Journal of Machine Learning Research 22 (2021) 1-48 Submitted 5/18; Revised 8/20; Published 2/21

Bayesian Text Classification and Summarization via A Class-Specified Topic Model

Feifei Wang FEIFEI.WANG@RUC.EDU.CN
Center for Applied Statistics
School of Statistics
Renmin University of China
Haidian District, Beijing 100872, China

Junni L. Zhang ZJN@NSD.PKU.EDU.CN
National School of Development, Center for Statistical Science
and Center for Data Science
Peking University
Haidian District, Beijing 100871, China

Yichao Li LIYICHAO16@MAILS.TSINGHUA.EDU.CN
Center for Statistical Science
Tsinghua University
Haidian District, Beijing 100084, China

Ke Deng KDENG@TSINGHUA.EDU.CN
Center for Statistical Science
Tsinghua University
Haidian District, Beijing 100084, China

Jun S. Liu JLIU@STAT.HARVARD.EDU
Department of Statistics
Harvard University
Cambridge, MA 02138, USA

Editor: Qiaozhu Mei

Abstract

We propose the class-specified topic model (CSTM) to deal with the tasks of text classification and class-specific text summarization. The model assumes that in addition to a set of latent topics that are shared across classes, there is a set of class-specific latent topics for each class. Each document is a probabilistic mixture of the class-specific topics associated with its class and the shared topics. Each class-specific or shared topic has its own probability distribution over a given dictionary. We develop a Bayesian inference of CSTM in the semisupervised scenario, with the supervised scenario as a special case. We analyze in detail the 20 Newsgroups dataset, a benchmark dataset for text classification, and demonstrate that CSTM has better performance than a two-stage approach based on latent Dirichlet allocation (LDA), several existing supervised extensions of LDA, and an L^1 penalized logistic regression. The favorable performance of CSTM is also demonstrated through Monte Carlo simulations and an analysis of the Reuters dataset.

Keywords: Text Mining, Latent Topic, Semisupervised Classification, L^1 Penalization

1. Introduction

Text classification is an important first step in organizing large document collections and is widely applied (e.g. Chen and Dumais, 2000; Broder et al., 2007; Gomez and Moens, 2012). Automatic text summarization allows users to quickly grasp and compare themes in given text corpora and has become increasingly important with the expanding accumulation of text documents in all fields (e.g. Zubiaga et al., 2011; Jia et al., 2014). In practice, it is often labor intensive and time consuming to obtain labeled documents, but it is easy to obtain a large number of unlabeled documents. By combining information from both unlabeled and labeled documents, semisupervised approaches have the potential to obtain a higher classification accuracy and better class-specific text summaries than supervised approaches that use labeled documents only.

A popular paradigm of representing documents in text analysis is vector space models (Salton, 1989; Belew, 2000). In a vector space model, each document is represented by a vector whose length is equal to the size of a dictionary. Each element in the vector is a certain normalized version of the count of the number of times that the corresponding term (word or phrase) in the dictionary appears in the document. In text classification problems, such document-term representation often leads to a high-dimensional feature set, which poses a challenge for many classification methods. Therefore, both supervised and semisupervised approaches have been proposed to perform dimension reduction (e.g., Yang and Pedersen, 1997; Baker and McCallum, 1998; Cong et al., 2004; Su et al., 2011; Gomez and Moens, 2012). For the task of class-specific text summarization, a recent notable supervised approach is the concise comparative summarization (CCS) method proposed by Jia et al. (2014). For a given class, CCS uses supervised sparse classification methods, Lasso (Tibshirani, 1996), and L^1-penalized logistic regression (e.g., Genkin et al., 2007; Ifrim et al., 2008) to select phrases that can predict belonging to a class. The resulting small set of selected phrases can serve as a summary of texts in the class.

Another paradigm of representing text documents and achieving dimension reduction simultaneously is a suite of hierarchical models known as topic models (e.g., Blei et al., 2003; Griffiths and Steyvers, 2004). Compared to vector space models, topic models can better reveal the thematic structure in the document collection. The most basic topic model is the latent Dirichlet allocation (LDA, Blei et al., 2003) model, in which there are a set of latent topics underlying all documents, with each topic being represented by its specific vector of probabilities over the dictionary. Each document is assumed to be generated by a probability distribution over these topics. The vector of topic probabilities for each document and the vector of word probabilities for each topic follow Dirichlet distributions.

LDA is an unsupervised model. A common practice of using LDA for text classification is to take a two-stage approach: first, estimating an LDA model using all training documents without reference to their class labels; and second, using the document-specific topic probabilities to train a classifier, such as a support vector machine (SVM). This two-stage approach often has better classification accuracy than classifiers directly built on the high-dimensional term features (Blei et al., 2003). A two-stage approach can also be used to obtain class-specific text summarization. For each class, some topics extracted by LDA can be selected to represent the class, and words with top probabilities under these topics can be used to summarize texts within the class. Another approach is to apply LDA separately to documents within each class (Blei et al., 2003, Section 4.2). For each document, a class label can be assigned based on comparing its likelihood under different classes. Topics extracted under each class can be used to obtain class-specific text summarization. We refer to this approach as LDA per class.

Numerous extensions of LDA have also been proposed to account for discrimination between classes in the model. In the stream of supervised extensions, Blei and McAuliffe (2007) proposed the supervised latent Dirichlet allocation model (sLDA), in which documents are generated in the same way as LDA, each document is paired with a response that is related to empirical frequencies of the topics in the document through a generalized linear model, and the parameters in LDA and in the generalized linear model are estimated jointly. Lacoste-Julien et al. (2008) propose the discriminatively trained LDA model (DiscLDA), in which each document is associated with a class label and the vector of topic probabilities for each document is obtained by applying a class-specific linear transformation to a Dirichlet random variable. Ramage et al. (2009) proposed labeled LDA, in which each document is associated with a set of labels, each label directly corresponds to a topic, and each document is constrained to use only those topics that correspond to its label set. Ramage et al. (2009) also mentioned an extension of the labeled LDA model that allows a common background topic in all documents. Resnik et al. (2015) proposed a supervised nested LDA model (SNLDA) in which, similar to sLDA, the response for a document is related to empirical frequencies of the topics in the document but the underlying topics are organized into a tree. Examples of other supervised extensions include Zhu et al. (2009), Lin et al. (2012), Cao et al. (2015), and Li et al. (2015).

There has been relatively less work on semisupervised extensions of LDA. Wang et al. (2012) and Lu et al. (2013) proposed semisupervised LDA (ssLDA) methods. In these methods, each class is associated with a topic. Hybrid generative models are assumed for labeled and unlabeled documents. A labeled document is restricted to use only the topic associated with its class, but an unlabeled document can use all topics. Zhang and Wei (2014) propose semisupervised topic models, which also use hybrid generative models. For a labeled document, the generative model describes how the words and the label are generated. For an unlabeled document, the generative model describes only how the words are generated.

In this paper, we propose the class-specified topic model (CSTM) as another extension of LDA. Our method is motivated through the observation that, in the aforementioned two-stage approach, the training of LDA in the first stage does not take into account any information on the known class labels in the training documents; therefore, many extracted topics are likely to be unhelpful for classification or class-specific summarization in the second stage. To alleviate this problem, we partition the latent topics into class-specific ones associated with each class and common ones shared across classes. The class-specific topics are designed to capture contents that are distinctively discussed in each class, and the shared topics are intended for contents that are common to all classes. We assume coherent generative models for labeled and unlabeled documents. Each document, regardless of whether it is labeled or unlabeled, can be represented by a probabilistic mixture of the class-specific topics associated with its class and the shared topics. Each class-specific or shared topic has its vector of probabilities over the dictionary.

CSTM imposes a sparsity constraint that each document has zero probability to choose class-specific topics not associated with its class. Therefore, it is more parsimonious than LDA and potentially has better generalizability. We will show that CSTM can represent unseen text documents better than LDA, which in turn can facilitate classification of these documents. Furthermore, because the class-specific topics in CSTM are directly related to class discrimination, we expect that CSTM can achieve higher classification accuracy and better class-specific text summaries than the two-stage approach based on LDA.

The rest of this paper is organized as follows. Section 2 introduces our main dataset consisting of articles from 20 Newsgroups, a benchmark dataset for text classification, applies LDA to this dataset, and discusses our motivations for introducing CSTM. Section 3 presents CSTM and compares analytical structures of the supervised CSTM with sLDA, DiscLDA and labeled LDA. Section 4 develops Bayesian inference of CSTM in the semisupervised scenario, with the supervised scenario as a special case. Section 5 applies CSTM to the Newsgroups dataset and makes a detailed comparison with the two-stage LDA approach, the LDA per class approach, sLDA, DiscLDA, labeled LDA, and a modified CCS approach. Section 5 also mentions further comparison between the supervised CSTM and its competitors through Monte Carlo simulations and an analysis of the Reuters dataset (details in Appendix). Section 6 concludes with a brief discussion.

2. The Dataset and Motivations

In this section, we first introduce the 20 Newsgroups dataset and then present its application by using LDA, the results of which motivated us to develop CSTM.

2.1 The 20 Newsgroups dataset

The 20 Newsgroups dataset[1] is a benchmark dataset for text classification. It contains 18,846 newsgroup documents, and on average there are 148.51 words in each document. These documents are partitioned (nearly) evenly across 20 different newsgroups, which are treated as 20 classes. Some of the newsgroups are semantically closely related to each other while others are highly unrelated; hence the 20 newsgroups can be further partitioned into six class groups. The dataset has also been partitioned into a training set with 60% of the documents and a test set with 40% of the documents. Table 1 reports the class groups, the classes, and the number of training and test documents in each class.

Following common practices in text mining, we first preprocess the documents by using the NLTK library in Python to remove numbers, punctuations and stopwords (words that are commonly used but have little semantic meaning in most occasions, such as "the" and "is"). After preprocessing, there are 117,918 unique words in the training documents, which constitute the dictionary.

We consider the semisupervised scenario in which a fraction φ of training documents are randomly chosen to be labeled and the remaining (1-φ) fraction of the training documents are unlabeled. We are interested in using all training documents, labeled and unlabeled, to build classifiers and extract class-specific text summaries.

2.2 Application of LDA to the 20 Newsgroups dataset

We apply LDA to all training documents. LDA assumes that there are K latent topics underlying all documents. Each document d is a mixture of these topics according to probability vector $\boldsymbol{\theta}_d = (\theta_{d,1}, ..., \theta_{d,K})^\top$. Each topic k is characterized by its probability distribution $\boldsymbol{\phi}_k = (\phi_{k,1}, ..., \phi_{k,V})^\top$ over the dictionary with size V. The generative process of D documents is illustrated in Figure 1 and described below.

1. Generate $\boldsymbol{\phi}_k$ $(k = 1, ..., K)$ independently from a Dirichlet distribution with parameter $\boldsymbol{\beta} = (\beta_1, \cdots, \beta_V)^\top$: $\boldsymbol{\phi}_k \sim Dir(\boldsymbol{\beta})$.

[1] This dataset is downloadable from http://qwone.com/~jason/20Newsgroups/.

Table 1: Class Groups, Classes and the Numbers of Documents in Each Class

Class Group	Class Number	Class Name	No. Training	No. Test
Computer Science	1	comp.graphics	584	389
	2	comp.os.ms-windows.misc	591	394
	3	comp.sys.ibm.pc.hardware	590	392
	4	comp.sys.mac.hardware	578	385
	5	comp.windows.x	593	395
For Sale	6	misc.forsale	585	390
Auto & Sports	7	rec.autos	594	396
	8	rec.motorcycles	598	398
	9	rec.sport.baseball	597	397
	10	rec.sport.hockey	600	399
Science	11	sci.crypt	595	396
	12	sci.electronics	591	393
	13	sci.med	594	396
	14	sci.space	593	394
Politics	15	talk.politics.guns	546	364
	16	talk.politics.mideast	564	376
	17	talk.politics.misc	465	310
Religion	18	alt.atheism	480	319
	19	soc.religion.christian	599	398
	20	talk.religion.misc	377	251

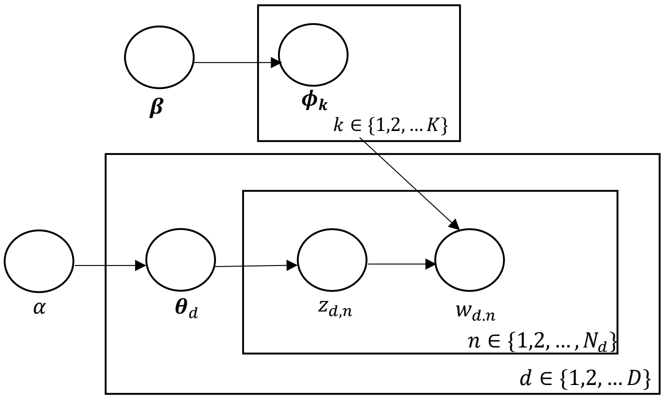

Figure 1: Generative Process for the LDA Model

2. Each document d ($d = 1, ..., D$) is independently generated as follows:

   (a) Generate $\boldsymbol{\theta}_d$ from a Dirichlet distribution with parameter $\boldsymbol{\alpha} = (\alpha_1, \cdots, \alpha_K)^\top$: $\boldsymbol{\theta}_d \sim Dir(\boldsymbol{\alpha})$.

   (b) Let N_d be the number of words in document d. The nth word in document d ($n = 1, \cdots, N_d$) is generated independently as follows:

      i. Choose a topic $z_{d,n}$ by drawing $z_{d,n} \sim Multi(\boldsymbol{\theta}_d)$;

      ii. Choose a word $w_{d,n}$ by drawing $w_{d,n} \sim Multi(\boldsymbol{\phi}_{z_{d,n}})$.

Here, $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are given hyperparameters. We consider a homogeneous Dirichlet distribution with $\alpha_1 = \cdots = \alpha_K = \alpha$ for the topic probabilities, which does not prefer any topic *a priori*.

Let $\boldsymbol{z}_d = (z_{d,1}, ..., z_{d,N_d})^\top$ and $\boldsymbol{w}_d = (w_{d,1}, ..., w_{d,N_d})^\top$ denote the vectors of topic indicators and words, respectively, for document d, and let $\boldsymbol{z} = \{\boldsymbol{z}_1, ..., \boldsymbol{z}_D\}$ and $\boldsymbol{w} = \{\boldsymbol{w}_1, ..., \boldsymbol{w}_D\}$ denote the collection of these indicator and word vectors for all training documents. Let $\boldsymbol{\Theta} = \{\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_D\}$ and $\boldsymbol{\Phi} = \{\boldsymbol{\phi}_1, ..., \boldsymbol{\phi}_K\}$. With $\boldsymbol{z}$ being treated as missing data, the full posterior distribution of $(\boldsymbol{\Theta}, \boldsymbol{\Phi}, \boldsymbol{z})$ can be derived from the generative process of the postulated topics model and is presented in Appendix A.

Since the Dirichlet priors are conjugate to the multinomial distributions, the marginal posterior distribution $p(\boldsymbol{z}|\boldsymbol{w})$ can be easily obtained by integrating out $\boldsymbol{\Theta}$ and $\boldsymbol{\Phi}$ from the full posterior distribution. Therefore, the LDA model can be trained using the collapsed Gibbs sampling algorithm (Liu, 1994; Griffiths and Steyvers, 2004), which samples $\boldsymbol{z}$ by iteratively sampling $z_{d,n}$ given the other elements of $\boldsymbol{z}$. After convergence is reached, given each draw of $\boldsymbol{z}$, $\boldsymbol{\Theta}$ and $\boldsymbol{\Phi}$ can be easily estimated. Let $\tilde{\theta}_{d,k}$ and $\tilde{\phi}_{k,v}$ denote the posterior means of estimates of $\theta_{d,k}$ and $\phi_{k,v}$. The meaning of topic k can be characterized by words with high values of $\tilde{\phi}_{k,v}$.

Following Griffiths and Steyvers (2004), we set $\alpha = 50/K$ and $\beta_v = 0.1$ for $v = 1, \cdots, V$. We also follow the Bayesian approach in Griffiths and Steyvers (2004) to train the LDA model, select the number of topics, and obtain topic probabilities for training or test documents. The marginal log-likelihood of $\boldsymbol{w}$ is used to select the number of topics from 1 to 201, with an increment of 5. The number of topics corresponding to the largest marginal log-likelihood is selected, resulting in 71 topics.

2.3 Motivation for developing CSTM

We set the fraction of labeled training documents as φ=20% in this section. We first show that many topics extracted by LDA do not have the ability to distinguish between classes.

For each topic extracted by LDA, we define its strength of association with each of the 20 classes as follows. Let M_j^{obs} denote the number of labeled training documents belonging to class j and Ω_j^{obs} the set of indexes for these documents. Recall that $\tilde{\theta}_{d,k}$ is the posterior mean of estimates of $\theta_{d,k}$ under the LDA model. For topic k and class j, we define

$$\rho_{k,j} = \frac{1}{M_j^{\text{obs}}} \sum_{d \in \Omega_j^{\text{obs}}} \tilde{\theta}_{d,k}, \tag{1}$$

which is the average topic probability on topic k for labeled training documents belonging to class j. We further normalize $\rho_{k,j}$ to

$$\delta_{k,j} = \frac{\rho_{k,j}}{\sum_{j'=1}^{J} \rho_{k,j'}}. \tag{2}$$

Clearly, if $\delta_{k,j}$ is concentrated on a very few classes, then topic k is useful for class separation; however, if $\delta_{k,j}$ is nearly uniform, then topic k is not helpful for class discrimination.

Figure 2 shows the heatmap of $\delta_{k,j}$, in which each row represents a topic and each column represents a class. The classes are ordered as in Table 1, and the 71 topics are also ordered for a clear visualization. Briefly, we first screen out those topics with $\max_j \delta_{k,j} < 0.1$ (listed at the bottom). Then, starting with class 1, for each class j we sequentially find topics that are most related to it. If a topic is picked up by two classes, we assign it to the one with the higher $\delta_{k,j}$.

We see from the figure that some topics are highly associated with specific classes. For instance, topic 20 is almost exclusively associated with class 10 (“rec.sport.hockey”). The top ten words with the highest values of $\tilde{\phi}_{k,v}$ under this topic are “team, game, hockey, play, NHL, games, season, players, go, period”, which are closely related to hockey games discussed in class 10. As another example, topic 47 is highly associated with classes 19 (“soc.religion.christian”) and 20 (“talk.religion.misc”) and hence may represent a mixture of contents for these two classes. The top words under topic 47 are “God, Jesus, Bible, church, Christian, Christ, Christians, faith, one, Gods”, which are also closely related to religion issues discussed in these two classes. By contrast, the topics at the bottom of Figure 2 are not strongly associated with any class. For example, the top ten words in topic 71 in Figure 2 are “one, would, people, may, many, us, even, also, must, question”, all of which are nonspecific. Although for each class there are certain topics that are more strongly associated with it than others, the overall picture given by Figure 2 is not clear. There is a lot of noise in discriminating between classes using the topics extracted by LDA.

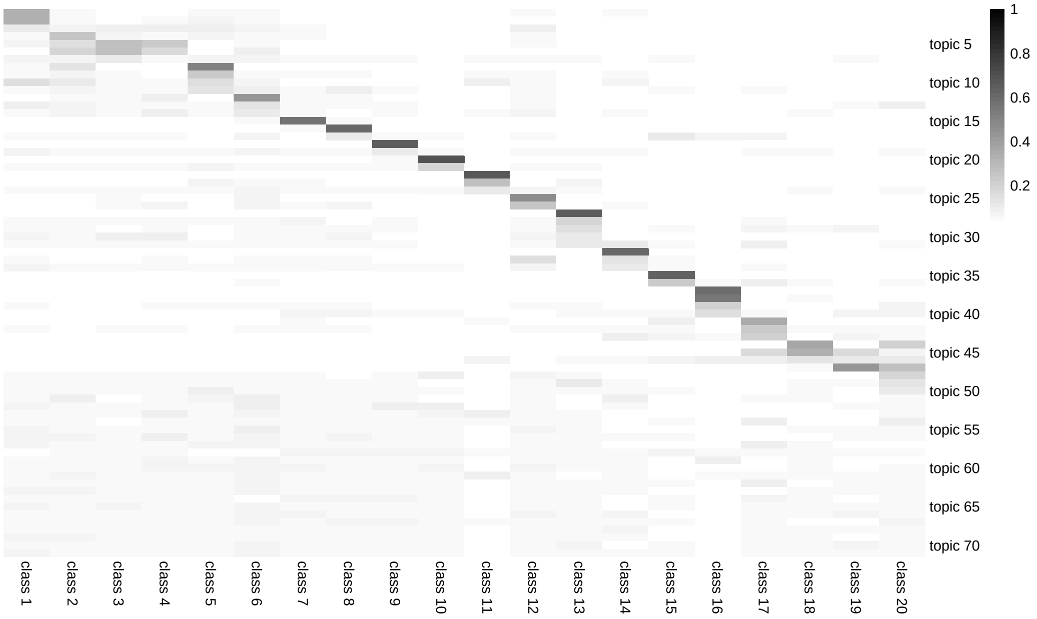

Figure 2: The heatmap of $\delta_{k,j}$ across 71 topics extracted by LDA and 20 classes

3. The Class-Specified Topic Model

This section introduces CSTM in detail and then compares the supervised CSTM with other supervised extensions of LDA.

3.1 Generative Process of the CSTM

CSTM partitions latent topics into h_j class-specific ones associated with each class j ($j = 1, ..., J$), which capture contents that are distinctively discussed in each class, and h_S “shared” topics commonly discussed by a large number of classes.

Let $\boldsymbol{H}_j$ be the set of indexes for topics specific to class j ($j = 1, ..., J$), $\boldsymbol{H}_C = \cup_{j=1}^{J} \boldsymbol{H}_j$ the set of all class-specific topics, and $\boldsymbol{H}_S$ the set of shared topics. Each document in class j is a probabilistic mixture of topics in the subset $\boldsymbol{\Lambda}_j = \boldsymbol{H}_j \cup \boldsymbol{H}_S$. The total number of topics is $K = \sum_{j=1}^{J} h_j + h_S$. Figure 3 illustrates our model with $J = 3$, $h_1 = 3$, $h_2 = 2$, $h_3 = 2$ and $h_S = 2$.

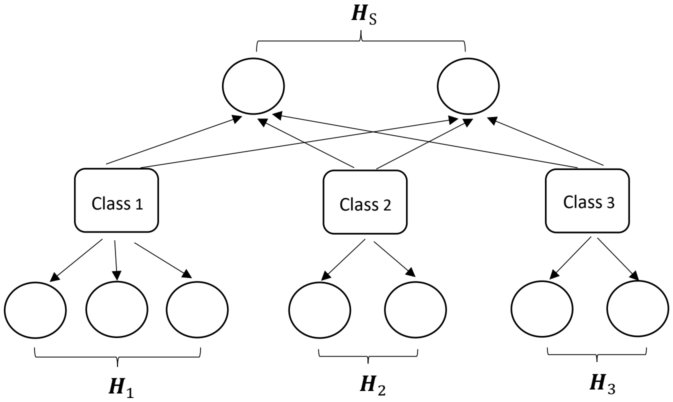

Figure 3: Illustration of CSTM

Let $y_d \in \{1, 2, ..., J\}$ denote the class label for document d. Given $y_d = j$, the topic probabilities in $\boldsymbol{\theta}_d$ are nonzero only for topics $k \in \boldsymbol{\Lambda}_j \equiv \boldsymbol{H}_j \cup \boldsymbol{H}_S$. Each class-specific or shared topic has a vector of probabilities $\boldsymbol{\phi}_k$ ($k = 1, ..., K$) over the dictionary. Let $\boldsymbol{\eta} = (\eta_1, ..., \eta_J)^\top$ denote the probabilities that documents belong to the J classes. The generative process of D documents is illustrated in Figure 4 and described below.

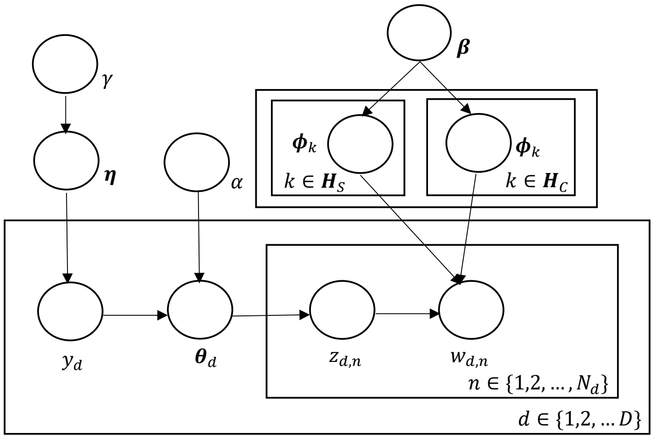

Figure 4: Generative Process for CSTM

1. Generate the class distribution (J dimensional) as $\boldsymbol{\eta} \sim Dir(\boldsymbol{\gamma})$;

2. Generate the probability distribution for each topic (V-dimensional) independently: $\boldsymbol{\phi}_k \sim Dir(\boldsymbol{\beta})$ for $k = 1, ..., K$;

3. Each document d is independently generated as follows:

(a) Generate class label $y_d \sim Multi(\boldsymbol{\eta})$;

(b) Generate its probability vector over all topics, $\boldsymbol{\theta}_d$ by first setting those $\theta_{d,l}$ corresponding to topics not in $\boldsymbol{\Lambda}_{y_d}$ to zero and then generating the remaining $h_{y_d} + h_S$ components from $Dir(\alpha, \cdots, \alpha)$;

(c) The nth word in document d ($n = 1, \cdots, N_d$) is independently generated as follows:

   i. Choose a topic by drawing $z_{d,n} \sim Multi(\boldsymbol{\theta}_d)$,

   ii. Choose a word $w_{d,n} \sim Multi(\boldsymbol{\phi}_{z_{d,n}})$.

Here, α, $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$ are given hyperparameters. We consider a homogeneous Dirichlet distribution for the class probabilities that does not prefer any class a priori, such that $\gamma_1 = \cdots = \gamma_J = \gamma$.

CSTM can be regarded as an extension of LDA, with the constraint that $\theta_{d,k} = 0$ if $k \notin \boldsymbol{\Lambda}_{y_d}$. The benefits of this constraint are twofold. First, when the number of topics is the same, CSTM is more parsimonious than LDA. Hence, CSTM potentially has better generalization ability than LDA. Second, the class-specific topics in CSTM directly extract information related to class discrimination, which can help achieve higher classification accuracy and better class-specific text summaries.

Under the semisupervised scenario, let $\boldsymbol{y}^{mis}$ denote the set of unknown class labels for unlabeled training documents. The full posterior distribution of $(\boldsymbol{\eta}, \boldsymbol{y}^{mis}, \boldsymbol{\Theta}, \boldsymbol{\Phi}, \boldsymbol{z})$ can be derived from the generative process and is presented in Appendix B. The supervised scenario is a special case with $\varphi = 100\%$, where $\boldsymbol{y}^{mis}$ is omitted.

3.2 Analytical Comparison of the Supervised CSTM and Other Supervised Extensions of LDA

Supervised topic modeling has been an active research direction since late 2000. Many effective supervised extensions of LDA have been developed in the past decade, including LDA per class (Blei et al., 2002), sLDA (Blei and McAuliffe, 2007), DiscLDA (Lacoste-Julien et al., 2008), and labeled LDA (Ramage et al., 2009). Here, we discuss the structural differences between the supervised CSTM (with $\varphi = 100\%$) and these earlier methods.

In the LDA per class approach, an LDA model is first estimated within each class. The log marginal likelihood of each test document is calculated under the estimated LDA model for each class. Then, the label of each test document is predicted to be the class with the largest log marginal likelihood. Under this approach, all of the topics can be regarded as class-specific topics. However, common contents appearing across classes are represented by different sets of topics under each class, which introduces extra model complexity and is not helpful for class discrimination. Compared to LDA per class, CSTM uses shared topics to account for common contents appearing across classes, and the extracted class-specific topics allow for a stronger discriminating power.

In sLDA, the generative process for the documents is the same as LDA. In the classification context, each document d is associated with a class label y_d, which is related to the empirical topic probabilities,

$$\left(\frac{1}{N_d} \sum_{n=1}^{N_d} I(z_{d,n} = 1), \cdots, \frac{1}{N_d} \sum_{n=1}^{N_d} I(z_{d,n} = K) \right),$$

through a generalized linear model. The parameters in LDA and those in the generalized linear model are jointly estimated. Compared with sLDA, CSTM directly incorporates class labels into the generative process for documents and is more parsimonious and coherent.

In DiscLDA, for each document d with class label y_d, the K-dimensional vector of topic probabilities $\boldsymbol{\theta}_d$ is obtained by applying a class-specific linear transformation $\boldsymbol{T}^{y_d}$ to an L-dimensional Dirichlet variable $\boldsymbol{\xi}_d$, that is, $\boldsymbol{\theta}_d = \boldsymbol{T}^{y_d}\boldsymbol{\xi}_d$. The transformation matrices $\boldsymbol{T}^y$ can be jointly estimated with the vectors of word probabilities $\boldsymbol{\phi}_k$. When the transformation matrices are not estimated but rather fixed in the following form with $(J+1)$ rows and two columns of block matrices,

$$\boldsymbol{T}^1 = \begin{pmatrix} \boldsymbol{I}_{K_0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \\ \vdots & \vdots \\ \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{I}_{K_1} \end{pmatrix}, \quad \boldsymbol{T}^2 = \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ \boldsymbol{I}_{K_0} & \mathbf{0} \\ \vdots & \vdots \\ \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{I}_{K_1} \end{pmatrix}, \quad \boldsymbol{T}^J = \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \\ \vdots & \vdots \\ \boldsymbol{I}_{K_0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{I}_{K_1} \end{pmatrix},$$

DiscLDA corresponds to a special case of CSTM with fixed values of h_j's and h_S: $h_j = K_0$ for all j, and $h_S = K_1$. Although there seems to be a close connection between DiscLDA and CSTM, there are a number of differences between these two models. First, DiscLDA allows more flexibility through general transformation matrices, whereas CSTM allows more flexibility through different numbers of class-specific topics for different classes. Second, with general transformation matrices, DiscLDA may suffer from high model complexity; with the sparsity constraint on topic probabilities, CSTM may be more parsimonious. Third, the estimation and prediction methods are different for DiscLDA and CSTM (see Section 5.2 and Section 4 for more details).

In labeled LDA, each document can be associated with a set of labels, and the number of topics is set to be the number of unique labels in the documents. In addition to these label-specific topics, a common background topic can be added (Ramage et al., 2009). For each document d, its vector of topic probabilities, $\boldsymbol{\theta}_d$, is restricted to be defined only over the topics that correspond to its label set and the common topic. In the classification context where each document is associated with only one class label, the number of topics equals the number of classes plus one, and each document d can only use the topic that corresponds to its class and the common topic. CSTM allows each document to use multiple class-specific topics and/or multiple shared topics, which is more flexible than the labeled LDA model.

4. Bayesian Inference of the CSTM

This section discusses model inference, model selection, and prediction of class labels in CSTM.

4.1 Inference of model parameters

Under the semisupervised scenario, with $\boldsymbol{y}^{mis}$ unknown, the collapsed Gibbs sampling algorithm cannot be applied directly to make inferences about CSTM because the corresponding collapsed target distribution is too complicated to sample from. We instead design a Gibbs sampling algorithm with an embedded Metropolis-Hastings step. Details are presented in Appendix C.

We note that CSTM has a nonidentifiability issue: the posterior distribution remains invariant after any permutation of the topics indexes within any $\boldsymbol{H}_j$ $(j = 1, \cdots, J)$ or within $\boldsymbol{H}_S$. Solving this problem can reduce variation in posterior samples and make the resulting topics more meaningful. We propose a solution to the nonidentifiability problem (see Appendix D for details). After that, R posterior samples of $(\boldsymbol{\eta}, \boldsymbol{y}^{mis}, \boldsymbol{\Theta}, \boldsymbol{\Phi}, \boldsymbol{z})$ are obtained.

4.2 Model selection

In CSTM, the numbers of class-specific topics, h_j $(j = 1, \cdots, J)$, and the number of shared topics, h_S, need to be selected. In practice, it is computationally expensive or even impossible to perform model selection among all possible combinations of h_j and h_S. We perform model selection through two steps.

In the first step, we apply LDA to all training documents and extract K^{LDA} topics. We calculate $\delta_{k,j}$ as a measure of the strength with which topic k is associated with class j as in Section 2.3. The distribution of $\delta_{k,j}$ is then examined to give candidate sets of values for h_j's and h_S.

With a given lower limit τ, we regard $\delta_{k,j} > \tau$ as indicating that an additional class-specific topic is needed to characterize class j. Since the average value of $\delta_{k,j}$ across J classes is $1/J$, we require $\tau \geq 1/J$. Because the shared topics do not directly discriminate between classes, we set an upper limit, $h_{S,\max}$, for the number of shared topics.

Given the values of τ and $h_{S,\max}$, the candidate set of values for h_j and h_S is derived as follows.

1. Initialize $h_j = 0$ $(j = 1, \cdots, J)$ and $h_S = 0$.
2. For each topic k $(k = 1, \cdots, K^{\text{LDA}})$:
   (a) For any class j^* that satisfies $\delta_{k,j^*} > \tau$, topic k indicates that an additional class-specific topic is needed for this class. Therefore, we set $h_{j^*} \leftarrow h_{j^*} + 1$.
   (b) If for all classes $\delta_{k,j} \leq \tau$, then topic k corresponds to a shared topic. Therefore, we set $h_S \leftarrow h_S + 1$.
3. Set $h_j \leftarrow \max(h_j, 1)$ for $j = 1, \cdots, J$ such that each class has at least one class-specific topic. Set $h_S \leftarrow \min(h_S, h_{S,\max})$ such that the number of shared topics is at most $h_{S,\max}$.

We let τ vary from $\max(1/J, 0.3)$ to 0.8, with an increment of 0.05, and let $h_{S,\max}$ take values among $\{1, 2, 3, 4\}$ and then 5 to $\min(K^{\text{LDA}} - \sum_{j=1}^{J} h_j, 20)$, with an increment of 5. Constraining τ to be at least 0.3 can save computational cost and does not yield different results according to our experience. The upper bound constraint on $h_{S,\max}$ indicates that the total number of topics in CSTM does not exceed K^{LDA} and that the number of shared topics does not exceed 20. Practically, 20 topics are rich enough to characterize contents shared across different classes. Constraining the number of shared topics to be at most 20 can save computational cost when K^{LDA} is large and does not yield different results according to our experience. For each set of values for τ and $h_{S,\max}$, we can obtain a candidate set of values for h_j $(j = 1, \cdots, J)$ and h_S.

In the second step, a fivefold cross-validation is used to choose the candidate set of values for h_j and h_S that yields the highest predictive accuracy.[2] Specifically, the labeled training documents are split into five equal-sized subsamples. Each of the subsamples is treated in turn as the validation set, and the remaining four subsamples combined with the unlabeled training documents are used to train each candidate model. For each validation set and each candidate model, we run three parallel chains and select the chain with the highest average log-posterior density. Posterior samples from the selected chain are applied to predict the class labels for documents in the validation set (details in Appendix E). We then calculate the average correct classification rate over the five validation sets for

[2] We also tried Bayesian model selection approaches using the Watanabe-Akaike information criterion (WAIC, Watanabe, 2010) or the marginal log-likelihood estimated through the harmonic mean identity (Raftery et al., 2007), but lower classification accuracy is achieved.

each candidate model and select the candidate model with the highest average correct classification rate.

4.3 Prediction of class labels

For an unlabeled training document d, we use the posterior samples of $\boldsymbol{y}^{\text{mis}}$ to predict its class label as the class with the largest posterior proportion. For a validation or test document whose class label is unknown, we use the posterior samples of $(\boldsymbol{\eta}, \boldsymbol{\Phi})$ and apply a Bayes rule to predict its class label. Details are given in Appendix E.

5. Analysis of the 20 Newsgroups Dataset

This section presents the analysis of the 20 Newsgroups dataset using CSTM and its competitors. To demonstrate the ability of CSTM to utilize information in unlabeled documents, we let the fraction φ of labeled training documents vary from 10% to 100%, with an increment of 10%.

5.1 Training CSTM

We first obtain a few candidate sets of values of h_j's and h_S as in Section 4.2. Each candidate model is then trained using all training documents with the methods presented in Section 4.1. For the hyperparameters, we set $\alpha = 0.5$, and $\beta_v = 0.1$ for $v = 1, \cdots, V$, which are commonly used in LDA applications (Koltcov et al., 2014; Chen et al., 2016; Qiang et al., 2017). Alternative specifications of α and $\boldsymbol{\beta}$ are considered in Section 5.7, and the results do not show significant differences. We also set $\gamma = 1$ so that the prior on the class probabilities is uniform. The tuning parameter for the Metropolis-Hastings step (see Appendix C) is set at $\xi = 0.1$, which gives an acceptance rate of approximately 10%.

Each chain is first run for $B = 200$ burn-in iterations without addressing the identifiability issue. A set of $G = 15$ samples of $\boldsymbol{\Phi}$ is then obtained by taking every 20th draw from the next 300 draws and is used to solve the identifiability issue as in Appendix D. The chain is then run for another 5000 iterations, with the first 1000 iterations discarded as burn-in iterations and the last 4000 iterations used for model inference. To check model convergence, we run three chains from different random starting points under the same setting and check the corresponding trace plots of parameters as well as the logarithm of posterior density. Figure 5 shows the trace plot of η_1 and the log-posterior density when $\varphi = 20\%$, $h_j = 1$ for all j and $h_S = 5$. We observe that different chains may converge to different modes. The chains with lower log-posterior density can also be associated with lower classification accuracy. Therefore, we do not mix draws from the three chains and instead obtain $R = 10$ samples of $(\boldsymbol{\eta}, \boldsymbol{y}^{mis}, \boldsymbol{\Phi})$ by taking every 400th draw from the last 4000 draws in the chain with the highest average log-posterior density. The class labels for validation documents, unlabeled training documents and test documents are then predicted as in Section 4.3.

We note that, for the supervised version of the CSTM, the Metropolis-Hastings step is not needed (see Appendix C for the detail). Running different chains under the same setting gives similar results; hence we can just run one chain under each setting. The computational time can be reduced considerably for the supervised CSTM. See Section 5.6.2 for further discussion.

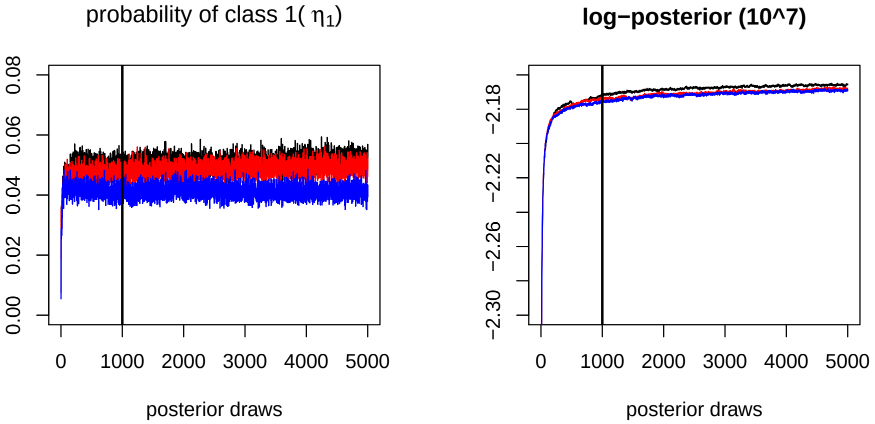

Figure 5: The trace plots of η_1 and log-posterior probability in three different chains when $\varphi = 20\%$, $h_j = 1$ for all j and $h_S = 5$

5.2 Training models for comparison

For comparison, we consider the two-stage approach based on LDA. First, using all training documents without reference to their class labels, we train a set of LDA models whose numbers of topics range from 1 to 201 with an increment of 5 and select the optimal number of topics based on the log marginal likelihood. Second, the document-specific topic probabilities under the selected LDA model and the class labels for the fraction φ of labeled training documents are used to train a random forest (RF) or a support vector machine (SVM). The resulting RF or SVM is then applied to the document-specific topic probabilities for the unlabeled training or test documents to predict the class labels of these documents. We also consider an LDA per class approach. Using all training documents within each class, we train a set of LDA models whose numbers of topics range from 2 to 20 with an increment of 1 and select the optimal number of topics based on the log marginal likelihood. For each test document, the log marginal likelihood under the selected LDA model for each class is calculated. The label of the test document is predicted to be the class with the largest log marginal likelihood.

We modify the CCS model in Jia et al. (2014) to obtain another statistical model for comparison. Similar to CSTM and the two-stage approach, we consider using words as terms in the dictionary. We also consider using phrases as terms following Jia et al. (2014), where the phrases include all words, bigrams (phrases consisting of two consecutive words) and trigrams (phrases consisting of three consecutive words) that appear in the training documents. Following Jia et al. (2014), we remove words or phrases appearing fewer than six times to obtain a more concise dictionary, resulting in 23,419 unique words or 69,772 unique phrases. Details of the modified CCS approach are given in Appendix F.

We also consider several supervised extensions of LDA: sLDA, DiscLDA and labeled LDA. For sLDA, we use fivefold cross-validation to select the number of topics from 20 to 200, with an increment of 10. We then re-estimate the model using the optimal number of topics and all training documents and apply the model to predict class labels for test documents. For DiscLDA, Lacoste-

Julien et al. (2008) conducted an experiment on the 20 Newsgroups dataset with a set of pre-defined transformation matrices ($\boldsymbol{T}^j$'s). They used SVM to classify documents based on document-specific topic probabilities. The reported classification accuracy of test documents was 80%. We instead estimate the transformation matrices, use fivefold cross-validation to select the number of topics from 20 to 200 with an increment of 10, and re-estimate the model using the optimal number of topics and all training documents. Following Lacoste-Julien et al. (2008), we also use SVM to classify documents based on document-specific topic probabilities. The resulting classification accuracy of test documents is increased to 81.38%. For labeled LDA, there is one class-specific topic for each class, plus a common topic, resulting in a total of 21 topics.

5.3 Comparison between CSTM and LDA

For each topic k extracted by CSTM, we follow the procedure proposed in Section 2.3 to calculate $\delta_{k,j}$, which measures the strength of the association of topic k with class j. For any class-specific topic k associated with class j_k, since its topic probability equals zero for any document not in class j_k, we always have $\delta_{k,j_k} = 1$ and $\delta_{k,j'} = 0$ when $j' \neq j_k$. Hence, CSTM gives a clearer picture of the association between topics and classes than LDA (see Figure 2).

As discussed in Section 3.1, CSTM potentially has better generalization ability than LDA, and topics extracted by CSTM may better represent unseen documents. To illustrate this point, we show in Table 2 the log marginal likelihood of $\boldsymbol{w}^{\text{test}}$ under LDA and CSTM for varying values of φ (fraction of labeled training documents), where $\boldsymbol{w}^{\text{test}}$ denotes the collection of words for all test documents. Table 2 also lists the numbers of topics under LDA and CSTM. CSTM is more parsimonious than LDA in two aspects. First, the total number of topics inferred by CSTM is 25 or 30, much smaller than that by LDA (71). Second, CSTM has the sparsity constraint on topic probabilities as discussed in Section 3.1. However, the log marginal likelihood for CSTM is much higher than that for LDA, indicating that the test documents are better represented under CSTM.

Table 2: Log Marginal Likelihood of Test Documents under LDA and CSTM

LDA		
number of topics 71		log marginal likelihood -2.103×10^7
CSTM		
φ	h_j and h_S	log marginal likelihood
10%	all $h_j = 1, h_S = 5$	-6.803×10^6
20%	all $h_j = 1, h_S = 5$	-6.817×10^6
30%	all $h_j = 1, h_S = 5$	-6.834×10^6
40%	all $h_j = 1, h_S = 5$	-6.852×10^6
50%	all $h_j = 1, h_S = 5$	-6.864×10^6
60%	all $h_j = 1, h_S = 5$	-6.858×10^6
70%	all $h_j = 1, h_S = 5$	-6.872×10^6
80%	all $h_j = 1, h_S = 10$	-6.814×10^6
90%	all $h_j = 1, h_S = 10$	-6.809×10^6
100%	all $h_j = 1, h_S = 10$	-6.816×10^6

5.4 Classification performances of different methods

We first focus on the supervised case with $\varphi = 100\%$ and compare the classification accuracy of test documents for the supervised CSTM and its competitors. Table 3 displays the results. We observe that CSTM outperforms all other tested methods for this example. We also conduct extensive simulation studies to compare the supervised CSTM and its competitors; see Appendix G for details. The simulation results show that, when viewing the data as generated from a topic model, the advantage of CSTM over its competitors is largest (1) when differences between classes in topic probabilities are moderate and (2) when topic probabilities differ between classes but differences between topics in word probabilities are moderate. In Appendix K, we apply the supervised CSTM and its competitors to analyze the Reuters dataset, another benchmark dataset for text classification. CSTM again outperforms the other methods.

Table 3: The Classification Accuracy of Test Documents for Supervised CSTM and Its Competitors

Method		Classification Accuracy
CSTM		85.65%
Two-stage	LDA+RF	71.80%
	LDA+SVM	70.24%
LDA per class		33.27%
Modified CCS	word	58.02%
	phrase	76.31%
sLDA		73.30%
DiscLDA		81.38%
labeled LDA		70.26%

We next compare the semisupervised CSTM with the two-stage method and the CCS method, with φ varying from 10% to 90%. Tables 4 and 5 present the classification accuracy for unlabeled training and test documents under these models. Generally, as φ and hence the number of labeled training documents increase, all methods achieve better classification accuracy. For all but one case, CSTM performs the best among all approaches considered in classifying both the unlabeled training documents and test documents.

Table 4: Classification Accuracy of Unlabeled Training Documents Under Different Models

φ	CSTM	LDA +RF	LDA +SVM	Modified CCS (word)	Modified CCS (phrase)
10%	64.81%	67.60%	59.86%	15.76%	41.49%
20%	74.50%	69.89%	63.62%	17.01%	52.64%
30%	76.85%	71.30%	66.28%	27.63%	61.75%
40%	79.66%	71.49%	67.59%	32.85%	68.79%
50%	80.85%	72.17%	68.90%	39.37%	73.81%
60%	82.62%	71.89%	69.62%	43.86%	77.06%
70%	83.08%	71.59%	69.33%	51.38%	78.82%
80%	83.25%	71.31%	69.68%	58.44%	80.92%
90%	84.86%	73.42%	71.67%	62.72%	81.67%

Recall that τ is the lower limit for $\delta_{k,j}$ (the measure of the strength of the association of topic k with class j) to indicate an additional class-specific topic for class j and $h_{S,\max}$ is the upper limit

Table 5: Classification Accuracy of Test Documents Under Different Models

φ	CSTM	LDA +RF	LDA +SVM	Modified CCS (word)	Modified CCS (phrase)
10%	68.95%	64.78%	50.38%	16.74%	37.75%
20%	75.84%	66.50%	58.72%	21.12%	47.36%
30%	76.45%	67.64%	61.32%	29.45%	54.29%
40%	78.88%	69.54%	65.16%	33.25%	60.58%
50%	80.50%	69.95%	66.09%	37.53%	65.43%
60%	81.51%	69.88%	66.95%	41.34%	68.55%
70%	82.73%	71.39%	67.52%	46.28%	70.99%
80%	83.90%	70.80%	68.28%	50.61%	72.98%
90%	85.01%	71.87%	69.03%	54.43%	74.75%

for the number of shared topics (see Section 4.2 for details). In Appendix H, using $\varphi = 20\%$ as an example, we show the candidate sets of values for h_j and h_S under different settings of τ and $h_{S,\max}$ and the accuracy of the corresponding candidate models for classifying unlabeled training documents and test documents. In our example, given the same value of $h_{S,\max}$, as τ increases (or as the model becomes simpler), the classification accuracy increases; given the same value of τ, as $h_{S,\max}$ increases (or as the model becomes more complicated), the classification accuracy first increases and then decreases. The best classification accuracy is achieved with $\tau \geq 0.35$ and $h_{S,\max} = 5$.

The phrase-based modified CCS works much better than the word-based modified CCS and eventually outperforms the two-stage approaches based on LDA for unlabeled training documents when $\varphi \geq 50\%$ and for testing documents when $\varphi \geq 80\%$. The better performance of the phrase-based modified CCS compared with the word-based modified CCS might be due to the use of bigrams and trigrams by the former, which helped it to capture the dependencies among words. Identifying such dependencies is important for better classification accuracy. Although the size of the dictionary becomes much larger when bigrams and trigrams are used, the modified CCS uses supervised sparse classification methods to select only a small set of phrases. Therefore, the use of a larger dictionary does not hurt classification accuracy for unlabeled documents.

In the LDA and CSTM approaches, dependencies among words are captured by their possible co-occurrences within a topic. The reason that the two-stage approaches based on LDA performed worse than the phrases-based modified CCS for larger values of φ might be that the topics extracted by LDA, and hence the within-topic co-occurrence patterns for words, have a lot of noise for discriminating between classes. In Appendix I, we investigate the performance of LDA and CSTM using bigrams or trigrams as the basic analysis units. The classification accuracy is generally slightly worse than that using words as the basic analysis units. Given the larger dictionary resulting from using bigrams or trigrams, the LDA and CSTM models become more complicated, which hurts the classification accuracy for unlabeled documents.

To understand how and in what circumstances CSTM performs better than competing approaches, we investigate the case with $\varphi = 20\%$. The test classification accuracy for each class is reported in Table 6, where those for words-based modified CCS are omitted. For most classes, CSTM has better classification accuracy than the other approaches.

The classification accuracy of each approach is related to words or phrases that the approach finds to be discriminative between classes. For each topic k in CSTM, Table 7 lists the top ten "most probable" words with highest posterior means of $\phi_{k,v}$. For LDA, there is no direct relationship

Table 6: Detailed Classification Accuracy of Test Documents Under Different Models when $\varphi = 20\%$

Class Group	Class	CSTM	LDA +RF	LDA +SVM	Modified CCS(phrase)
Computer Science	comp.graphics	62.72%	43.88%	26.02%	50.00%
	comp.os.ms-windows.misc	51.27%	60.67%	49.61%	54.50%
	comp.sys.ibm.pc.hardware	68.78%	53.06%	29.70%	43.15%
	comp.sys.mac.hardware	87.24%	57.14%	77.92%	51.17%
	comp.windows.x	47.53%	53.16%	42.03%	52.15%
	Group Average	63.51%	53.58%	45.06%	50.19%
For Sale	misc.forsale	66.67%	72.05%	50.51%	81.54%
Autos & Sports	rec.autos	88.64%	88.94%	76.01%	50.76%
	rec.motorcycles	90.40%	89.20%	69.60%	55.53%
	rec.sport.baseball	96.98%	93.95%	89.17%	51.64%
	rec.sport.hockey	97.48%	92.23%	76.69%	45.11%
	Group Average	93.38%	91.08%	77.87%	50.76%
Science	sci.crypt	88.38%	85.10%	65.66%	45.45%
	sci.electronics	71.25%	58.38%	47.84%	32.32%
	sci.med	54.45%	65.15%	37.88%	22.73%
	sci.space	80.81%	77.78%	60.41%	49.24%
	Group Average	73.72%	71.60%	52.94%	37.43%
Politics	talk.politics.guns	92.89%	79.67%	76.10%	39.84%
	talk.politics.mideast	95.48%	65.96%	59.31%	52.39%
	talk.politics.misc	56.57%	43.23%	35.48%	35.48%
	Group Average	81.65%	62.95%	56.96%	42.57%
Religion	alt.atheism	73.67%	56.74%	57.99%	42.63%
	soc.religion.christian	70.35%	72.86%	70.10%	52.01%
	talk.religion.misc	39.84%	21.51%	19.52%	29.88%
	Group Average	61.29%	50.37%	49.21%	41.51%

between the discovered topics and the classes. We define a leading topic for each class j as the topic with the largest value of $\rho_{k,j}$ defined in Equation (1). Table A7 in Appendix J lists the top ten words with the highest posterior means of $\phi_{k,v}$ under each leading topic.

For the class “comp.sys.ibm.pc.hardware”, CSTM has much higher classification accuracy than the other approaches. The rate of misclassifying test documents from this class into the similar class “comp.sys.mac.pc.hardware” is only 12.5% for CSTM but is much higher for the other approaches. CSTM extracts a class-specific topic with the words “Apple” and “Mac” under the class “comp.sys.mac.hardware” (see Table 7), thus reducing confusion between these two classes. In LDA, the topic associated with the word “Mac” is the leading topic for both classes (see Table A7 in Appendix J); for the modified CCS approach, the phrases “Mac IBM”, “pc Mac” and “IBM pc” are selected for both classes.

For the class “talk.politics.mideast”, CSTM also performs much better than the other approaches. The weak performance of the two-stage approaches is because topics generated by LDA have mixed meanings of politics, Mideast and religion. For example, the topic with the second largest value of $\rho_{k,j}$ for the class “talk.politics.mideast” is featured with words such as “Israel”, “Jews” and “policy”, and the topic with the third largest value of $\rho_{k,j}$ is featured with words such as “atheism”, “political” and “religion”. Hence, a large proportion of documents in the class “talk.politics.mideast” are wrongly classified into the class “alt.atheism”, “talk.religion.misc”, or “talk.politics.misc”. The mixed-meaning problem also exists for the modified CCS approach, whose selected phrases are not targeted to these specific classes. For example, the phrases “Turkey politics”, “political atheists”, “religious wars” and “mideast” are selected for all of the above four classes. These analyses suggest that CSTM is able to capture subtle differences between similar classes.

5.5 Text summarization performances of different methods

We first compare the meaning of the topics extracted by CSTM and LDA. For CSTM, the top ten words under each class-specific topic can be used as the text summary for the corresponding class. From the results in Table 7, it is obvious that the meaning of the top words under each class-specific topic is closely related to the corresponding class. The top words under the shared topics do not seem to be related to any specific class. For LDA, the top ten words under each leading topic for a class can be used as a text summary for the class. Based on the results in Table A7 in Appendix J, the meaning of the leading topics is not necessarily relevant to the classes, and there are several cases in which different classes have the same leading topic. The topics extracted by CSTM are therefore more meaningful than those extracted by LDA.

For the phrase-based modified CCS, Table A8 in Appendix J presents the ten phrases that first enter the L^1 penalized binary logistic regression for each class, which can be treated as a text summary for the class. By using phrases as terms in the dictionary, this approach has the benefit of being able to detect meaningful phrases such as “VGA graphics mode”, “Windows operating system” and “answering machine”. However, the selected phrases are not always complete or meaningful. For example, the phrase “Windows operating” is part of the meaningful phrase “Windows operating system”, and the phrases “use VGA” and “program produces” are not meaningful.

To compare the summarization performances of CSTM and the modified CCS, we take a closer look at the ten words extracted by CSTM and the ten phrases extracted by the modified CCS for the class “misc.forsale”, for which the modified CCS achieves the highest classification accuracy (see Table 6). For each of the ten words extracted by CSTM for the class “misc.forsale”, Figure

Table 7: Top Ten Words with Highest Probabilities Under the Class-Specific Topic for Each Class and the Shared Topics When $\varphi = 20\%$ (CSTM)

Topic	Top Ten Words with Highest Probabilities
comp.graphics	lines,subject,graphics,file,writes,article,polygon,points,DOS, image
comp.os.ms-windows.misc	Windows,lines,organization,university,file,use,DOS,version, program,system
comp.sys.ibm.pc.hardware	drive,SCSI,card,university,system,Windows,use,MB,disk,IDE
comp.sys.mac.hardware	Apple,know,Mac,like,drive,use,new,problem,computer,monitor
comp.windows.x	Window,file,use,program,server,available,motif,widget,set, application
misc.forsale	sale,new,please,offer,shipping,used,interested,price,asking, condition
rec.autos	car,like,good,new,engine,oil,time,speed,drive,dealer
rec.motorcycles	bike,DOD,like,ride,motorcycle,good,BMW,riding,go,helmet
rec.sport.baseball	year,game,team,university,baseball,last,players,season,win, first
rec.sport.hockey	team,game,play,hockey,NHL,go,season,VS,players,period
sci.crypt	key,encryption,clipper,chip,use,government,security,privacy, information,public
sci.electronics	use,power,need,circuit,current,work,ground,AMP,voltage, radar
sci.med	MSG,banks,people,food,use,medical,disease,patients,science, health
sci.space	space,NASA,launch,orbit,Moon,lunar,data,Earth,satellite, system
talk.politics.guns	gun,people,guns,right,weapons,firearms,government,use, control,law
talk.politics.mideast	Israel,Turkish,people,Israeli,Jews,Armenian,Turks,Armenia, organization,Turkey
talk.politics.misc	president,people,new,Clayton,Cramer,men,gay,government, war,south
alt.atheism	God,organization,people,think,atheists,morality,objective, believe,moral,must
soc.religion.christian	God,Jesus,people,Bible,think,Christians,believe,church,Christ, faith
talk.religion.misc	Judas,new,Greek,Christian,acts,Bible,Matthew,man, Christianity,iniquity
shared topics	would,subject,lines,organization,one,writes,people,like,article, know forest,old,experiment,clouds,flyby,funk,proceed,raffle, acquaintance,prominent trouble,likely,survivor,washes,Muhammad,cassettes,race,road, related,faith method,mate,plan,automatical,reboot,alphabetic,reused, removed,thought,advise employed,commissions,cut,diagnose,relaxes,join,adult, outbreaks,pilot,school

6 shows the total number of documents that contain the word and the proportion of documents in each class that contain the word. For each of the ten phrases extracted by the modified CCS for the class “misc.forsale”, Figure 7 shows the total number of documents that contain the phrase and the proportion of documents in each class that contain the phrase.

For CSTM, the extracted ten words all have total frequencies greater than 200. For eight of the ten words, the class “misc.forsale” has the highest proportion of documents containing the word. For the modified CCS, for nine of the extracted ten phrases, the class “misc.forsale” has the highest proportion of documents containing the phrase. However, eight of the ten extracted phrases have total frequencies lower than 100, among which six phrases have total frequencies lower than 20.

The same phenomenon is observed for the other 19 classes (results not shown). Therefore, although the modified CCS can discover phrases that can discriminate between classes, these phrases are often low-frequency ones and are not appropriate for summarizing the main content in each class. By contrast, CSTM discovers high-frequency words that can discriminate between classes and hence outperforms the modified CCS in delivering class-specific text summaries.

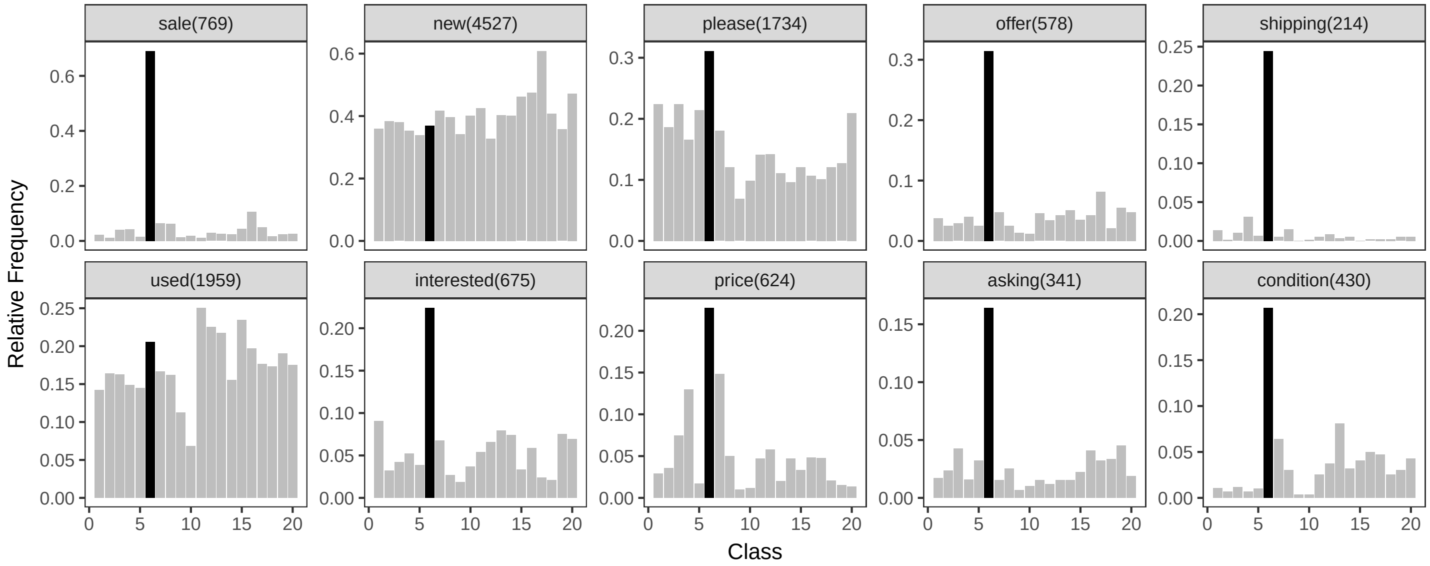

Figure 6: For each of the ten words extracted by CSTM for the class “misc.forsale”, the title of the corresponding subplot presents the word and the total number of documents that contain the word. The corresponding subplot shows the proportion of documents in each class that contain the word, with the black bar indicating the class “misc.forsale”.

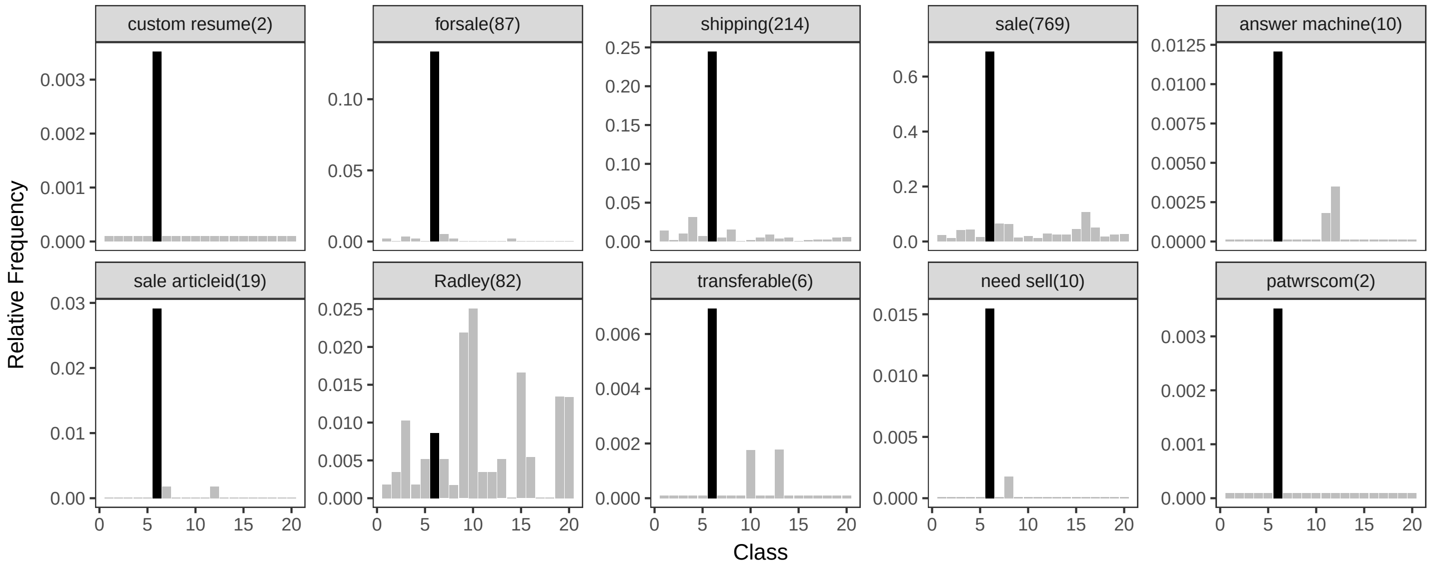

Figure 7: For each of the ten phrases extracted by the modified CCS for the class “misc.forsale”, the title of the corresponding subplot presents the phrase and the total number of documents that contain the phrase. The corresponding subplot shows the proportion of documents in each class that contain the phrase, with the black bar indicating the class “misc.forsale”.

5.6 Runtime comparison of different methods

This subsection first analytically compares the runtime between CSTM and its competitors and then presents the real runtime comparison on the 20 Newsgroups dataset.

5.6.1 ANALYTICAL RUNTIME COMPARISON

We now compare the computational complexity of CSTM to that of the two-stage approach based on LDA.

For the two-stage approach, most of the computational time is spent on obtaining the topic probabilities for training documents and predicting the topic probabilities for test documents. The collapsed Gibbs sampling method (Liu, 1994; Griffiths and Steyvers, 2004) is used for both training and prediction, where the topic indicators $z_{d,n}$ need to be updated in each iteration. With K^{LDA} topics, D training documents and the average document length $\bar{N}$, the computational complexity in training is $O(K^{\text{LDA}}D\bar{N})$. Similarly, the computational complexity in prediction is $O(K^{\text{LDA}}D^{\text{test}}\bar{N}^{\text{test}})$, where D^{test} is the number of test documents and $\bar{N}^{\text{test}}$ is the average length of test documents.

CSTM is trained using a Gibbs sampling algorithm detailed in Appendix C. Each iteration consists of several steps. In the first step, class probabilities $\boldsymbol{\eta}$ are updated, with computational complexity on the order of $O(J)$. In the second step, each topic indicator $z_{d,n}$ is updated within its class-specific topic set $\boldsymbol{\Lambda}_d$. The computational complexity is $O((\bar{h}_C + h_S)D\bar{N})$, where $\bar{h}_C$ is the average number of class-specific topics for each class. In the third step, for each topic k, the vector of word probabilities $\boldsymbol{\phi}_k$ is updated, with computational complexity $O(KV)$. In the fourth step, for each of the D_1 labeled documents, the nonzero elements of $\boldsymbol{\theta}_d$ over $\boldsymbol{\Lambda}_d$ are updated, with computational complexity $O((\bar{h}_C + h_S)D_1)$. In the fifth step, for each of the $D - D_1$ unlabeled documents, the class label y_d and the nonzero elements of θ_d are updated together, with computational complexity $O((J + \bar{h}_C + h_S)(D - D_1))$. The total computational complexity is $O(J+(\bar{h}_C+h_S)D\bar{N}+KV+(\bar{h}_C+h_S)D+J(D-D_1))$. Because $\bar{h}_C+h_S < K^{\text{LDA}}$, $K \leq K^{\text{LDA}}$ and $V < D\bar{N}$, the order of computational complexity in each Gibbs iteration for the CSTM is smaller than that for the two-stage approach based on LDA.

For the CSTM, class labels of the test documents are predicted based on a fixed number of posterior samples of $(\boldsymbol{\eta}, \boldsymbol{\Phi})$ (for details, see Appendix E). For each posterior sample of $(\boldsymbol{\eta}, \boldsymbol{\Phi})$, there are two steps: (1) conditional on $y_d = j$ ($j = 1, \cdots, J$), apply the EM algorithm to obtain estimates of the topic probabilities, $\hat{\boldsymbol{\theta}}_{d,k}$ for $k \in \Lambda_j$; (2) apply the Bayes rule to calculate the class probabilities for each document.

In the E-step of the EM algorithm, for each class j ($j = 1, \cdots, J$), conditional on $y_d = j$, the conditional expectation of $I(z_{d,n} = k)$ for $k \in \boldsymbol{\Lambda}_j$ is calculated. The computational complexity is $O((\bar{h}_C + h_S)D^{\text{test}}\bar{N}^{\text{test}}J)$. In the M-step of the EM algorithm, for each class j ($j = 1, \cdots, J$), conditional on $y_d = j$, the nonzero elements of $\boldsymbol{\theta}_d$ over $\boldsymbol{\Lambda}_j$ are updated. The computational complexity is $O((\bar{h}_C + h_S)D^{\text{test}}J)$. Applying the Bayes rule involves two steps. First, the likelihood of $\boldsymbol{w}_d$ under each class j is calculated, where all possible values of $z_{d,n} \in \boldsymbol{\Lambda}_j$ are summed for each word n. The computational complexity is $O((\bar{h}_C + h_S)D^{\text{test}}\bar{N}^{\text{test}}J)$. Second, the probability that each document d belongs to each class j is calculated. The computational complexity is $O(D^{\text{test}}J)$. The total computational complexity for prediction in CSTM is $O((\bar{h}_C + h_S)D^{\text{test}}\bar{N}^{\text{test}}J + (\bar{h}_C + h_S)D^{\text{test}}J + D^{\text{test}}J)$. If $(\bar{h}_C + h_S)J$ is smaller than (equal to, or larger than) K^{LDA}, the order of computational complexity for prediction in CSTM is smaller than (equal to, or larger than) that in the two-stage approach based on LDA.

5.6.2 REAL RUNTIME COMPARISON

Taking the 20 Newsgroups dataset as an example, we first consider an ideal scenario in which there are an unlimited number of cores and all independent tasks can be run in parallel. We then discuss a more realistic scenario in which only 16 independent tasks can be run in parallel.

For the two-stage approach based on LDA, a model needs to be trained for each candidate number of topics, which ranges from 1 to 201 with an increment of 5. There are 40 models in total. In the ideal scenario, these 40 models can be trained in parallel. We regard the training time as the time for training the most complicated model with 201 topics. We regard the prediction time as the time for using the optimal model with 71 topics to predict the labels for test documents. Specifically, we apply the GibbsLDA++ algorithm implemented in C/C++ (Phan and Nguyen, 2007). In training each candidate model, we run an MCMC chain with 3000 iterations, of which the first 500 iterations are discarded as burn-in and every 25th iteration of the remaining 2500 iterations is kept. In prediction, we run 500 iterations and keep every 25th iteration of the last 125 iterations. Almost the entire prediction time is used for predicting the topic probabilities, and the time for training or applying RF or SVM is negligible. On a Dell XPS laptop with 2.8GHz CPU and 8Gb RAM, the training time is 625 minutes, and the prediction time is 35 minutes.

For full-fledged CSTM with model selection, we first need to train a set of LDA models and select the optimal number of topics. Afterwards, almost the entire training time is used in two tasks: (1) for each training subsample in the fivefold cross-validation, running three chains for each candidate set of values for h_j's and h_S; (2) for each validation subsample, using posterior samples from the selected chain for each candidate model to predict the class labels. Since there are 16 candidate sets of values for h_j and h_S (see Appendix H for detail), there are $16 \times 5 \times 3 = 240$ subtasks for task (1) and $16 \times 5 = 80$ subtasks for task (2). In the ideal scenario, all 240 or 80 subtasks can be run in parallel. We regard the training time for CSTM as the time for training an LDA model with 201 topics, plus the longest time for running one chain of a candidate CSTM model on a training subsample, plus the longest time for predicting class labels for a validation subsample using posterior samples from the selected chain for a candidate model. The prediction task uses the five selected chains for the optimal model to predict the class labels for test documents. Prediction using these five chains can be run in parallel. The prediction time is the time for using one selected chain for the optimal model to predict the class labels for test documents.

Similarly, we calculate the training time and the prediction time under the ideal scenario for the other methods. For sLDA, the code accompanying Blei and McAuliffe (2007) is used. Since there are no codes accompanying the original work for DiscLDA in Lacoste-Julien et al. (2008) and for labeled LDA in Ramage et al. (2009), we instead use some open-source codes on GitHub.

For each method other than the two-stage approach, Table 8 presents the ratio of the training time or prediction time to that of the two-stage approach in the supervised case under the ideal scenario. The computational time for CSTM, sLDA and DiscLDA is approximately twice that for the two-stage approach. In terms of other models, labeled LDA takes much less computational time than the two-stage approach due to its smaller number of topics. The LDA per class approach and the modified CCS methods have negligible computational time compared with the two-stage approach.

For CSTM and the modified CCS methods, Table 9 presents the ratio of the training time or prediction time to those of the two-stage approach in the semisupervised case under the ideal scenario. The training time for CSTM is less than 2.5 times that for the two-stage approach, and the prediction

Table 8: Ratio of training time or prediction time to that of the two-stage approach in the supervised case (ideal scenario)

	Training	Prediction
CSTM	1.903	1.093
LDA per class	0.003	0.008
Modified CCS (word)	0.006	0.060
Modified CCS (phrase)	0.020	0.115
sLDA	2.031	0.912
DiscLDA	1.784	0.827
labeled LDA	0.236	0.281

time for CSTM is comparable to that for the two-stage approach. As the fraction of labeled training documents φ decreases, the training time for CSTM is longer.

Table 9: Ratio of training time or prediction time to that of the two-stage approach in the semisupervised case (ideal scenario)

	CSTM		Modified CCS (word)		Modified CCS (phrase)	
φ	Training	Prediction	Training	Prediction	Training	Prediction
10%	2.446	0.963	0.001	0.057	0.002	0.113
20%	2.390	0.981	0.001	0.058	0.003	0.115
30%	2.349	0.944	0.002	0.056	0.005	0.112
40%	2.285	0.926	0.003	0.059	0.006	0.114
50%	2.219	0.963	0.004	0.055	0.007	0.112
60%	2.148	1.019	0.004	0.060	0.010	0.116
70%	2.099	0.981	0.005	0.056	0.012	0.115
80%	2.046	1.130	0.006	0.057	0.014	0.117
90%	1.950	1.152	0.006	0.059	0.016	0.118

We next compare the runtimes of different methods under the more realistic scenario when only 16 independent tasks can be run in parallel. Take the two-stage approach as an example. In training, there are 40 candidate models, and we need $\lceil 40/16 \rceil = 3$ runs. For simplicity, we assume that the runtime for each run equals the training time in the ideal scenario. The training time under the realistic scenario is then $625 \times 3 = 1875$ minutes. Because the prediction only uses one model and does not need parallel computing, the prediction time under the realistic scenario is the same as that under the ideal scenario and equals 35 minutes. We calculate the training time and prediction time for the other methods similarly. Table 10 and Table 11 present the ratio of the training time or prediction time to that of the two-stage approach in the supervised and semisupervised cases under the realistic scenario.

As mentioned in Section 5.1, in training the supervised CSTM, we can just run one chain instead of three chains for each training subsample and each candidate model. This does not make a difference under the ideal scenario but makes a difference under the realistic scenario. As shown in the first column of Table 10, running only one chain reduces the computational time by $(5.469 - 2.505)/5.469 = 54.2\%$. With one chain under each setting, training the supervised CSTM takes $2.505 \times 1875/60/24 = 3.26$ days. In the semisupervised case with $\varphi = 20\%$, training the

Table 10: Ratio of training time or prediction time to that of the two-stage approach in the supervised case (realistic scenario)

	Training	Prediction
CSTM (3 chains)	5.469	1.093
CSTM (1 chain)	2.505	1.093
LDA per class	0.024	0.016
Modified CCS (word)	0.002	0.060
Modified CCS (phrase)	0.007	0.115
sLDA	2.196	0.912
DiscLDA	2.064	0.827
labeled LDA	0.079	0.281

Table 11: Ratio of training time or prediction time to that of the two-stage approach in the semisupervised case (realistic scenario)

	CSTM		Modified CCS (word)		Modified CCS (phrase)	
φ	Training	Prediction	Training	Prediction	Training	Prediction
10%	8.225	0.963	0.000	0.057	0.001	0.113
20%	7.941	0.981	0.000	0.058	0.001	0.115
30%	7.731	0.944	0.001	0.056	0.002	0.112
40%	7.406	0.926	0.001	0.059	0.002	0.114
50%	7.072	0.963	0.001	0.055	0.002	0.112
60%	6.712	1.019	0.001	0.060	0.003	0.116
70%	6.463	0.981	0.002	0.056	0.004	0.115
80%	6.193	1.130	0.002	0.057	0.005	0.117
90%	5.708	1.152	0.002	0.059	0.005	0.118

CSTM takes $7.941 \times 1875/60/24 = 10.34$ days. Whether this time is worthwhile depends on practical considerations such as the importance of classification accuracy. One can also reduce the computational time by reducing the number of candidate models by increasing the lower limit τ or decreasing the maximum number of shared topics $h_{S,\max}$. Once the optimal CSTM model is chosen, prediction of class labels for the test documents takes much less time. In the semisupervised case with $\varphi = 20\%$, predicting the class labels for 7,532 test documents takes $0.981 \times 35 = 34.33$ minutes. Because the prediction can be done independently across documents, on average predicting the class label for one document takes $34.33 \times 60/7532 = 0.27$ seconds.

5.7 Robustness check

The hyperparameter α governs the prior distribution of a document's topic probabilities and can be seen as the pseudo count of topic k in a document. In addition to setting $\alpha = 0.5$, we also consider setting $\alpha = 0.1$ or $\alpha = 1$. The hyperparameter $\boldsymbol{\beta}$ governs the prior distribution of a topic's word probabilities, where β_v can be seen as the pseudo count for word v and $\sum_{v=1}^{V} \beta_v$ can be seen as the total pseudo count for all words. When $\beta_v = 0.1$ for $v = 1, \cdots, V$, the total pseudo count is $0.1V$. We also consider two alternative specifications of $\boldsymbol{\beta}$ where the total pseudo count is again $0.1V$ but the distribution of the pseudo counts over words varies with the word counts in the training documents. In the first alternative specification, we set $\beta_v \propto c_v$, where c_v denotes the count of the vth word in the training documents. Since the distribution of the original word counts could be very skewed, we also consider a second alternative specification that is less skewed: $\beta_v \propto \sqrt{c_v}$.

For varying values of the fraction of labeled training documents φ, under each setting of α and $\boldsymbol{\beta}$, Table 12 presents the selected values of h_j and h_S, as well as the classification accuracy of test documents. For each value of φ, the selected values of h_j and h_S are the same, and the classification accuracy of test documents is similar under different settings of α and β. For each value of φ, the maximum absolute difference in classification accuracy among the nine settings is less than 2%. These results demonstrate that CSTM's performance is robust to different settings of α and $\boldsymbol{\beta}$.

6. Discussion

While early research on text classification and summarization has focused on using term frequencies, topic models have opened new paths and proved to be useful. When the basic topic model LDA is applied to all documents without reference to their class labels, all of the topics are shared across classes. The extracted topics are noisy, do not have strong discriminating power between classes, and provide poor class-specific summaries. The LDA per class approach applies LDA separately to documents within each class, and all of the topics are class-specific. However, common contents appearing across classes may appear in different sets of topics under each class, which adds confusion between classes and is not helpful for class discrimination. In addition, the meanings of the extracted topics have to be manually checked in order to remove topics associated with common contents and keep the remaining ones for class-specific summaries.

CSTM serves as an intermediate model between the above two extremes, with both topics that are shared across classes and topics that are class-specific. Since the shared topics have accounted for common contents appearing across classes, the extracted class-specific topics allow for stronger discriminating power and automatically provide better class-specific text summaries.

Compared with the existing supervised topic models such as sLDA, DiscLDA and labeled LDA, CSTM is flexible enough to capture content similarities and differences across classes and is

Table 12: Classification Accuracy of Test Documents Under Different Settings of α and β

φ	Selected h_j and h_S	$\boldsymbol{\beta}$	$\alpha = 0.1$	$\alpha = 0.5$	$\alpha = 1$
10%	all $h_j = 1, h_s = 5$	$\beta_v = 0.1$	68.25%	68.95%	69.25%
		$\beta_v \propto c_v$	68.76%	67.88%	67.32%
		$\beta_v \propto \sqrt{c_v}$	67.43%	67.41%	67.23%
20%	all $h_j = 1, h_s = 5$	$\beta_v = 0.1$	75.77%	75.84%	75.63%
		$\beta_v \propto c_v$	75.09%	74.94%	75.60%
		$\beta_v \propto \sqrt{c_v}$	74.76%	75.96%	76.22%
30%	all $h_j = 1, h_s = 5$	$\beta_v = 0.1$	76.15%	76.45%	76.31%
		$\beta_v \propto c_v$	76.42%	76.32%	76.48%
		$\beta_v \propto \sqrt{c_v}$	76.49%	77.22%	76.53%
40%	all $h_j = 1, h_s = 5$	$\beta_v = 0.1$	77.81%	78.88%	78.72%
		$\beta_v \propto c_v$	79.82%	78.07%	77.34%
		$\beta_v \propto \sqrt{c_v}$	78.26%	79.52%	78.54%
50%	all $h_j = 1, h_s = 5$	$\beta_v = 0.1$	79.26%	80.50%	80.37%
		$\beta_v \propto c_v$	79.99%	79.66%	80.24%
		$\beta_v \propto \sqrt{c_v}$	80.76%	80.15%	79.56%
60%	all $h_j = 1, h_s = 5$	$\beta_v = 0.1$	80.11%	81.47%	81.51%
		$\beta_v \propto c_v$	80.71%	81.53%	80.80%
		$\beta_v \propto \sqrt{c_v}$	81.27%	80.87%	81.77%
70%	all $h_j = 1, h_s = 5$	$\beta_v = 0.1$	83.24%	82.73%	82.69%
		$\beta_v \propto c_v$	83.42%	83.31%	82.62%
		$\beta_v \propto \sqrt{c_v}$	82.67%	82.79%	82.58%
80%	all $h_j = 1, h_s = 10$	$\beta_v = 0.1$	83.71%	83.90%	84.26%
		$\beta_v \propto c_v$	82.87%	84.15%	84.32%
		$\beta_v \propto \sqrt{c_v}$	83.94%	83.48%	84.12%
90%	all $h_j = 1, h_s = 10$	$\beta_v = 0.1$	84.87%	85.01%	85.25%
		$\beta_v \propto c_v$	83.89%	85.31%	84.73%
		$\beta_v \propto \sqrt{c_v}$	84.63%	85.39%	84.72%
100%	all $h_j = 1, h_s = 10$	$\beta_v = 0.1$	85.67%	85.65%	85.49%
		$\beta_v \propto c_v$	85.28%	84.66%	86.20%
		$\beta_v \propto \sqrt{c_v}$	85.67%	86.10%	86.26%

parsimonious enough to have good generalization ability. CSTM is also a promising approach for text classification and summarization in the semisupervised scenario.

Acknowledgments

We would like to acknowledge support for this project from the National Natural Science Foundation of China (72001205, 11971504), the fund for building world-class universities (disciplines) of Renmin University of China, the Fundamental Research Funds for the Central Universities and the Research Funds of Renmin University of China (2021030047), Foundation from Ministry of Education of China (20JZD023), and the Ministry of Education's Focus on Humanities and Social Science Research Base (Major Research Plan 17JJD910001).

Appendix A: Full posterior distribution for LDA

Throughout the article, we adopt the following notations: $\boldsymbol{\theta}_d = (\theta_{d,1}, ..., \theta_{d,K})^\top$ denotes the topic probabilities for document d, and $\boldsymbol{\phi}_k = (\phi_{k,1}, ..., \phi_{k,V})^\top$ denotes the word probabilities for topic k; $w_{d,n}$ is the nth word in document d, $z_{d,n}$ indicates the topic associated with it, and $\boldsymbol{w}_d = (w_{d,1}, ..., w_{d,N_d})^\top$ and $\boldsymbol{z}_d = (z_{d,1}, ..., z_{d,N_d})^\top$ are the vector forms of the words and topics indicators, respectively. We let $\boldsymbol{z} = \{\boldsymbol{z}_1, ..., \boldsymbol{z}_D\}$ and $\boldsymbol{w} = \{\boldsymbol{w}_1, ..., \boldsymbol{w}_D\}$ denote the collection of these indicator and word vectors for all training documents. Finally, we denote $\boldsymbol{\Theta} = \{\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_D\}$ and $\boldsymbol{\Phi} = \{\boldsymbol{\phi}_1, ..., \boldsymbol{\phi}_K\}$.

According to the generative process of LDA in Figure 1, the full posterior distribution of $(\boldsymbol{\Theta}, \boldsymbol{\Phi}, \boldsymbol{z})$ is as follows.

$$\begin{aligned}
& f\left(\boldsymbol{\Theta}, \boldsymbol{\Phi}, \boldsymbol{z}|\boldsymbol{w}, \alpha, \boldsymbol{\beta}\right) \propto f\left(\boldsymbol{\Theta}|\alpha\right) f\left(\boldsymbol{\Phi}|\boldsymbol{\beta}\right) f\left(\boldsymbol{w}, \boldsymbol{z}|\boldsymbol{\Theta}, \boldsymbol{\Phi}\right) \\
& \propto \left(\prod_{d=1}^{D}\prod_{k=1}^{K}\theta_{d,k}^{\alpha-1}\right)\left(\prod_{k=1}^{K}\prod_{v=1}^{V}\phi_{k,v}^{\beta_v-1}\right)\left\{\prod_{d=1}^{D}\prod_{n=1}^{N_d}\left(\theta_{d,z_{d,n}}\phi_{z_{d,n},w_{d,n}}\right)\right\} \\
& \propto \left(\prod_{d=1}^{D}\prod_{k=1}^{K}\theta_{d,k}^{\alpha+n_{d,k}^{(1)}-1}\right)\left(\prod_{k=1}^{K}\prod_{v=1}^{V}\phi_{k,v}^{\beta_v+n_{k,v}^{(2)}-1}\right),
\end{aligned} \tag{3}$$

where $n_{d,k}^{(1)} = \sum_{n=1}^{N_d} I(z_{d,n} = k)$ denotes the number of words in document d that are associated with topic k, $n_{k,v}^{(2)} = \sum_{d=1}^{D}\sum_{n=1}^{N_d} I(z_{d,n} = k \;\&\; w_{d,n} = v)$ denotes the number of times the vth word in the dictionary is associated with topic k in the training documents. Here $I(\cdot)$ is the general indicator function.

Appendix B: Full posterior distribution for CSTM

CSTM partitions latent topics into h_j class-specific ones associated with each class j ($j = 1, ..., J$), and h_S "shared" topics. Let $\boldsymbol{H}_j$ be the set of indexes for topics specific to class j ($j = 1, ..., J$), let $\boldsymbol{H}_C = \cup_{j=1}^{J}\boldsymbol{H}_j$ denote the set of all class-specific topics, and let $\boldsymbol{H}_S$ denote the set of shared topics. Each document in class j is a probabilistic mixture of topics in the subset $\boldsymbol{\Lambda}_j = \boldsymbol{H}_j \cup \boldsymbol{H}_S$. The total number of topics is $K = \sum_{j=1}^{J} h_j + h_S$. Let $\boldsymbol{\eta} = (\eta_1, \cdots, \eta_J)^\top$ denote the probabilities for the J classes. Let $y_d \in \{1, 2, \cdots, J\}$ denote the class label for document d, and let $\boldsymbol{y} = \{y_1, \cdots, y_D\}$.

According to the generative process of CSTM in Figure 4, the joint distribution of $(\boldsymbol{\eta}, \boldsymbol{y}, \boldsymbol{\Theta}, \boldsymbol{\Phi}, \boldsymbol{z}, \boldsymbol{w})$ given $(\alpha, \boldsymbol{\beta}, \gamma)$ is

$$\begin{aligned}
&f\left(\boldsymbol{\eta}, \boldsymbol{y}, \boldsymbol{\Theta}, \boldsymbol{\Phi}, \boldsymbol{z}, \boldsymbol{w} | \alpha, \boldsymbol{\beta}, \gamma\right) \\
\propto & f\left(\boldsymbol{\eta}|\gamma\right) \prod_{k=1}^{K} f\left(\boldsymbol{\phi}_k|\boldsymbol{\beta}\right) \prod_{d=1}^{D} \left\{ f\left(y_d|\boldsymbol{\eta}\right) f\left(\boldsymbol{\theta}_d|\alpha, y_d\right) \prod_{n=1}^{N_d} \left[f\left(z_{d,n}|\boldsymbol{\theta}_d\right) f\left(w_{d,n}|\boldsymbol{\Phi}, z_{d,n}\right) \right] \right\} \\
\propto & \left(\prod_{j=1}^{J} \eta_j^{\gamma-1} \right) \left(\prod_{k=1}^{K} \prod_{v=1}^{V} \phi_{k,v}^{\beta_v - 1} \right) \\
& \prod_{d=1}^{D} \left[\eta_{y_d} \left(\frac{\Gamma((h_{y_d} + h_S)\alpha)}{[\Gamma(\alpha)]^{h_{y_d}+h_S}} \prod_{k \in \boldsymbol{\Lambda}_{y_d}} \theta_{d,k}^{\alpha-1} \right) \prod_{n=1}^{N_d} \left(\theta_{d,z_{d,n}} \phi_{z_{d,n}, w_{d,n}} \right) \right] \\
\propto & \left(\prod_{j=1}^{J} \eta_j^{\gamma + M_j - 1} \right) \left(\prod_{k=1}^{K} \prod_{v=1}^{V} \phi_{k,v}^{\beta_v + n_{k,v}^{(2)} - 1} \right) \prod_{d=1}^{D} \left(\frac{\Gamma((h_{y_d} + h_S)\alpha)}{[\Gamma(\alpha)]^{h_{y_d}+h_S}} \prod_{k \in \boldsymbol{\Lambda}_{y_d}} \theta_{d,k}^{\alpha + n_{d,k}^{(1)} - 1} \right) .
\end{aligned} \tag{4}$$

Here M_j is the number of training documents in class j.

In the semi-supervised scenario, only part of the class labels in $\boldsymbol{y}$ are known. Without loss of generality, assume that the class labels are known for the first D_1 documents, and unknown for the remaining $D - D_1$ documents. Hence, the set of observed class labels is $\boldsymbol{y}^{\text{obs}} = \{y_1, ..., y_{D_1}\}$, and the set of unknown class labels is $\boldsymbol{y}^{\text{mis}} = \{y_{D_1+1}, ..., y_D\}$. Correspondingly, M_j can be written as the sum of M_j^{obs}, the number of labeled training documents in class j, and M_j^{mis}, the number of unlabeled training documents in class j. Similarly, $n_{k,v}^{(2)}$, the number of times the vth word in the dictionary is associated with topic k, can also be written as the sum of $n_{k,v}^{(2)\text{ obs}}$ and $n_{k,v}^{(2)\text{ mis}}$ which are calculated respectively using the labeled and unlabeled training documents.

The full posterior distribution of $(\boldsymbol{\eta}, \boldsymbol{y}^{mis}, \boldsymbol{\Theta}, \boldsymbol{\Phi}, \boldsymbol{z})$ is proportional to (4), and can be written as

$$\begin{aligned}
&f\left(\boldsymbol{\eta}, \boldsymbol{y}^{mis}, \boldsymbol{\Theta}, \boldsymbol{\Phi}, \boldsymbol{z} | \boldsymbol{y}^{obs}, \boldsymbol{w}, \alpha, \boldsymbol{\beta}, \gamma\right) \\
\propto & \left(\prod_{j=1}^{J} \eta_j^{\gamma + M_j^{\text{obs}} - 1} \prod_{k=1}^{K} \prod_{v=1}^{V} \phi_{k,v}^{\beta_v + n_{k,v}^{(2)\text{ obs}} - 1} \prod_{d=1}^{D_1} \prod_{k \in \boldsymbol{\Lambda}_{y_d}} \theta_{d,k}^{\alpha + n_{d,k}^{(1)} - 1} \right) \\
& \left(\prod_{j=1}^{J} \eta_j^{M_j^{\text{mis}}} \prod_{k=1}^{K} \prod_{v=1}^{V} \phi_{k,v}^{n_{k,v}^{(2)\text{ mis}}} \prod_{d=D_1+1}^{D} \left(\frac{\Gamma((h_{y_d} + h_S)\alpha)}{[\Gamma(\alpha)]^{h_{y_d}+h_S}} \prod_{k \in \boldsymbol{\Lambda}_{y_d}} \theta_{d,k}^{\alpha + n_{d,k}^{(1)} - 1} \right) \right) ,
\end{aligned} \tag{5}$$

where the two terms in the product are related to the labeled and unlabeled training documents.

Appendix C: Details of the Gibbs Sampling algorithm for CSTM

Let $n_{k,\cdot}^{(2)\text{ obs}} = \sum_{v=1}^{V} n_{k,v}^{(2)\text{ obs}}$ and $n_{k,\cdot}^{(2)\text{ mis}} = \sum_{v=1}^{V} n_{k,v}^{(2)\text{ mis}}$. Although $\mathbf{\Theta}$ and $\mathbf{\Phi}$ can be integrated out from (5) similar to the LDA model, yielding the marginal posterior distribution

$$\begin{aligned} & f\left(\boldsymbol{\eta}, \boldsymbol{y}^{mis}, \boldsymbol{z} | \boldsymbol{y}^{obs}, \boldsymbol{w}, \alpha, \boldsymbol{\beta}, \gamma\right) \\ \propto & \left(\prod_{j=1}^{J} \eta_j^{\gamma+M_j^{\text{obs}}+M_j^{\text{mis}}-1}\right) \left\{\prod_{d=1}^{D} \frac{\prod_{k=1}^{K}\Gamma(\alpha+n_{d,k}^{(1)})}{\Gamma\left(K\alpha+N_d\right)}\right\} \prod_{d=D_1+1}^{D} \frac{\Gamma((h_{y_d}+h_S)\alpha)}{[\Gamma(\alpha)]^{h_{y_d}+h_S}} \\ & \left\{\prod_{k=1}^{K} \frac{\prod_{v=1}^{V}\Gamma(\beta_v+n_{k,v}^{(2)\text{ obs}}+n_{k,v}^{(2)\text{ mis}})}{\Gamma\left(\sum_{v=1}^{V}\beta_v+n_{k,\cdot}^{(2)\text{ obs}}+n_{k,\cdot}^{(2)\text{ mis}}\right)}\right\}, \end{aligned} \tag{6}$$

this distribution is very difficult to sample from. The reason is that for a document d with class label y_d unknown, the possible values that $z_{d,n}$ ($n = 1, ..., N_d$) can take depends on the value of y_d; hence y_d and all elements of $\boldsymbol{z}_d$ need to be sampled together. But $\boldsymbol{z}_d$ is a N_d-dimensional discrete vector with $(h_{y_d} + h_S)^{N_d}$ possible values, making such sampling extremely expensive or even impossible. Therefore collapsed Gibbs sampling cannot be applied.

We propose to use a Gibbs sampling algorithm instead. The class probabilities $\boldsymbol{\eta}$ are updated using its full conditional distribution

$$f(\boldsymbol{\eta} \mid \cdot) \propto \prod_{j=1}^{J} \eta_j^{\gamma+M_j^{\text{obs}}+M_j^{\text{mis}}-1}, \tag{7}$$

which is a Dirichlet distribution with parameters $\{\gamma + M_j^{\text{obs}} + M_j^{\text{mis}}\}_{j=1}^{J}$. For $d = 1, ..., D$ and $n = 1, ..., N_d$, the topic indicator $z_{d,n}$ is updated using its full conditional distribution

$$f(z_{d,n} = k \mid \cdot) \propto \theta_{d,k}\phi_{k,w_{d,n}},\ k \in \mathbf{\Lambda}_{y_d}, \tag{8}$$

which can be easily derived from (4). For $k = 1, ..., K$, $\boldsymbol{\phi}_k$ is updated using its full conditional distribution

$$f(\boldsymbol{\phi}_k \mid \cdot) \propto \prod_{v=1}^{V} \phi_{k,v}^{\beta_v+n_{k,v}^{(2)\text{ obs}}+n_{k,v}^{(2)\text{ mis}}-1}, \tag{9}$$

which is a Dirichlet distribution with parameters $\{\beta_v + n_{k,v}^{(2)\text{ obs}} + n_{k,v}^{(2)\text{ mis}}\}_{v=1}^{V}$. For each document $d \in \{1, \cdots, D_1\}$ whose class label is known to be j, the nonzero elements of $\boldsymbol{\theta}_d$ can be updated using their full conditional distribution

$$f\left(\{\theta_{d,k}, k \in \mathbf{\Lambda}_j\} \middle| \cdot\right) \propto \prod_{k \in \mathbf{\Lambda}_j} (\theta_{d,k})^{\alpha+n_{d,k}^{(1)}-1}, \tag{10}$$

which is a Dirichlet distribution with parameters $\{\alpha + n_{d,k}^{(1)}\}_{k\in\mathbf{\Lambda}_j}$.

The most complicated step involves drawing y_d and $\boldsymbol{\theta}_d$ together for any document $d \in \{D_1 + 1, ..., D\}$ with y_d unknown. The conditional posterior distribution of $(y_d, \boldsymbol{\theta}_d)$ given $\boldsymbol{\eta}$ and $\mathbf{\Phi}$ can be derived from (4) by summing over all possible values of $\boldsymbol{z}_d$,

$$\begin{aligned} &f(y_d, \boldsymbol{\theta}_d \mid \boldsymbol{\eta}, \boldsymbol{\Phi}, \boldsymbol{y}^{obs}, \boldsymbol{w}, \alpha, \beta, \gamma) \\ \propto &\eta_{y_d} \frac{\Gamma((h_{y_d} + h_S)\alpha)}{[\Gamma(\alpha)]^{h_{y_d}+h_S}} \prod_{k \in \boldsymbol{\Lambda}_{y_d}} (\theta_{d,k})^{\alpha-1} \prod_{n=1}^{N_d} \sum_{k \in \boldsymbol{\Lambda}_{y_d}} \left(\theta_{d,k} \phi_{k,w_{d,n}}\right). \end{aligned} \tag{11}$$

This distribution is difficult to sample from.

We propose to use the following Metropolis-Hastings algorithm. The marginal proposal distribution of y_d is given by $q(y_d) \sim Multi(\boldsymbol{\eta}^*)$, where $\boldsymbol{\eta}^* = (M_1^{\text{obs}}/D_1, ..., M_J^{\text{obs}}/D_1)$ are the class proportions in the documents with known class labels. In the conditional proposal distribution of $\boldsymbol{\theta}_d$ given y_d, to approximate the conditional distribution of $\boldsymbol{\theta}_d$ given y_d implied by (11), we assume each word $w_{d,n}$ is assigned to each topic $k \in \boldsymbol{\Lambda}_{y_d}$ with proportion $\phi_{k,w_{d,n}} / \sum_{k' \in \boldsymbol{\Lambda}_{y_d}} \phi_{k',w_{d,n}}$. This is equivalent to approximating the weighted arithmetic mean of $\theta_{d,k}$ with weights $\phi_{k,w_{d,n}} / \sum_{k' \in \boldsymbol{\Lambda}_{y_d}} \phi_{k',w_{d,n}}$ using the corresponding weighted geometric mean. Since these weights are often highly unequal across topics, the approximation is not accurate. Hence we introduce a tuning parameter ξ to adjust the weight in the proposal distribution for information coming from such approximation, in order to achieve a reasonable acceptance rate. The resulting conditional proposal distribution for $\boldsymbol{\theta}_d$ is

$$\begin{aligned} q(\boldsymbol{\theta}_d \mid y_d, \cdot) &\propto \prod_{k \in \boldsymbol{\Lambda}_{y_d}} \theta_{d,k}^{\alpha-1} \left(\prod_{n=1}^{N_d} \prod_{k \in \boldsymbol{\Lambda}_{y_d}} \theta_{d,k}^{\frac{\phi_{k,w_{d,n}}}{\sum_{k' \in \boldsymbol{\Lambda}_{y_d}} \phi_{k',w_{d,n}}}} \right)^{\xi} \\ &\propto \prod_{k \in \boldsymbol{\Lambda}_{y_d}} \theta_{d,k}^{\alpha + \xi \sum_{n=1}^{N_d} \left(\frac{\phi_{k,w_{d,n}}}{\sum_{k' \in \boldsymbol{\Lambda}_{y_d}} \phi_{k',w_{d,n}}} \right) - 1}. \end{aligned} \tag{12}$$

This is a Dirichlet distribution with parameters

$$\left\{ \alpha + \xi \sum_{n=1}^{N_d} \left(\frac{\phi_k, w_{d,n}}{\sum_{k' \in \boldsymbol{\Lambda}_{y_d}} \phi_{k',w_{d,n}}} \right) \right\}_{k \in \boldsymbol{\Lambda}_{y_d}}.$$

The joint proposal distribution for $(y_d, \boldsymbol{\theta}_d)$ is given by $q(y_d)q(\boldsymbol{\theta}_d \mid y_d, \cdot)$. The acceptance ratio can be calculated correspondingly.

We treat the supervised scenario as a special case of the semi-supervised scenario, in which $D_1 = D$ and the Metropolis-Hastings step is omitted.

Appendix D: Addressing the non-identifiability issue with CSTM

We illustrate the solution for topics within an $\boldsymbol{H}_j$, and the solution is similar for topics within $\boldsymbol{H}_s$.

We first run the MCMC algorithm in Section 4.1 without addressing the identifiability problem. After B burn-in iterations, G posterior samples of $\boldsymbol{\Phi}$, $\boldsymbol{\Phi}^{(1)}, ..., \boldsymbol{\Phi}^{(G)}$, are obtained. The word probability of a topic $k \in \boldsymbol{H}_j$ on any word v is estimated as $\frac{1}{G} \sum_{g=1}^{G} \phi_{k,v}^{(g)}$, and can be normalized to be

$$p_{k,v} = \frac{\frac{1}{G} \sum_{g=1}^{G} \phi_{k,v}^{(g)}}{\sum_{k' \in \boldsymbol{H}_j} \left(\frac{1}{G} \sum_{g=1}^{G} \phi_{k',v}^{(g)} \right)},$$

such that the sum over all topics in $\boldsymbol{H}_j$ is one. We then find the word with smallest entropy based on the normalized probabilities,

$$v_j = \text{argmin}_v \left[- \sum_{k \in \boldsymbol{H}_j} p_{k,v} \log p_{k,v} \right], \tag{13}$$

which can be treated as the most discriminative word for topics within $\boldsymbol{H}_j$.

We then order topics within $\boldsymbol{H}_j$ by their estimated word probabilities on v_j as:

$$p_{k_1^{(j)},v_j} > \cdots > p_{k_{h_j}^{(j)},v_j}, \tag{14}$$

where $\{k_1^{(j)}, ..., k_{h_j}^{(j)}\}$ forms a permutation of $\boldsymbol{H}_j$. In later MCMC iterations, in the step for sampling $\boldsymbol{\Phi}$, we will impose the constraint

$$\phi_{k_1^{(j)},v_j} > \cdots > \phi_{k_{h_j}^{(j)},v_j}. \tag{15}$$

Details are as follows.

For $k \in \boldsymbol{H}_j$ and $v = 1, ..., V$, let $\beta_{k,v} = \beta_v + n_{k,v}^{(1)} + n_{k,v}^{(1)\ mis}$. According to (9), in the step for sampling $\boldsymbol{\Phi}$, the marginal distribution of ϕ_{k,v_j} is $Beta(\beta_{k,v_j}, \sum_{v=1}^V \beta_{k,v} - \beta_{k,v_j})$, and the conditional distribution of the remaining elements of $\boldsymbol{\phi}_k$ given ϕ_{k,v_j} is

$$(1 - \phi_{k,v_j}) Dir(\beta_{k,1}, ..., \beta_{k,v_j-1}, \beta_{k,v_j+1}, ..., \beta_{k,V}). \tag{16}$$

We first sample ϕ_{k,v_j}, the word probabilities on the discriminative word v_j, for $k \in \boldsymbol{H}_j$ with the constraint (15). In particular, $\phi_{k_1^{(j)},v_j}$ is generated from $Beta(\beta_{k_1^{(j)},v_j}, \sum_{v=1}^V \beta_{k_1^{(j)},v} - \beta_{k_1^{(j)},v_j})$; then for $i = 2, ..., h_j$, $\phi_{k_i^{(j)},v_j}$ is generated from $Beta(\beta_{k_i^{(j)},v_j}, \sum_{v=1}^V \beta_{k_i^{(j)},v} - \beta_{k_i^{(j)},v_j})$ truncated over the interval $0 < \phi_{k_i^{(j)},v_j} < \phi_{k_{i-1}^{(j)},v_j}$. After sampling ϕ_{k,v_j}, probabilities on words other than v_j for $k \in \boldsymbol{H}_j$ are sampled using (16).

Appendix E: Prediction of class labels under CSTM

We first discuss how to use a single sample of $(\boldsymbol{\eta}, \boldsymbol{\Phi})$ to predict the class probabilities for a validation or test document d.

Conditional on $y_d = j$, if we can get estimates of the topic probabilities, $\hat{\theta}_{d,k}$ for $k \in \boldsymbol{\Lambda}_j$, then the likelihood of observing $\boldsymbol{w}_d$ can be estimated as

$$\hat{f}_j(\boldsymbol{w}_d) = \prod_{n=1}^{N_d} \left(\sum_{k \in \boldsymbol{\Lambda}_j} \hat{\theta}_{d,k} \phi_{k,w_{d,n}} \right). \tag{17}$$

According to the Bayes rule, the normalized value

$$\hat{q}_{d,j} = \frac{\eta_j \hat{f}_j(\boldsymbol{w}_d)}{\sum_{j'} \eta_{j'} \hat{f}_{j'}(\boldsymbol{w}_d)}$$

can be treated as the probability that document d belongs to class j.

We now discuss estimation of the topic probabilities given $y_d = j$ in detail. Treating the topic indicators $\boldsymbol{z}_d$ as missing data and treating $\boldsymbol{\eta}$ and $\boldsymbol{\Phi}$ as fixed, we can apply the EM algorithm (Dempster et al., 1977) to get $\hat{\theta}_{d,k}$ for $k \in \boldsymbol{\Lambda}_j$. The complete-data log-likelihood function is

$$\log f_j(\boldsymbol{w}_d, \boldsymbol{z}_d) = \sum_{n=1}^{N_d} \left(\sum_{k \in \boldsymbol{\Lambda}_j} I(z_{d,n} = k) \left(\log \theta_{d,k} + \log \phi_{k,w_{d,n}} \right) \right). \tag{18}$$

The EM algorithm starts with an initial set of values: $\theta_{d,k}^{[0]} = 1/(h_j + h_S)$ for $k \in \boldsymbol{\Lambda}_j$. Let $\theta_{d,k}^{[t]}$ be the estimate of $\theta_{d,k}$ at the t'th iteration, and iteration $t+1$ of EM proceeds as follows.

1. (E-step) Find the conditional expectation of $I(z_{d,n} = k)$ for $k \in \boldsymbol{\Lambda}_j$ given $\theta_{d,k}^{[t]}$ and the observed data. It can be easily derive that this is equal to

$$\Pr^{[t]}(z_{d,n} = k) = \frac{\theta_{d,k}^{[t]} \phi_{k,w_{d,n}}}{\sum_{k' \in \boldsymbol{\Lambda}_j} \theta_{d,k'}^{[t]} \phi_{k',w_{d,n}}}.$$

2. (M-step) Determine $\theta_{d,k}^{[t+1]}$ for $k \in \boldsymbol{\Lambda}_j$ by maximizing

$$\sum_{n=1}^{N_d} \sum_{k \in \boldsymbol{\Lambda}_j} \left\{ \Pr^{[t]}(z_{d,n} = k) \log \theta_{d,k} \right\}$$

subject to the constraint $\sum_{k \in \boldsymbol{\Lambda}_j} \theta_{d,k} = 1$. We can easily derive that

$$\theta_{d,k}^{[t+1]} = \sum_{n=1}^{N_d} \Pr^{[t]}(z_{d,n} = k).$$

In order to save computational time, we only run the EM algorithm for a few (R_1) iterations.

In the cross-validation procedure, when using a chain with R posterior samples to predict the class labels for the validation documents, we only use R_2 posterior samples of $(\boldsymbol{\eta}, \boldsymbol{\Phi})$ ($R_2 < R$), and assign document d to the class with maximum average value of $\hat{q}_{d,j}$.

After a candidate model is chosen, we have five selected chains (from the five-fold cross-validation), with R posterior samples in each chain. For an unlabeled training document d, we use the posterior samples of $\boldsymbol{y}^{mis}$ to assign it to the class with maximum value of $\sum_{c=1}^{5} \sum_{r=1}^{R} I(y_d^{(c,r)} = j)/(5R)$, where $y_d^{(c,r)}$ is the value of y_d in the rth sample of the cth chain. For a test document d, we use R_2 posterior samples of $(\boldsymbol{\eta}, \boldsymbol{\Phi})$ from each chain to assign it to the class with maximum value of $\sum_{c=1}^{5} \sum_{r=1}^{R_2} \hat{q}_{d,j}^{(c,r)}/(5R_2)$, where $\hat{q}_{d,j}^{(c,r)}$ is the calculated value of $\hat{q}_{d,j}$ for the rth sample of the cth chain.

We set $R_1 = 5$ and $R_2 = 5$. Experiments show that using larger value of R_1 or R_2 does not make much difference (results unreported).

Appendix F: Details of the modified CCS approach

In the modified CCS approach, we consider using words or phrases as the analysis units. Let $c_{d,v}$ denote the count that the vth term (word or phrase) in the dictionary appears in document d. Following Jia et al. (2014), these counts are L^2-normalized across labeled training documents to give features $x_{d,v} = c_{d,v}/\sqrt{\sum_{d'=1}^{D_1} c_{d',v}^2}$ $(d = 1, \cdots, D_1)$. Let $\boldsymbol{x}_d = (x_{d,1}, \cdots, x_{d,V})^\top$. To learn a sparse classifier, a separate L^1 penalized binary logistic regression is trained for each class using the fraction φ of labeled training documents:

$$\begin{aligned}(\hat{a}_j, \hat{\boldsymbol{b}}_j) = \underset{\boldsymbol{b}_j, a_j}{\operatorname{argmin}} \Big\{ &-\sum_{d=1}^{D_1} \log\Big(1 + \exp\Big((1 - 2I(y_d = j))\Big(a_j + \boldsymbol{b}_j^\top \boldsymbol{x}_d\Big)\Big)\Big) \\ &+\lambda_j \sum_{v=1}^{V} |b_{j,v}|\Big\},\end{aligned} \tag{19}$$

where a_j and $\boldsymbol{b}_j = (b_{j,1}, \cdots, b_{j,V})^\top$ $(j = 1, \cdots, J)$ are the intercept and coefficients of $\boldsymbol{x}_d$, and λ_j is a tuning parameter. Rather than selecting λ_j to achieve a desired prespecified number of selected features as in Jia et al. (2014), we follow Genkin et al. (2007) and Ifrim et al. (2008) to use ten-fold cross-validation to select λ_j that minimizes the misclassification rate. Features selected by at least one binary logistic regression are collected, and a multinomial logistic model for all classes is retrained on these features. The resulting model is used to predict class labels for unlabeled training or test documents.

We use the improved GLMNET algorithm (Yuan et al., 2012) to train penalized binary logistic regression, and use the limited memory BFGS (Liu and Nocedal, 1989) algorithm to train multinomial logistic regression. Both algorithms are implemented in the *scikit-learn* package in Python, and can handle the case when the number of features exceeds the number of observations, which is very common in text classification.

We also tried predicting with the class that has maximum predicted value over all binary logistic regressions, but less classification accuracy is achieved. Using Lasso rather than the L^1 penalized logistic regression, as suggested by Jia et al. (2014), also achieves less classification accuracy.

Appendix G: Monte Carlo simulations under the supervised scenario

As discussed in Section 5.4, the CSTM is able to capture subtle differences between classes. In this Section, we further use Monte Carlo simulations to help better understand in what circumstances the CSTM outperforms the competing approaches under the supervised scenario.

G.1 Data generation processes

We set the number of documents to be $D = 500$, the number of words in the dictionary to be $V = 1000$, and the number of topics to be $K = 10$. The number of words in each document d, N_d, follows a discrete uniform distribution from 40 to 60. We set up different variants of two types of data generation processes (DGPs): CSTM and DiscLDA. The details are as follows.

CSTM1. The word probabilities for each topic k ($k = 1, \cdots, K$) are independently generated from a Dirichlet distribution: $\boldsymbol{\phi}_k \sim Dir(\boldsymbol{\beta}_k)$, where $\boldsymbol{\beta}_k = (\beta_{k,1}, \cdots, \beta_{k,V})^\top$, with

$$\beta_{k,v} = \begin{cases} \tilde{\beta} & \text{if } v \in \{90(k-1)+1, \cdots, 90k\}, \\ 1 & \text{otherwise.} \end{cases}$$

If $\tilde{\beta} > 1$, for each topic, there is a set of 90 distinct words that on average have larger probabilities under the topic. The larger $\tilde{\beta}$ is, the larger distinction there is between the topics. If $\tilde{\beta} = 1$, then on average there is no distinction between the topics. However, because $\boldsymbol{\phi}_k$'s are generated from $Dir(\boldsymbol{\beta}_k)$ rather than being fixed to be $\boldsymbol{\beta}_k$, there will still be some distinction between the topics even when $\tilde{\beta} = 1$. Each of the last 100 words has the same average probability under any topic, and therefore, on average, these words are noise with no distinction between the topics.

We set the number of classes to be $J = 5$, the class-specific topics to be $\boldsymbol{H}_j = \{j\}$ ($j = 1, \cdots, 5$), and the shared topics to be $\boldsymbol{H}_S = \{6, 7, 8, 9, 10\}$. We generate the class label for each document, y_d, randomly from 1 to 5. For each document in class j ($j = 1, \cdots, 5$), we generate the value of $\boldsymbol{\theta}_d$ by drawing the probabilities over topics in $\boldsymbol{\Lambda}_j = (j, 6, 7, 8, 9, 10)$ from $Dir(\tilde{\alpha}, 1, 1, 1, 1, 1)$, and setting the probabilities over other topics to zero.

Given N_d, $\boldsymbol{\theta}_d$ and $\boldsymbol{\phi}_k$'s, the words in document d are generated as usual: for $n = 1, \cdots, N_d$, a topic is chosen from $z_{d,n} \sim Multi(\boldsymbol{\theta}_d)$, and a word is chosen from $w_{d,n} \sim Multi(\boldsymbol{\phi}_{z_{d,n}})$.

Note that the average probability for a class-specific topic in each document is $\tilde{\alpha}/(\tilde{\alpha}+5)$. The larger $\tilde{\alpha}$ is, the larger average probability of a class-specific topic there is in each document, and the higher the average classification accuracy can be. Also, the larger $\tilde{\beta}$ is, the larger distinction there is between the five class-specific topics, and the higher the average classification accuracy can be.

CSTM2. To reduce the distinction between the class-specific topics and therefore the distinction between classes, we allow the words with possibly larger average probabilities under different topics to have overlap. Specifically, we set

$$\beta_{1,v} = \begin{cases} \tilde{\beta} & \text{if } v \in \{1, \cdots, 100\}, \\ 1 & \text{otherwise;} \end{cases}$$

$$\beta_{2,v} = \begin{cases} \tilde{\beta} & \text{if } v \in \{51, \cdots, 150\}, \\ 1 & \text{otherwise;} \end{cases}$$

$$\beta_{3,v} = \begin{cases} \tilde{\beta} & \text{if } v \in \{151, \cdots, 250\}, \\ 1 & \text{otherwise;} \end{cases}$$

$$\beta_{4,v} = \begin{cases} \tilde{\beta} & \text{if } v \in \{201, \cdots, 300\}, \\ 1 & \text{otherwise;} \end{cases}$$

$$\beta_{k,v} = \begin{cases} \tilde{\beta} & \text{if } v \in \{100(k-5)+301, \cdots, 100(k-5)+400\}, \\ 1 & \text{otherwise} \end{cases} \qquad (k = 5, \cdots, 10).$$

There is overlap between the words with possibly larger average probabilities under topics 1 and 2, or under topics 3 and 4.

All other components of the DGP are the same as in CSTM1.

CSTM3. We consider a setting where the numbers of class-specific topics are different among classes. We set the number of classes to be $J = 3$, and the class-specific topics to be $\boldsymbol{H}_1 = \{1, 3\}$,

$\boldsymbol{H}_2 = \{2, 4\}$ and $\boldsymbol{H}_3 = \{5\}$. The shared topics are again $\boldsymbol{H}_S = \{6, 7, 8, 9, 10\}$. We generate the class label for each document randomly from 1 to 3. For each document in class 1 (or class 2), we generate the value of $\boldsymbol{\theta}_d$ by drawing the probabilities over topics in $\boldsymbol{\Lambda}_1 = (1, 3, 6, 7, 8, 9, 10)$ (or $\boldsymbol{\Lambda}_2 = (2, 4, 6, 7, 8, 9, 10)$) from $Dir(\tilde{\alpha}, \tilde{\alpha}, 1, 1, 1, 1, 1)$, and setting the probabilities over other topics to zero. For each document in class 3, we generate the value of $\boldsymbol{\theta}_d$ by drawing the probabilities over topics in $\boldsymbol{\Lambda}_3 = (5, 6, 7, 8, 9, 10)$ from $Dir(\tilde{\alpha}, 1, 1, 1, 1, 1)$, and setting the probabilities over other topics to zero.

All other components of the DGP are the same as in CSTM1. The relationship between the classification accuracy and the values of $\tilde{\alpha}$ and $\tilde{\psi}$ is similar to what is discussed for DiscLDA1.

CSTM4. We generate the topics' word probabilities ($\boldsymbol{\phi}_k$'s) as in CSTM2. All other components of the DGP are the same as in CSTM3.

DiscLDA1. In this DGP, we use the DiscLDA model described in Section 3.2. The topics' word probabilities, $\boldsymbol{\phi}_k$'s, are generated as in CSTM1. We set the number of classes to be $J = 5$, and the dimension for the underlying Drichlet variables to be $L = 6$. The class-specific linear transformation matrices $\boldsymbol{T}^j$'s are generated as follows. We first define a 10×6 class-specific matrix for each class j, $\boldsymbol{\psi}^j$, for which the element in row j and column 1 equals $\tilde{\psi}$ and the other elements equal 1. For $l = 1, \cdots, L$, the lth column of $\boldsymbol{T}^j$ is generated independently from a Dirichlet distribution with parameter given by the lth column of $\boldsymbol{\psi}^j$.

For each document, a 6-dimensional variable $\boldsymbol{\xi}_d$ is generated from $Dir(\tilde{\alpha}, 1, 1, 1, 1, 1)$, the class label y_d is randomly drawn from 1 to 5, and then the vector of topic probabilities, $\boldsymbol{\theta}_d$, is calculated as $\boldsymbol{\theta}_d = \boldsymbol{T}^{y_d}\boldsymbol{\xi}_d$. Given N_d, $\boldsymbol{\theta}_d$ and $\boldsymbol{\phi}_k$'s, the words in document d are generated as usual.

For a document in class j, the topic probability for topic k equals $T^j_{k,1}\xi_{d,1} + T^j_{k,2}\xi_{d,2} + \cdots + T^j_{k,6}\xi_{d,6}$. From the generation process for $\boldsymbol{T}^j$, we can derive that, for a given class j, the average value of $T^j_{k,1}$ equals $\tilde{\psi}/(\tilde{\psi} + 9)$ if $k = j$, and equals $1/(\tilde{\psi} + 9)$ otherwise, and that the average value of $T^j_{k,l}$ $(l = 2, \cdots, 6)$ equals $1/10$ for any k. Therefore, when $\tilde{\psi} > 1$, on average documents in class j have larger probabilities over topic j than over other topics. The larger $\tilde{\psi}$ is, the higher the average classification accuracy can be. When $\tilde{\psi} = 1$, on average each document is evenly distributed over different topics, and the classes cannot be well discriminated. However, because $\boldsymbol{T}^j$'s are randomly generated rather than being fixed to be $\boldsymbol{\psi}^j$, there will still be some class discrimination even when $\tilde{\psi} = 1$.

Note that the average value of $\xi_{d,1}$ equals $\tilde{\alpha}/(\tilde{\alpha} + 5)$ and the average value of $\xi_{d,l}$ $(l = 2, \cdots, 6)$ equals $1/(\tilde{\alpha} + 5)$. With given values of $\tilde{\beta}$ and $\tilde{\psi} > 1$, the larger $\tilde{\alpha}$ is, the larger average probability documents in class j have over topic j, and the higher the average classification accuracy can be. Also, with given values of $\tilde{\alpha}$ and $\tilde{\psi} > 1$, the larger $\tilde{\beta}$ is, the more distinction between the topics, and the higher the average classification accuracy can be.

DiscLDA2. We generate the topics' word probabilities as in CSTM2. All other components of the DGP are the same as in DiscLDA1.

DiscLDA3. Similar to CSTM3, we set the number of classes to be $J = 3$. We set the dimension for the underlying Drichlet variables to be $L = 7$. The class-specific linear transformation matrices $\boldsymbol{T}^j$ are generated as follows. For class 1, we define a 10×7 matrix $\boldsymbol{\psi}^1$, for which the element in row 1 and column 1 and the element in row 3 and column 2 both equal $\tilde{\psi}$, and the other elements equal

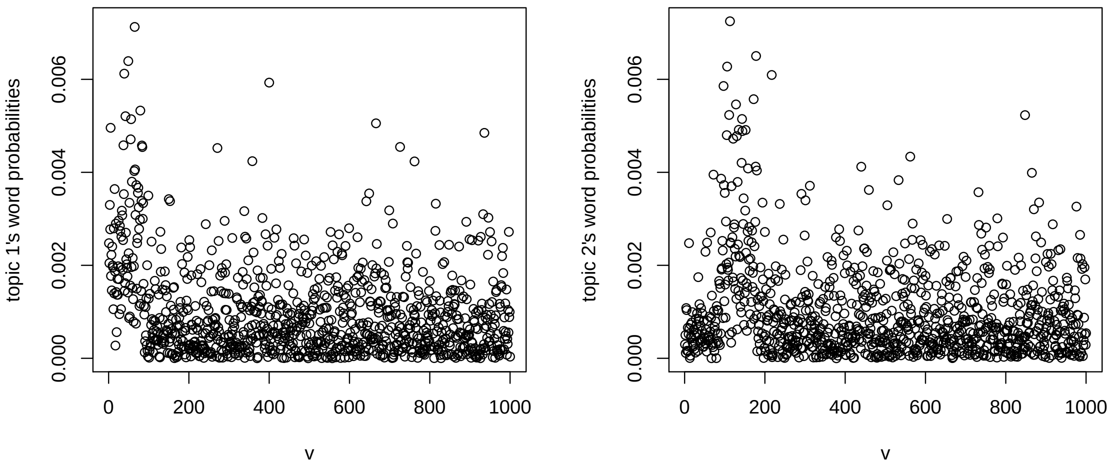

Figure 8: A set of generated values of $\boldsymbol{\phi}_1$ and $\boldsymbol{\phi}_2$ under CSTM1.

1. For class 2, we define a 10×7 matrix $\boldsymbol{\psi}^2$, for which the element in row 2 and column 1 and the element in row 4 and column 2 both equal $\tilde{\psi}$, and the other elements equal 1. For class 3, we define a 10×7 matrix $\boldsymbol{\psi}^3$, for which the element in row 5 and column 1 equals $\tilde{\psi}$, and the other elements equal 1.

For each document, a 7-dimensional variable $\boldsymbol{\xi}_d$ is generated from $Dir(\tilde{\alpha}, \tilde{\alpha}, 1, 1, 1, 1, 1)$, the class label y_d is randomly drawn from 1 to 3, and then the vector of topic probabilities, $\boldsymbol{\theta}_d$, is calculated as $\boldsymbol{\theta}_d = \boldsymbol{T}^{y_d}\boldsymbol{\xi}_d$. All other components of the DGP are the same as in DiscLDA1.

Similar to the discussion for DiscLDA1, we can easily derive that, when $\tilde{\psi} > 1$, on average documents in class 1 have larger probabilities over topics 1 and 3 than over other topics, on average documents in class 2 have larger probabilities over topics 2 and 4 than over other topics, and on average documents in class 3 have larger probabilities over topic 5 than over other topics. The relationship between the classification accuracy and the values of $\tilde{\alpha}$, $\tilde{\beta}$ and $\tilde{\psi}$ is similar to what is discussed for DiscLDA1.

DiscLDA4. We generate the topics' word probabilities as in CSTM2. All other components of the DGP are the same as in DiscLDA3.

G.2 Simulation results

We generate 100 datasets under each DGP. In each dataset, the documents are randomly splitted into a training set with 300 documents, and a test set with 200 documents.

We first set $\tilde{\alpha} = 5$ and $\tilde{\beta} = 3$, and consider $\tilde{\psi} = 20$ or $\tilde{\psi} = 1$. As an illustration, Figure 8 shows one set of generated values of $\boldsymbol{\phi}_1$ and $\boldsymbol{\phi}_2$ under CSTM1. We can see that there are differences in word probabilities for the two class-specific topics, but these differences are blurred by some noise. Therefore the distinction between classes is not clear-cut.

Table A1 presents the average classification accuracy of test documents for the supervised CSTM and its competitors under different DGPs. As expected, under the DiscLDA DGPs, the average classification accuracy is higher when $\tilde{\psi} = 20$ than when $\tilde{\psi} = 1$. Under all DGPs, CSTM outperforms all of its competitors. The average classification accuracy achieved by CSTM is higher than the highest average classification accuracy achieved by its competitors by 8.85% to 21.85%. Whether there is overlap between the words with larger average probabilities under different topics does not seem to matter much for the performance of the methods (e.g., the row of average classification accuracy when the DGP is CSTM1 is not much different from that when the DGP is CSTM2). Therefore, in future comparisons in this section, we only consider the case when $\boldsymbol{\phi}_k$'s are generated as in CSTM1.

Table A1: Average classification accuracy of test documents for the supervised CSTM and its competitors under different DGPs, with $\tilde{\alpha} = 5$ and $\tilde{\beta} = 3$.

DGP	CSTM	LDA+		LDA	Modified CCS		sLDA	DiscLDA	labeled
		RF	SVM	per-class	word	phrase			LDA
CSTM1	0.801	0.264	0.319	0.216	0.448	0.520	0.506	0.589	0.411
CSTM2	0.794	0.268	0.326	0.216	0.436	0.513	0.504	0.575	0.402
CSTM3	0.903	0.529	0.588	0.354	0.613	0.691	0.788	0.756	0.573
CSTM4	0.897	0.523	0.581	0.350	0.597	0.669	0.790	0.743	0.562
DiscLDA1 ($\tilde{\psi} = 20$)	0.561	0.211	0.249	0.216	0.298	0.358	0.323	0.362	0.279
DiscLDA2 ($\tilde{\psi} = 20$)	0.560	0.215	0.240	0.213	0.292	0.350	0.317	0.360	0.279
DiscLDA3 ($\tilde{\psi} = 20$)	0.532	0.340	0.355	0.343	0.369	0.413	0.400	0.411	0.374
DiscLDA4 ($\tilde{\psi} = 20$)	0.520	0.338	0.352	0.339	0.373	0.415	0.403	0.412	0.377
DiscLDA1 ($\tilde{\psi} = 1$)	0.473	0.314	0.322	0.310	0.342	0.378	0.219	0.372	0.346
DiscLDA2 ($\tilde{\psi} = 1$)	0.473	0.315	0.325	0.312	0.341	0.377	0.225	0.373	0.347
DiscLDA3 ($\tilde{\psi} = 1$)	0.495	0.337	0.348	0.337	0.368	0.406	0.355	0.397	0.370
DiscLDA4 ($\tilde{\psi} = 1$)	0.497	0.338	0.346	0.339	0.366	0.405	0.350	0.399	0.371

Table A2 presents the average classification accuracy of test documents when $\tilde{\alpha} = 1$ and $\tilde{\beta} = 1$. As expected, with smaller values for $\tilde{\alpha}$ and $\tilde{\beta}$, the classification accuracy is lower than that in Table A1. With $\tilde{\alpha} = 1$ and $\tilde{\beta} = 1$, the distinction between classes is very small, and the classification problem seems to be hard for any method. The advantage of the CSTM over its competitors is much less than that in Table A1. The average classification accuracy achieved by CSTM is higher than the highest average classification accuracy achieved by its competitors by 2.73% to 7.35%.

Table A2: Average classification accuracy of test documents for the supervised CSTM and its competitors under different DGPs, with $\tilde{\alpha} = 1$ and $\tilde{\beta} = 1$.

DGP	CSTM	LDA+		LDA	Modified CCS		sLDA	DiscLDA	labeled
		RF	SVM	per-class	word	phrase			LDA
CSTM1	0.294	0.201	0.206	0.207	0.206	0.231	0.218	0.223	0.219
CSTM3	0.476	0.338	0.341	0.335	0.359	0.403	0.368	0.401	0.362
DiscLDA1 ($\tilde{\psi} = 20$)	0.273	0.205	0.194	0.202	0.202	0.228	0.220	0.210	0.214
DiscLDA3 ($\tilde{\psi} = 20$)	0.380	0.342	0.337	0.331	0.336	0.350	0.349	0.346	0.347
DiscLDA1 ($\tilde{\psi} = 1$)	0.236	0.206	0.206	0.200	0.197	0.208	0.203	0.205	0.201
DiscLDA3 ($\tilde{\psi} = 1$)	0.381	0.343	0.332	0.336	0.335	0.354	0.343	0.354	0.345

Table A3 presents the average classification accuracy when $\tilde{\beta} = 3$, the same as for Table A1, and $\tilde{\alpha}$ is increased to 15. Table A4 presents the average classification accuracy when $\tilde{\alpha} = 5$, the same as for Table A1, and $\tilde{\beta}$ is increased to 10. As expected, the average classification accuracy in Tables A3 and A4 is higher than that in Table A1, except under the DiscLDA DGPs with $\tilde{\psi} = 1$. CSTM again outperforms all of its competitors. However, when there is very high distinction between classes and the classification problem is easy (e.g., when the DGP is CSTM3 in Table A3 or Table A4), the advantage of the CSTM over the best of its competitors is small.

Table A3: Average classification accuracy of test documents for the supervised CSTM and its competitors under different DGPs, with $\tilde{\alpha} = 15$ and $\tilde{\beta} = 3$.

DGP	CSTM	LDA+		LDA	Modified CCS		sLDA	DiscLDA	labeled
		RF	SVM	per-class	word	phrase			LDA
CSTM1	0.978	0.407	0.443	0.219	0.630	0.711	0.905	0.899	0.701
CSTM3	0.988	0.689	0.685	0.355	0.750	0.816	0.968	0.942	0.801
DiscLDA1 ($\tilde{\psi} = 20$)	0.775	0.256	0.318	0.221	0.395	0.477	0.501	0.522	0.352
DiscLDA3 ($\tilde{\psi} = 20$)	0.697	0.347	0.400	0.332	0.445	0.503	0.499	0.520	0.408
DiscLDA1 ($\tilde{\psi} = 1$)	0.322	0.210	0.201	0.199	0.215	0.237	0.234	0.243	0.224
DiscLDA3 ($\tilde{\psi} = 1$)	0.435	0.344	0.350	0.325	0.359	0.382	0.364	0.369	0.348

Table A4: Average classification accuracy of test documents for the supervised CSTM and its competitors under different DGPs, with $\tilde{\alpha} = 5$ and $\tilde{\beta} = 10$.

DGP	CSTM	LDA+		LDA	Modified CCS		sLDA	DiscLDA	labeled
		RF	SVM	per-class	word	phrase			LDA
CSTM1	0.967	0.356	0.391	0.236	0.742	0.820	0.880	0.946	0.897
CSTM3	0.988	0.665	0.678	0.386	0.836	0.891	0.980	0.979	0.935
DiscLDA1 ($\tilde{\psi} = 20$)	0.770	0.314	0.346	0.222	0.446	0.524	0.543	0.617	0.425
DiscLDA3 ($\tilde{\psi} = 20$)	0.757	0.399	0.478	0.339	0.495	0.550	0.577	0.576	0.426
DiscLDA1 ($\tilde{\psi} = 1$)	0.347	0.191	0.212	0.206	0.225	0.251	0.237	0.257	0.222
DiscLDA3 ($\tilde{\psi} = 1$)	0.493	0.325	0.358	0.335	0.362	0.384	0.385	0.393	0.366

To summarize, when viewing the data as generated from a topic model, the advantage of the CSTM over its competitors is largest when (1) differences between classes in topic probabilities are moderate, and (2) where topic probabilities do differ between classes, differences between topics in word probabilities are moderate. Otherwise, the advantage of the CSTM over its competitors is limited.

Appendix H: Candidate models under different settings of τ and $h_{S,max}$

We use $\varphi = 20\%$ as an example. Let τ vary from $\max(1/J, 0.3)$ to 0.8, with an increment of 0.05. Let $h_{S,\max}$ take values among $\{1, 2, 3, 4\}$ and then 5 to $\min(K^{\text{LDA}} - \sum_{j=1}^{J} h_j, 20)$, with an increment of 5. Table A5 shows the candidate values for h_j's and h_S obtained under each setting of τ and $h_{S,\max}$, as well as the accuracy of each candidate model on classifying unlabeled training documents and test documents.

The values of h_j's are determined by τ. With the increase of τ, h_j's become smaller and thus the candidate model becomes simpler. When τ reaches certain value that is large enough (e.g., 0.35

in our example), all h_j's equal to 1. The value of h_S is determined by $h_{S,\max}$, which is in turn determined by K^{LDA} and $\sum_{j=1}^{J} h_j$. In our example, given the same value of $h_{S,\max}$, as τ increases (or as the model becomes simpler), the classification accuracy increases; given the same value of τ, as $h_{S,\max}$ increases (or as the model becomes more complicated), the classification accuracy first increases and then decreases. The best classification accuracy is given by $\tau \geq 0.35$ and $h_{S,\max} = 5$.

Table A5: The Candidate Models and Their Accuracy under Different Settings for τ and $h_{S,\max}$.

Setting		Candidate Model		Classification Accuracy	
τ	$h_{S,\max}$	$h'_j s$	h_S	unlabeled train	test
0.3	1	$h_1 = h_2 = h_{18} = 2$, other h_j's=1	1	67.3%	66.52%
	2		2	67.81%	67.11%
	3		3	68.55%	69.04%
	4		4	70.19%	69.5%
	5		5	70.87%	70.81%
	10		10	69.33%	68.63%
	15		15	67.66%	67.02%
	20		20	65.51%	65.58%
0.35-0.8	1	all h_j's=1	1	72.19%	72.74%
	2		2	72.91%	73.17%
	3		3	73.84%	73.91%
	4		4	74.03%	74.82%
	5		5	74.5%	75.84%
	10		10	73.2%	74.91%
	15		15	70.14%	70.62%
	20		20	69.82%	68.73%

Appendix I: Performance of LDA and CSTM using bigrams or trigrams

Table A6 shows the classification accuracy of LDA and CSTM on unlabeled training documents and test documents, using bigrams or trigrams as the basis analysis units. Compared with Table 4 and 5, we see that the classification accuracy using bigrams or trigrams as the basic analysis units are generally slightly worse than that using words as the basic analysis units.

Table A6: Classification Accuracy of Unlabeled Training Documents and Test Documents Using Bigrams and Trigrams.

φ	Unlabeled Training Documents			Test Documents		
	CSTM	LDA+RF	LDA+SVM	CSTM	LDA+RF	LDA+SVM
10%	60.93%	61.36%	54.30%	62.41%	58.81%	51.27%
20%	73.95%	66.19%	57.61%	74.65%	62.15%	56.70%
30%	75.48%	67.32%	61.70%	75.44%	63.13%	58.75%
40%	78.69%	68.68%	63.40%	78.29%	64.80%	61.10%
50%	80.17%	70.33%	64.64%	80.51%	67.38%	63.01%
60%	80.32%	71.00%	67.20%	80.92%	68.05%	64.02%
70%	81.67%	71.37%	67.56%	82.50%	68.67%	65.56%
80%	82.26%	71.23%	68.47%	83.36%	69.23%	66.33%
90%	82.88%	72.32%	71.81%	83.60%	70.13%	68.04%
100%	–	–	–	84.36%	70.47%	68.70%

Appendix J: Text summaries for each class under LDA and the modified CCS approach

See Table A7 and Table A8.

Table A7: Top Ten Words with Highest Probabilities Under The Leading Topic for Each Class When $\varphi = 20\%$ (LDA)

Class	Top Ten Words with Highest Probabilities
comp.graphics	organization,lines,subject,university,thanks,anyone,please, know,email,would
comp.os.ms-windows.misc	dont,would,like,writes,get,one,re,think,lines,article
comp.sys.ibm.pc.hardware	card,SCSI,video,bit,Mac,monitor,memory,MHZ,MB,bus
comp.sys.mac.hardware	card,SCSI,video,bit,Mac,monitor,memory,MHZ,MB,bus
comp.windows.x	window,server,subject,motif,using,widget,use,application, set,problem
misc.forsale	sale,lines,organization,subject,price,new,offer,printer, shipping,sell
rec.autos	car,cars,engine,speed,oil,dealer,miles,new,drive,driving
rec.motorcycles	bike,DOD,ride,riding,motorcycle,bikes,BMW,dog,helmet,front
rec.sport.baseball	year,baseball,game,last,team,runs,games,players,win,hit
rec.sport.hockey	team,game,hockey,play,NHL,games,season,players,go,period
sci.crypt	key,encryption,chip,clipper,government,keys,security,use, system,technology
sci.electronics	radio,use,audio,radar,output,power,input,signal,detector,circuit
sci.med	gordon,banks,pain,doctor,medical,treatment,patients,soon, medicine,Candida
sci.space	space,NASA,launch,earth,moon,orbit,satellite,lunar,shuttle,first
talk.politics.guns	gun,guns,control,crime,weapons,firearms,police,criminals, handgun,public
talk.politics.mideast	Turkish,Armenian,Armenia,Turks,Turkey,people,genocide, world,Greek,war
talk.politics.misc	one,would,people,may,many,us,even,also,must,question
alt.atheism	one,would,people,may,many,us,even,also,must,question
soc.religion.christian	God,Jesus,Bible,church,Christian,Christ,Christians,faith, one,Gods
talk.religion.misc	one,would,people,may,many,us,even,also,must,question

Appendix K: An analysis of the Reuters dataset

Reuters-21578[3] is another benchmark dataset for text classification. The original dataset contains structured information about newswire articles. Each article is associated with multiple labels. The original collection contains 21,578 documents, including documents without labels or with typographical errors. A subset of the collection including only documents with labels and without typographical errors, usually referred to as "ModApte" split, is widely used (Erenel et al., 2011; Napoletano et al., 2012; Sciarrone, 2013; Al-Salemi et al., 2015). The "ModApte" split uses 9,603 articles before April 7, 1987 as the training set, and uses 3,299 articles after this date as the test set. A further step is to focus only on classes that have at least one document in both the training set and

[3] `http://kdd.ics.uci.edu/databases/reuters21578/reuters21578.html`

Table A8: Top Ten Entering Phrases for the Modified CCS When $\varphi = 20\%$

Class	Top Ten Entering Phrases
comp.graphics	Robert JC Kyanko,normals,Crowe,slices,VGA graphics mode, mpeg,Birmingham, orientations,graphics library,point polygon
comp.os.ms-windows.misc	plus Windows problems,WinBench,Steve Gibson,Windows operating, Windows, prodigy services,Speedisk,Windows operating system, WinBench results,re win
comp.sys.ibm.pc.hardware	Gordon Lang subject,Stephen Husak,gateway,controller,VESA localbus,hard drive, double warranty,card,controller,ns ns
comp.sys.mac.hardware	Centris,price drop,PowerBook,reconditioned,Apple,Gary Snow, internal hard, use VGA,Duo Dock,grab hold
comp.windows.x	Motif,pixmap,use Motif,Motif Window,building XR,flags,program produces, hints,program entry,teleuse UIMX
misc.forsale	custom resume,forsale,shipping,sale,answering machine,sale articleid, Radley, transferable,need sell,patwrscom
rec.autos	re top reasons,re integra,love cr humor,car may,subject re integra, top reasons, top reasons love,re dumbest automotive,car, cr humor impaired
rec.motorcycles	drinking riding,re speeding ticket,tools tools,Woodward,helmetless, protective, protective gear,riding,ticket CHP,backpack
rec.sport.baseball	Rockies,Su writes,designated hitter rule,baseball,opening day,new uniforms, young catchers,national league,Internet comes,home runs
rec.sport.hockey	Colons,cup organization,years biggest worst,worst opinion,biggest worst, first round pick,many Europeans NHL,years biggest,two thrown game, re many Europeans
sci.crypt	encryption scheme,clipper chip encryptionlist,key,hard drive security, chip encryption organization,scicrypt,Carl Ellison,encryption, drive security,security
sci.electronics	radarjust,radio car,radarjust work,help ultralong timing,Jacobs ladder, police radarjust,help ultralong,power line balls,subject re zero, police radarjust work
sci.med	jiggers,univ Pittsburgh,Gordon Banks,univ Pittsburgh computer, Dan Wallach, Pittsburgh computer science,Pittsburgh computer, aspirin,smokers lungs, organization univ Pittsburgh
sci.space	space faq,NASA,orbit,re space food,distribution sci,space, space station, space research spin,nuclear waste organization, re space research
talk.politics.guns	deer hunting,smuggle,make guns,hell TV news,weapon ban, Clements, Ray Clements,subject re gun,Manes writes, Aaron Ray Clements
talk.politics.mideast	Arabs,Serdar Argic,re Islam borders,Argic distribution,Islam borders, Serdar Argic distribution,Argic distribution world, Israelis,Panos,Argic
talk.politics.misc	deane subject,white males,someone may,David Matthew,Bill Riggs, top ten reasons,tax evasion,protect consumers, reasons aid Russians, insurance commissioner
alt.atheism	atheists would,Nanci Ann,re Americans evolution,Jon Livesey, atheism, women organization,Americans evolution,Livesey, Keith Allan,subject re political
soc.religion.christian	organization Kulikauskas,Kulikauskas,Christianity,rushing angels, Christ, Kulikauskas home,environmentalism,Kulikauskas subject, Jesus name,angels fear
talk.religion.misc	biblical contradictions organization,basis values,morality organization, re food,question popular morality,Joslin,lexicon, contradictions organization, question popular,re question pop

the test set. After this step, the dataset has 90 classes with a training set of 7,769 articles and a test set of 3,019 articles.

Because the CSTM can only deal with single-label classification problems, we further modify the dataset using the following steps. First, for each document, we only retain its first class label. After this, we find both the training set and the test set have uneven numbers of documents across the classes. Figure 9 shows the barplot of the number of training documents in each class. The top eight classes account for 87% of all documents. Therefore, we only focus on the top eight classes in the subsequent analysis. This leads to a final training set of 6,486 documents and a test set of 2,545 documents. Table A9 shows the retained eight classes and the numbers of training and test documents in each class.

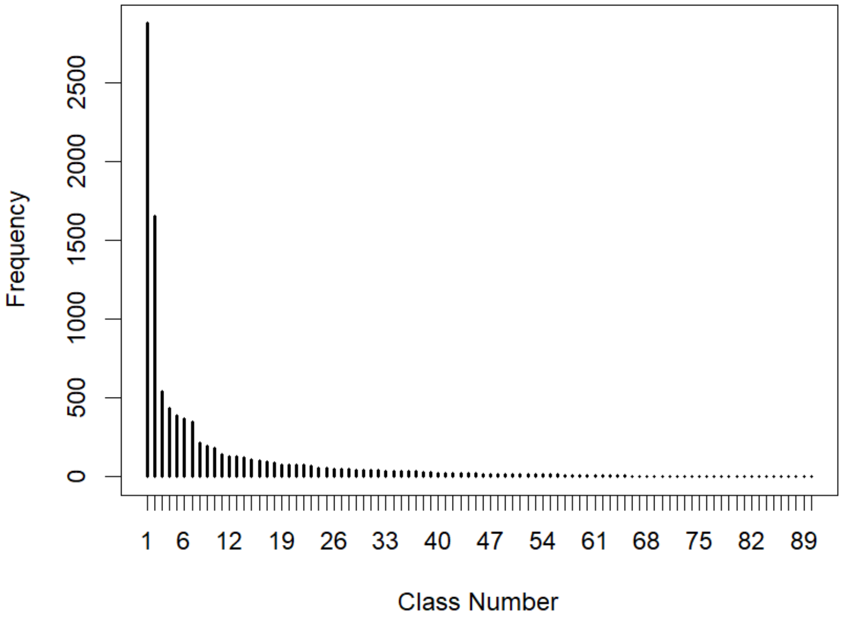

Figure 9: Barplot of the number of training documents in each class.

Table A9: Retained Classes and The Numbers of Training and Test Documents in Each Class

Class Number	Class Name	No. Training	No. Test
1	earn	2877	1087
2	acq	1634	716
3	money-fx	537	179
4	grain	430	148
5	crude	359	180
6	trade	323	105
7	interest	203	87
8	ship	123	43

We follow the common practices in text mining to remove numbers, punctuations and stopwords in Reuters-Truncated data, in the same way as what we did for 20 Newsgroup group dataset. After preprocessing, there are 59,773 unique words appearing in the training documents, which constitute the dictionary. We then compare the classification accuracy of test documents for the supervised

CSTM and its competitors. For each model, we follow similar steps discussed in Section 5 for model training, selection, and prediction. The selected CSTM model has $h_j = 3$ class-specific topics for each class, and $h_S = 20$ shared topics. The selected number of topics for LDA, sLDA, and DiscLDA are 81, 50, and 71, respectively. For labeled LDA, there are again one class-specific topic for each class and one common topic, resulting in a total of 9 topics. Table A10 displays the results. We find that CSTM slightly outperforms all the other methods.

Table A10: The Classification Accuracy of Test Documents for Supervised CSTM and Its Competitors in Reuters-Truncated Data

Method		Classification Accuracy
CSTM		94.18%
Two-stage	LDA+RF	89.86%
	LDA+SVM	90.33%
per-class LDA		5.81%
Modified CCS	word	91.00%
	phrase	91.51%
sLDA		92.30%
DiscLDA		86.40%
labeled LDA		20.19%

References

Bassam Al-Salemi, Mohd. Juzaiddin Ab Aziz, and Shahrul Azman Noah. LDA-Adaboost.MH: Accelerated Adaboost.MH based on latent Dirichlet allocation for text categorization. *Journal of Information Science*, 41(1):27–40, 2015.

L. Douglas Baker and Andrew Kachites McCallum. Distributional clustering of words for text classification. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 96–103, 1998.

Richard K. Belew. *Finding out about: a cognitive perspective on search engine technology and the WWW.* Cambridge University Press, 2000.

David M. Blei and Jon D. McAuliffe. Supervised topic models. In *Proceedings of the 20th International Conference on Neural Information Processing Systems*, pages 121–128, 2007.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent Dirichlet allocation. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14*, pages 601–608. MIT Press, 2002.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent Dirichlet allocation. *The Journal of Machine Learning Research*, 3(1):993–1022, 2003.

Andrei Z. Broder, Marcus Fontoura, Vanja Josifovski, and Lance Riedel. A semantic approach to contextual advertising. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 559–566, 2007.

Ziqiang Cao, Sujian Li, Yang Liu, Wenjie Li, and Heng Ji. A novel neural topic model and its supervised extension. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, pages 2210–2216, 2015.

Hau Chen and Susan Dumais. Bringing order to the web: Automatically categorizing search results. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems*, pages 145–152, 2000.

Hongshu Chen, Yi Zhang, and Donghua Zhu. Identifying technological topic changes in patent claims using topic modeling. In Daim T., Chiavetta D., Porter A., and Saritas O., editors, *Anticipating Future Innovation Pathways Through Large Data Analysis*, pages 187–209. 2016.

Gao Cong, Wee S. Lee, Haoran Wu, and Bing Liu. Semi-supervised text classification using partitioned EM. In Lee Y., Li J., Whang KY., and Lee D., editors, *Database Systems for Advanced Applications*, pages 482–493. 2004.

Arthur P. Dempster, Nan M. Laird, and Donald B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1–38, 1977.

Zafer Erenel, Hakan Altinay, and Ekrem Varolu. Explicit use of term occurrence probabilities for term weighting in text categorization. *Journal of Information Science & Engineering*, 27(3): 819–834, 2011.

Alexander Genkin, David D. Lewis, and David Madigan. Large-scale Bayesian logistic regression for text categorization. *Technometrics*, 49(3):291–304, 2007.

Juan C. Gomez and Marie-Francine Moens. PCA document reconstruction for email classification. *Computational Statistics and Data Analysis*, 56(3):741–751, 2012.

Thomas L. Griffiths and Mark Steyvers. Finding scientific topics. *Proceedings of the National Academy of Sciences of the United States of America*, 101(1):5228–5235, 2004.

Georgiana Ifrim, Gokhan Bakir, and Gerhard Weikum. Fast logistic regression for text categorization with variable-length n-grams. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 354–362, 2008.

Jinzhu Jia, Luke Miratrix, Bin Yu, Brian Gawalt, El Ghaoui Laurent, Luke Barnesmoore, Sophie Clavier, et al. Concise comparative summaries (CCS) of large text corpora with a human experiment. *The Annals of Applied Statistics*, 8(1):499–529, 2014.

Sergei Koltcov, Olessia Koltsova, and Sergey Nikolenko. Latent dirichlet allocation: stability and applications to studies of user-generated content. In *Proceedings of the 2014 ACM conference on Web science*, pages 161–165, 2014.

Simon Lacoste-Julien, Fei Sha, and Michael I. Jordan. DiscLDA: Discriminative learning for dimensionality reduction and classification. *Proceedings of Neural Information Processing Systems*, pages 897–904, 2008.

Ximing Li, Jihong Ouyang, and Xiaotang Zhou. Supervised topic models for multi-label classification. *Neurocomputing*, 149(part B):811–819, 2015.

Chenghua Lin, Yulan He, Carlos Pedrinaci, and John Domingue. Feature LDA: a supervised topic model for automatic detection of web api documentations from the web. In *Proceedings of the 11th International Semantic Web Conference*, pages 328–343, 2012.

Dong C. Liu and Jorge Nocedal. On the limited memory BFGS method for large scale optimization. *Mathematical Programming*, 45(1):503–528, 1989.

Jun S. Liu. The collapsed Gibbs sampler in Bayesian computations with applications to a gene regulation problem. *Journal of the American Statistical Association*, 89(427):958–966, 1994.

Youwei Lu, Shogo Okada, and Katsumi Nitta. Semi-supervised latent Dirichlet allocation for multi-label text classification. In *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*, pages 351–360, 2013.

Paolo Napoletano, Francesco Colace, Massimo De Santo, and Luca Greco. Text classification using a graph of terms. In *Proceedings of the 6th International Conference on Complex, Intelligent, and Software Intensive Systems*, pages 1030–1035, 2012.

Xuan-Hieu Phan and Cam-Tu Nguyen. GibbsLDA++: A C/C++ implementation of latent Dirichlet allocation (LDA), 2007. URL `http://gibbslda.sourceforge.net`.

Jipeng Qiang, Ping Chen, Tong Wang, and Xindong Wu. Topic modeling over short texts by incorporating word embeddings. In *Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 363–374, 2017.

Adrian E. Raftery, Michael A. Newton, Jaya M. Satagopan, and Pavel N. Krivitsky. Estimating the integrated likelihood via posterior simulation using the harmonic mean identity (with discussion). *Bayesian Statistics*, 8:1–45, 2007.

Daniel Ramage, David Hall, Ramesh Nallapati, and Christopher D Manning. Labeled LDA: A supervised topic model for credit attribution in multi-labeled corpora. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 248–256, 2009.

Philip Resnik, William Armstrong, Leonardo Claudino, Thang Nguyen, Viet-An Nguyen, and Jordan Boyd-Graber. Beyond LDA: exploring supervised topic modeling for depression-related language in Twitter. In *Proceedings of the 2nd Workshop on Computational Linguistics and Clinical Psychology: From Linguistic Signal to Clinical Reality*, pages 99–107, 2015.

Gerard Salton. *Automatic Text Processing: the Transformation, Analysis, and Retrieval of Information by Computer*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1989.

Filippo Sciarrone. An extension of the Q diversity metric for information processing in multiple classifier systems: A field evaluation. *International Journal of Wavelets Multiresolution and Information Processing*, 11(6), 2013. 1350049.

Jiang Su, Jelber S. Shirab, and Stan Matwin. Large scale text classification using semi-supervised multinomial naive Bayes. In *Proceedings of the 28th International Conference on Machine Learning*, pages 97–104, 2011.

Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, 58(1):267–288, 1996.

Di Wang, Marcus Thint, and Ahmad Al-Rubaie. Semi-supervised latent Dirichlet allocation and its application for document classification. In *Proceedings of the 2012 IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technology*, pages 306–310, 2012.

Sumio Watanabe. Asymptotic equivalence of Bayes cross validation and widely applicable information criterion in singular learning theory. *Journal of Machine Learning Research*, 11(1):3571–3594, 2010.

Yiming Yang and Jan O. Pedersen. A comparative study on feature selection in text categorization. In *Proceedings of the 14th International Conference on Machine Learning*, pages 412–420, 1997.

Guoxun Yuan, Chiahua Ho, and Chih-Jen Lin. An improved GLMNET for l1-regularized logistic regression. *The Journal of Machine Learning Research*, 13(1):1999–2030, 2012.

Yanning Zhang and Wei Wei. A jointly distributed semi-supervised topic model. *Neurocomputing*, 134:38–45, 2014.

Jun Zhu, Amr Ahmed, and Eric P. Xing. MedLDA: Maximum margin supervised topic models for regression and classification. In *Proceedings of the 26th International Conference on Machine Learning*, pages 14–18, 2009.

Arkaitz Zubiaga, Damiano Spina, Víctor Fresno, and Raquel Martínez. Classifying trending topics: a typology of conversation triggers on Twitter. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, pages 2461–2464, 2011.