb{S}^D$ and STM +$\mathbb{S}^L$ is the smallest, compared to the other alternatives. Moreover, STM +$\mathbb{S}^L$ outperforms GE-FL+$\mathbb{S}^L$/$\mathbb{S}^D$, DescLDA+$\mathbb{S}^L$, SNB-EM+$\mathbb{S}^L$ on 8 classification tasks, and outperforms TLC++, DescLDA+$\mathbb{S}^D$ on 8 and 6 tasks respectively. As reported in the last columns of Tables 2 and 3, almost all categories studied here have fewer than 5 $\mathbb{S}^L$ seed words, except for the major category *comp* which has 13 $\mathbb{S}^L$ seed words. With just a few semantically relevant words, STM can deliver promising classification performance. The superiority of STM +$\mathbb{S}^L$ suggests that STM is not very sensitive to the number of seed words.

We further conduct a set of experiments by training a SVM classifier based on the classification results of STM +$\mathbb{S}^D$ over the training set (denoted as STM +$\mathbb{S}^D$ →SVM in Table4). Specifically, we rank each prediction of the document in the training set by taking the ratio of $p(c_d = c_1)$ to $p(c_d = c_2)$, where c_1 and c_2 denote the most and the second most probable category respectively. Then, the top 90% of the classified documents from the original training set are used to train a SVM classifier. The last column in Table 4 lists the average Macro-F_1 scores on the 9 classification tasks. Observe that STM +$\mathbb{S}^D$ →SVM achieves lower performance than SVM trained by using the groundtruth labels. This is reasonable since the classification results of STM contain false positive instances. We also observe that STM +$\mathbb{S}^D$ →SVM performs slightly worse than STM +$\mathbb{S}^D$ on all the tasks except for 10 categories on Reuters-10. On this task, STM +$\mathbb{S}^D$ →SVM obtains large improvement over STM +$\mathbb{S}^D$. Through this set of experiments, we show that, without labeling a large number of documents, STM +$\mathbb{S}^D$ can be used as an automatic document labeling tool to train traditional supervised learning classifiers. This is important in the scenario where the existing system only requires labeled documents rather than classifiers. It is also important when the documents to be classified are from a document stream.

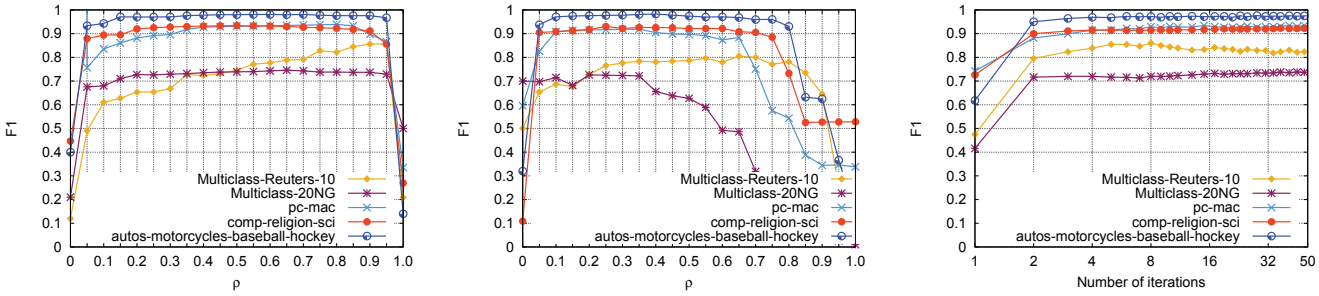
Overall, the experimental results show that STM can successfully learn the correct category labels of the documents with a few seed words from either the category label or its descripton.

4.4 Analysis of STM

We now investigate the impact of the parameter settings (*i.e.,* $\rho, T, \alpha_1, \tau_{w,c}$) in STM to its classification performance by using the $\mathbb{S}^D$ setting. We also study the convergence rate of STM in terms of classification performance. Specifically, we report the experimental results by varying one specific parameter value on these 5 classification tasks: pc-mac, comp-religion-sci, autos-motorcycles-baseball-hockey, Multiclass-20NG, and Multiclass-Reuters-10. The last two tasks use all the 20 and 10 categories in the corresponding dataset. Similar performance patterns are also observed for other classification tasks studied in Section 4.3. Note that when studying a specific parameter, we set the other parameters to the values used in Section 4.3.

Impact of ρ value. Recall in Equation 8, ρ controls the importance of the word co-occurrence information between the words and the seed words. As discussed in Section 3.1, $\rho = 0$ is equivalent to using a standard LDA with T general-topics. STM only relies on the general-topic distribution θ_d and category general-topic distribution φ_c to classify document d in this setting. On the other hand, STM has no general-topic at all when $\rho = 1$, and is equivalent to TLC++ model proposed in [17]. The only difference is that the category-topic is solely guided by the seed words in this setting. Figure 4(a) plots the performance patterns when using different ρ values in the range of $[0, 1]$ with a step of 0.05.

Observe that STM achieves stable performance on 4 classification tasks: Multiclass-20NG, pc-mac, comp-reigion-sci, autos-motorcycles-baseball-hockey, in the range of $[0.4, 0.85]$. On the task of *Multiclass-Reuters-10*, STM also achieves stable performance in the range of $[0.8, 0.95]$. Since Reuters-10 is very imbalanced, the documents of the larger categories could carry more diversity in their semantics, leading to certain semantic overlap between a large category and other categories. In this sense, some statistically relevant words of a large category could be also relevant to the other categories to some extent, especially the smaller categories. The relevant words become less discriminative enough for the large category. Given the difference in the sizes of the categories, many documents of the large category can be wrongly assigned due to



(a) ρ values (b) ρ values ($\tau_{w,c} = 1$) (c) number of iterations

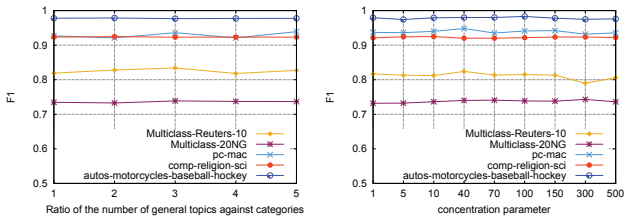
Figure 4: Performance of STM with different ρ values, different ρ values when $\tau_{w,c} = 1$, and different number of iterations.

the probabilistic sampling nature of the topic model. We therefore need a larger ρ to make the relevant words of the large category more discriminative. Another observation is that STM performs significantly worse when $\rho < 0.3$ or $\rho > 0.9$ on most tasks. That is, without general-topics ($\rho = 1$) or category-topics ($\rho = 0$), STM loses its ability to learn the discriminative information for each category. This validates the legitimacy of using both general-topics and category-topics in STM for dataless text classification. Note that the optimal ρ value is almost identical across the datasets. This is essentially valuable for real applications. In our experiments, we use $\rho = 0.85$.

Impact of $\tau_{w,c}$ value. Recall in Equations 2-5 and 8, a higher $\tau_{w,c}$ indicates that word w is more likely to be a category word for category c. If we do not explicitly estimate $\tau_{w,c}$ and set this value to 1.0 in Equation 8, STM simply relies on the global probability ρ and the implicit word co-occurrence pattern to do the category word selection. Figure 4(b) plots the performance with different ρ values when $\tau_{w,c}$ is set to 1.0. Observe that the optimal ρ values shifts from 0.85 to 0.35. Also, the range of the optimal ρ values shrinks significantly on each task. For example, the optimal range of ρ values is $[0.2, 0.35]$ for the multiclass task on 20NG, compared to the range of $[0.4, 0.85]$ observed in Figure 4(a). Furthermore, we observe that the classification performance deteriorates as well. The optimal Macro-F_1 drops from 0.856 to 0.800 on the Reuters-10 dataset. This confirms that estimating the category word probability to discriminate the relevant words of each category brings significant benefit for the classification performance.

Impact of T value. The T value specifies the number of general-topics used in STM. In the above experiments, we fix T to be 3 times of category-topics. Here, we evaluate the effect of T value by setting it to be 1 to 5 times of the number of category-topics. Figure 5(a) plots the performance of STM with different ratio of the number of general-topics against category-topics . We observe that STM is insensitive to the choice of general-topic number T. There is little performance variations in STM when we use different T values across the datasets. On the contrary, the experimental results reported in [7] demonstrate that DescLDA is severely affected by the choice of topic number. We also observe the significant performance degradation in the experiments when the suboptimal value of the topic number is evaluated. We set $T = 3 \cdot C$ for STM in our experiments.

Impact of α_1 value. The concentration parameter α_1 controls the degree that the general-topic distribution θ_d of document d of category c can deviate from the general-topic distribution φ_c of that category. When α_1 is very large, each document of category c has almost the identical general-topic distribution θ_d. On the other hand, $\alpha_1 \to 0$ is equivalent to assigning each document a general-topic distribution without the category constraint. Here, we investigate the performance of STM by varying α_1 values in the range of $[1, 500]$. The experimental results are plotted in Figure 5(b). It shows that the performance of STM is stable in a broad range of α^1 values across the datasets. Most classification tasks experience very little performance fluctuation when using different α_1 values. It seems that the category constraint is less important. However, when α_1 is set to 1, we observe significant performance degradation with varying ρ values (results not shown). This suggests that the category constraint is indeed helpful in STM. Based on the results, we set $\alpha_1 = 100$ in our experiments.



(a) T values (b) α_1 values

Figure 5: Performance of STM with varying T and α_1.

Impact of the number of iterations. We like to investigate the impact of the number of iterations to the classification performance of STM. Figure 4(c) plots the performance of STM with different number of iterations. STM can achieve very good performance after only 2 iterations. The stable performance is reached with about 5 iterations on all the 5 different tasks. This suggests that STM can successfully exploiting the semantic information provided by the seed words in an efficient manner, just like what humans are capable of. We further investigate the impact of estimating the initial category distribution (Equation 7) for each document based on the seed words. We find that the classification performance is not affected if this initial distribution estimation is not provided. However, STM takes more iterations to achieve the stable performance. Due to page limit, we do not show these results.

Recall that each category is associated with a single category-topic in STM. It is easily to extend STM to accommodate with more than one category-topic for a category. However, we observe that taking more than one category-topic for a category results in classification performance deterioration on almost all the tasks to some

degree. This suggests that a single category-topic for each category is competent to discriminate the semantic information of different categories.

5. CONCLUSION

In this paper, we propose a seed-guided topic model for dataless text classification, named STM. Without any labeled documents, STM takes only a few seed words to label documents through topic influence. By modeling the documents using both category-topics and general-topics, STM successfully captures the diverse semantic information of the dataset by separating category-specific information and general semantic information. We develop a strategy to extract the discriminative category information by explicitly calculating the relevance between the seed words and regular words based on the word co-occurrences. The experimental results show that STM outperforms existing state-of-the-art dataless text classifiers. It is interesting to observe that STM even surpasses the state-of-the-art supervised classifier like sLDA and SVM on several tasks, including the difficult classification tasks like pc-mac. We also empirically validate the robustness of STM against the different parameter settings. Nevertheless, there is still room to improve our model in several directions. For example, we could extend STM to enable semi-supervised learning. It is also interesting to check whether STM can automatically adjust the seed words and re-estimate the category word probability in the paradise of the semi-supervised learning. STM is designed to label each document a single category label. Extending STM for multi-label classification is also a part of our future work.

Acknowledgements. This research was supported by National Natural Science Foundation of China (No. 61502344), Natural Science Foundation of Hubei Province (No. 2015CFB337), Natural Scientific Research Program of Wuhan University (No. 2042015kf0014), and Singapore Ministry of Education Academic Research Fund Tier 2 (MOE2014-T2-2-066). Chenliang Li is the corresponding author.

6. REFERENCES

[1] D. Andrzejewski, X. Zhu, and M. Craven. Incorporating domain knowledge into topic modeling via dirichlet forest priors. In *ICML*, 2009.

[2] D. M. Blei and J. D. McAuliffe. Supervised topic models. In *NIPS*, 2007.

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

[4] S. Chakraborti, U. C. Beresi, N. Wiratunga, S. Massie, R. Lothian, and D. Khemani. Visualizing and evaluating complexity of textual case bases. In *ECCBR*, 2008.

[5] M. Chang, L. Ratinov, D. Roth, and V. Srikumar. Importance of semantic representation: Dataless classification. In *AAAI*, 2008.

[6] C. Chemudugunta, P. Smyth, and M. Steyvers. Modeling general and specific aspects of documents with a probabilistic topic model. In *NIPS*, 2006.

[7] X. Chen, Y. Xia, P. Jin, and J. A. Carroll. Dataless text classification with descriptive LDA. In *AAAI*, 2015.

[8] Z. Chen and B. Liu. Mining topics in documents: standing on the shoulders of big data. In *SIGKDD*, 2014.

[9] Z. Chen, A. Mukherjee, B. Liu, M. Hsu, M. Castellanos, and R. Ghosh. Leveraging multi-domain prior knowledge in topic models. In *IJCAI*, 2013.

[10] S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman. Indexing by latent semantic analysis. *JASIS*, 41(6):391–407, 1990.

[11] D. Downey and O. Etzioni. Look ma, no hands: Analyzing the monotonic feature abstraction for text classification. In *NIPS*, 2008.

[12] G. Druck, G. S. Mann, and A. McCallum. Learning from labeled features using generalized expectation criteria. In *SIGIR*, 2008.

[13] E. Gabrilovich and S. Markovitch. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *IJCAI*, 2007.

[14] A. Gliozzo, C. Strapparava, and I. Dagan. Improving text categorization bootstrapping via unsupervised learning. *ACM Trans. Speech Lang. Process.*, 6(1):1:1–1:24, Oct. 2009.

[15] Y. Grandvalet and Y. Bengio. Semi-supervised learning by entropy minimization. In *NIPS*, 2004.

[16] H. Guan, J. Zhou, and M. Guo. A class-feature-centroid classifier for text categorization. In *WWW*, 2009.

[17] S. Hingmire and S. Chakraborti. Topic labeled text classification: A weakly supervised approach. In *SIGIR*, 2014.

[18] S. Hingmire, S. Chougule, G. K. Palshikar, and S. Chakraborti. Document classification by topic labeling. In *SIGIR*, 2013.

[19] T. Hofmann. Probabilistic latent semantic indexing. In *SIGIR*, 1999.

[20] J. Jagarlamudi, H. D. III, and R. Udupa. Incorporating lexical priors into topic models. In *EACL*, 2012.

[21] M. J. Kusner, Y. Sun, N. I. Kolkin, and K. Q. Weinberger. From word embeddings to document distances. In *ICML*, 2015.

[22] B. Liu, X. Li, W. S. Lee, and P. S. Yu. Text classification by labeling words. In *AAAI*, 2004.

[23] C. Li, H. Wang, Z. Zhang, A. Sun, and Z. Ma. Topic Modeling for Short Texts with Auxiliary Word Embeddings. In *SIGIR*, 2016.

[24] C. D. Manning, P. Raghavan, and H. Schtze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[25] D. Mimno, H. M. Wallach, E. Talley, M. Leenders, and A. McCallum. Optimizing semantic coherence in topic models. In *EMNLP*, 2011.

[26] A. Mukherjee and B. Liu. Aspect extraction through semi-supervised modeling. In *ACL*, 2012.

[27] K. Nigam, A. K. MacCallum, S. Thrun, and T. Mitchell. Text classification from labeled and unlabeled documents using em. *Machine learning*, 39(2-3):103–134, 2000.

[28] H. Raghavan, O. Madani, and R. Jones. Active learning with feedback on features and instances. *Journal of Machine Learning Research*, 7:1655–1686, 2006.

[29] Y. Song and D. Roth. On dataless hierarchical text classification. In *AAAI*, 2014.

[30] T. Straatsma, H. Berendsen, and A. Stam. Estimation of statistical errors in molecular simulation calculations. *Molecular Physics*, 57(1):89–95, 1986.

[31] G. Thomas and S. Mark. Finding scientific topics. In *PNAS*, 2004.

[32] H. M. Wallach, D. M. Mimno, and A. McCallum. Rethinking LDA: why priors matter. In *NIPS*, 2009.

[33] P. Xie and E. P. Xing. Integrating document clustering and topic modeling. In *UAI*, 2013.