Noname manuscript No.
(will be inserted by the editor)

Supervised Topic Models with Word Order Structure for Document Classification and Retrieval Learning

Shoaib Jameel · Wai Lam · Lidong Bing

15 August 2014 / 13 April 2015

Abstract One limitation of most existing probabilistic latent topic models for document classification is that the topic model itself does not consider useful side-information, namely, class labels of documents. Topic models, which in turn consider the side-information, popularly known as supervised topic models, do not consider the word order structure in documents. One of the motivations behind considering the word order structure is to capture the semantic fabric of the document. We investigate a low-dimensional latent topic model for document classification. Class label information and word order structure are integrated into a supervised topic model enabling a more effective interaction among such information for solving document classification. We derive a

The work described in this paper is substantially supported by grants from the Research Grant Council of the Hong Kong Special Administrative Region, China (Project Codes: 413510 and 14203414) and the Direct Grant of the Faculty of Engineering, CUHK (Project Code: 4055034). This work is also affiliated with the CUHK MoE-Microsoft Key Laboratory of Human-centric Computing and Interface Technologies. The authors would like to thank anonymous reviewers for their comments and suggestions.

Department of Systems Engineering and Engineering Management,
The Chinese University of Hong Kong,
Hong Kong.
Tel.: +852 97349180
E-mail: msjameel@se.cuhk.edu.hk

Department of Systems Engineering and Engineering Management,
The Chinese University of Hong Kong,
Hong Kong.
Tel.: +852 3943-8306
E-mail: wlam@se.cuhk.edu.hk

Machine Learning Department,
Carnegie Mellon University,
USA.
Tel.: +412 268-9338
E-mail: lbing@cs.cmu.edu

collapsed Gibbs sampler for our model. Likewise, supervised topic models with word order structure have not been explored in document retrieval learning. We propose a novel supervised topic model for document retrieval learning which can be regarded as a pointwise model for tackling the learning-to-rank task. Available relevance assessments and word order structure are integrated into the topic model itself. We conduct extensive experiments on several publicly available benchmark datasets, and show that our model improves upon the state-of-the-art models.

Keywords Topic Modeling · Maximum-Margin · Document Classification · Learning-to-Rank · Structured Topic Model

1 Introduction

Most existing probabilistic latent topic models such as Latent Dirichlet Allocation (`LDA`) [Blei et al., 2003], [Blei et al., 2001] are unsupervised probabilistic topic models which analyze a high dimensional term space and discover a low-dimensional topic space [Blei et al., 2003], [Steyvers and Griffiths, 2007], [Blei and Lafferty, 2009], [Blei, 2012]. They have been employed for tackling text mining problems [Sun et al., 2012] including document classification [Jameel and Lam, 2013b], [Rubin et al., 2012], [Li et al., 2015] and document retrieval [Wei and Croft, 2006], [Wang et al., 2007], [Chen, 2009], [Yi and Allan, 2009], [Egozi et al., 2011], [Andrzejewski and Buttler, 2011], [Wang et al., 2011], [Wang et al., 2013a], [Lu et al., 2011], [Yi and Allan, 2008], [Cao et al., 2007a], [Park and Ramamohanarao, 2009], [Duan et al., 2012]. These models can achieve better performance via detecting the latent topic structure and establishing a relationship between the latent topic and the goal of the problem. One limitation of unsupervised topic models for document classification is that the topic model itself does not consider the class labels of documents during inference. Various advantages of considering this variable in the latent topic models have been discussed in [Zhu et al., 2012a], and [Blei and McAuliffe, 2008]. Another limitation of latent topic models is that they do not exploit the word order structure of the documents. Some works attempt to integrate the class label information into a topic model for solving document classification, for example, supervised Latent Dirichlet Allocation (`sLDA`) [Blei and McAuliffe, 2008], multi-class supervised Latent Dirichlet Allocation (`mcLDA`) [Wang et al., 2009], supervised Hierarchical Dirichlet Processes Zhang et al. [2013], Storkey and Dai [2014], and maximum margin supervised topic model, `MedLDA`, [Zhu et al., 2012a]. These models have shown to improve document classification performance [Zhu et al., 2013a], [Jiang et al., 2012], [Zhu et al., 2014]. However, one common limitation of the above models is that they do not make use of the word order structure in text documents that could interact with the class label information for solving the document classification task. Obviously, technical challenges in considering the word order structure in a supervised topic model are high. First, the mathematical derivation of Gibbs sampling equations need to be revised from that of the unigram models as

our classification model considers distribution over bigrams. Such requirement involves refinement based on theoretical aspect. Bag-of-words models assume exchangeability in the probability space, whereas models which maintain the order of words in the document relax such a strong assumption [Aldous, 1985]. The form of input data to the model changes from the traditional word document co-occurrence matrix to full documents with word order.

Likewise, unsupervised topic models such as Topical N-Gram (`TNG`) [Wang et al., 2007], [Wang and McCallum, 2005] and Latent Dirichlet Allocation (`LDA`) have been used in developing document retrieval model [Wang et al., 2007], [Wei and Croft, 2006]. But they have not been explored for document retrieval learning which can be essentially cast into a learning-to-rank problem [Hang, 2011]. Learning-to-rank models make use of available relevance judgment information of a document for a query in the training process. The task is then to predict a desired ordering of documents. Several learning-to-rank models have been introduced, for example, [Wang et al., 2014], [Zong and Huang, 2014], [Yu et al., 2014], [Niu et al., 2014], but none of them considers the similarity between the document and the query under a low-dimensional topic space within the topic model itself.

The main idea in both of our models is to conduct posterior regularization [Ganchev et al., 2010], [Ganchev et al., 2010] in a Bayesian inference parameter learning setup [Zhu et al., 2014]. In posterior regularization using Bayesian inference, we intend to find a new desired posterior which is regularized using a regularization model. In our framework, our regularization is due to a maximum margin classifier which mainly helps predict the relevant class of the data. The notion is that for points which are difficult to classify by the classifier, the classifier gets an extra classifying signal from the topic model to help classify that point to its correct class. Such hard points are mainly located at the margin of the classifier or may be generally mis-classified by the classifier without any latent topic information. This posterior regularization mainly is a new posterior obtained by the topic model.

1.1 Our Main Contributions

We propose two topic models that build upon previous works on topic models with word order [Wallach, 2006], [Wallach, 2008], [Noji et al., 2013], [Jameel and Lam, 2013b], [Jameel and Lam, 2013c], [Kawamae, 2014], [Wang et al., 2007], etc which discuss in detail the challenges, motivation, and advantages of such models for solving various text mining tasks. One of the main advantages is that such models can better capture the semantic fabric of the document, which is lost when the order of words in the document is relaxed. In particular, our models incorporate the notion of side-information within the latent topic model itself. In contrast, none of the existing topic models with word order considers it. Side-information is mainly handled by the maximum margin classifier which is tightly integrated into the topic model. Topic models with word order have shown to produce more interpretable latent topics as compared to

unigram models [Wang et al., 2007], [Jameel and Lam, 2013b], [Jameel and Lam, 2013c], [Lindsey et al., 2012]. In addition, they have also shown to perform better on other quantitative tasks [Jameel and Lam, 2013b]. But such models fail to take advantage of side-information to produce more discriminative and interpretable latent topics. Our hybrid models can accomplish such goal. Our first model is a low-dimensional latent topic model for document classification. Class label information and word order structure are integrated into our supervised topic model with maximum margin learning enabling more effective interaction among such information for solving document classification. Mathematical derivation of Gibbs sampling equations are quite complex due the Markovian assumption on the order of the words for our model. Since our classification model considers the distribution over bigrams, the framework described in [Jiang et al., 2012], [Zhu et al., 2012a] needs considerable changes due to the exchangeability [Heath and Sudderth, 1976] assumption, [Aldous, 1985]. We adopt collapsed Gibbs sampler [Shao and Ibrahim, 2000] framework with considerable changes from [Jiang et al., 2012] because it collapses out the nuisance variables and speeds up the inference [Porteous et al., 2008]. The design and the study of the interplay between the side-information and word order is an interesting finding. Our model provides insights about how word order interacts with the side-information in a topic model. The implementation of the model is also challenging, where the input is not the word co-occurrence matrix, but a full document with word order.

Another contribution is that we propose a new supervised topic model for document retrieval learning which can be regarded as a pointwise model for tackling learning-to-rank task. Available relevance assessments and word order structure are integrated into the topic model itself. We jointly model the similarity between the query and the document under a low-dimensional topic space in a maximum margin framework. The main motivation for proposing this model is that in the document retrieval learning setting, our model apart from using the usual query-dependent features such as similarity metrics between the query and the document and query-independent features [Qin et al., 2010] such as PageRank [Brin and Page, 1998], can also use the topic similarity feature which can help find the similarity between the query and the document in the latent topic space. Fundamentally, even if the words between the query and the documents do not overlap, but their low-dimensional representations are semantically close or the same in their latent topic assignments, then we get a signal that they are describing about the same thematic content. We conduct extensive experiments on several publicly available benchmark datasets, and show that our model improves upon the state-of-the-art models. One major difference between our model and existing learning-to-rank models is that existing learning-to-rank models do not consider latent topic information in the learning framework. Our pointwise learning-to-rank model lays a foundation upon which future research on document retrieval learning can be done, for example, allowing further development of pairwise and listwise document retrieval learning probabilistic latent topic models. Note that we develop our model based on the design paradigm from [Jiang et al., 2012], [Zhu et al., 2012a]

for our document retrieval learning and classification models. An important point to note is that these methods have shown superior performance than the two-stage heuristic methods which first compute the latent topic vector representation and then these vectors are fed to another prediction model. In order to adapt the classification model for solving document retrieval learning problem, new design has to be made. First, the definition of the discriminant function needs to be designed to handle document retrieval learning task along with the other formulations that follow the discriminant function. Second, the relevance judgment associated with the query-document pair is also considered in our model. Third, the prediction task on unseen query and document pairs needs to be formulated as the prediction for the classification model will not directly work for document retrieval learning task.

1.2 Our Previous Works

Recently, in [Jameel and Lam, 2013b] we presented a topic model which is inspired from the Bigram Topic Model (`BTM`) [Wallach, 2006]. This model relaxes the bag-of-words assumption, and generates collocations just like the `LDA`-Collocation Model (`LDACOL`) [Griffiths et al., 2007]. It also differs from our new models proposed in this paper as we have incorporated side-information, where our previous model is unsupervised. Our temporal model proposed in [Jameel and Lam, 2013c], also generates more interpretable latent topics with word order. However, this model does not consider side-information and cannot solve document retrieval learning task. Our nonparametric topic model proposed in [Jameel and Lam, 2013a] significantly differs from the models proposed in this paper. Although our model maintains the order of words, and shows promising empirical performance, the model proposed in [Jameel and Lam, 2013a] does not incorporate side-information and it is a nonparametric topic model. Recently, we also proposed a nonparametric topic model where order of words is maintained [Jameel et al., 2015]. This model introduced a new non-exchangeable metaphor known as the Chinese Restaurant Franchise with Buddy Customers (CRF-BC). This model is significantly different from the models proposed in this work in that the CRF-BC model does not incorporate side-information. Also, the model is well suited for generated collocations and is nonparametric.

2 Related Work

Unsupervised and supervised topic models have been applied on the document classification task [Blei et al., 2003], [Blei and McAuliffe, 2008], [Wang et al., 2013b]. An advantage that supervised topic models have over unsupervised ones is that supervised topic models consider the available side-information as response variables in the topic model itself. This helps discover more predictive low dimensional representation of the data for better classification [Zhu

et al., 2012a]. Blei et al., proposed the Supervised Latent Dirichlet Allocation (sLDA) [Blei and McAuliffe, 2008] model which captures the real-valued document rating as a regression response. The model relies upon a maximum-likelihood based mechanism for parameter estimation. Wang et al., [Wang et al., 2009] proposed multi-class sLDA (mcLDA) which directly captures discrete labels of documents as a classification response. The Discriminative LDA (DiscLDA) [Lacoste-Julien et al., 2008] also performs classification in a different mechanism than sLDA. Different from the above models, Zhu et al., [Zhu et al., 2012a] proposed Maximum Entropy Discrimination LDA model known as MedLDA that directly minimizes a margin based loss derived from an expected prediction rule. The MedLDA model uses a variational inference method for parameter estimation. Subsequently, Markov Chain Monte Carlo techniques were proposed in [Zhu et al., 2013a], [Zhu et al., 2013c], [Jiang et al., 2012], [Zhu et al., 2013b]. In [Ramage et al., 2009], the authors proposed a supervised topic model which jointly models available class labels and text content by defining a one-to-one correspondence between latent topics and class label information. This allows their model to directly learn word-tag correspondences in the topic model itself. What has not been studied in supervised topic modeling is the role that the word order structure in the text content that could play along with the side-information in the document classification task. Our proposed supervised topic model falls in the class of parametric topic models where the number of latent topics has to supplied by the user, but recently, Kawamae [Kawamae, 2014] presented a nonparametric supervised n-gram topic model based on a Pitman-Yor process prior [Pitman and Yor, 1997] for phrase extraction which takes the advantage of labels during training process. However, it cannot perform document retrieval learning as in our model. Moreover, in [Bartlett et al., 2010], it has been stated that nonparametric models with Pitman-Yor process priors cannot scale to large scale datasets. There are other proposed supervised nonparametric topic modeling approaches such as [Perotte et al., 2011], [Storkey and Dai, 2014], [Lakshminarayanan and Raich, 2011], [Xie and Passonneau, 2012], [Liao et al., 2014], [Acharya et al., 2013]. These models too cannot perform document retrieval learning task. In addition, such nonparametric topic models are computationally very expensive [Wallach et al., 2009].

Unsupervised topic models have also been used to perform document classification. As mentioned above, they do not make use of the available side-information in the topic model itself. The LDA model is one example and it achieves better performance than that of Support Vector Machines (SVM) [Joachims, 1998], [Cortes and Vapnik, 1995], [Vapnik, 2000]. In [Rubin et al., 2012], the authors showed a model that maintains the order of words in documents which helps achieve better classification results. In [Li and McCallum, 2006], the authors presented an unsupervised hierarchical topic model which generates super and sub-topics. The authors showed good classification performance than the comparative methods. The model is represented by a Directed Acyclic graph, which has a capability to capture correlations between two levels of topics. In fact, topic models have also been used on other datasets apart

from text documents for classification under the unsupervised setting [Bicego et al., 2010], [Pinoli et al., 2014].

It has been studied in the past that considering the order of words in documents helps improve both quantitative and qualitative performance of probabilistic topic models. For example, Wallach [Wallach, 2008] has studied that word order is an important component in many applications such as natural language processing, speech recognition, text compression, etc. Therefore, bag-of-words models might not be very suitable for such applications. Wallach proposed the Bigram Topic model (`BTM`) which is an extension to the `LDA` model. The `BTM` adopts a Markovian assumption on the order of words in documents, and has shown to perform better than the `LDA` model in predictive tasks. But the `BTM` had limitation in that it only generates bigram words, which may not be desirable for some tasks. Griffiths et at., [Griffiths et al., 2007] proposed the `LDA` collocation model (`LDACOL`) which can generate either unigram or bigram words based on the context information. But in `LDACOL` model, only the first term has a topic assignment whereas the second term does not, which was addressed in the topical n-gram model (`TNG`) [Wang and McCallum, 2005], [Wang et al., 2007]. Some improvements to the `BTM` have been proposed in [Noji et al., 2013]. In all these works it has been suggested that word order plays important role in topic models. In terms of qualitative results, words appear more interpretable [Lindsey et al., 2012], and in terms of quantitative results it has been shown to improve many applications such as document classification [Jameel and Lam, 2013b], information retrieval [Wang et al., 2007], etc.

Learning-to-rank models have been extensively investigated and they can be categorized into pointwise, pairwise, and listwise approaches [Liu, 2009]. One early work used some bag-of-features in training a `SVM` model in order to conduct document retrieval learning which can be regarded as a pointwise approach for the learning-to-rank task [Nallapati, 2004]. This approach predicts a binary relevance prediction. Documents are then ranked based on the confidence scores given by the discriminative classifier. Subsequently other discriminative learning-to-rank models have been proposed such as those which handle multi-class relevance assessments [Busa-Fekete et al., 2013], [Li et al., 2007]. Many state-of-the-art learning-to-rank models have been proposed recently. For example, Gao et. al [Gao and Yang, 2014] recently presented a listwise learning-to-rank model, a novel semi-supervised rank learning model which is extended to an adaptive ranker to domains where no training data is available. In [Lai et al., 2013], the authors presented a sparse learning-to-rank model for information retrieval. Dang et al., [Dang et al., 2013] proposed a two-stage learning-to-rank framework to address the problem of sub-optimal ranking when many relevant documents are excluded from the ranking list using bag-of-words retrieval models. In Tan et al. [2013] the authors proposed a model which directly optimizes the ranking measure without resorting to any upper bounds or approximations. However, a major difference between these learning-to-rank models and our proposed document retrieval learning

model is that our model considers the latent topic information unified within a discriminative framework.

In the past, few proposals have been made to conduct document retrieval where the low-dimensional latent semantic space has been used. In [Li and Xu, 2014] the authors summarize many of those works. The main motivation for incorporating the semantic information in document retrieval task is mainly to compute the similarity between the latent factors which is based on the semantic content of the document. In [Bai et al., 2010], the authors proposed a discriminative model called supervised semantic indexing which can be trained on labeled data. Their model can compute query-document and document-document similarity in the semantic space. Their focus is primarily on traditional document retrieval than learning-to-rank using an extensive set of feature values. Gao et al., in [Gao et al., 2011], and Jagarlamudi et al., in [Jagarlamudi and Gao, 2013] proposed topic models which jointly consider the query and the title of the document to conduct document retrieval task using a language modeling framework. Their motivation for considering title fields in the documents is mainly because queries [Broder, 2002] as well as titles are mostly short in nature, thus short document titles could represent more informative power than the entire document for a query. One difference between our model and their framework is that their model is not designed to solve the learning-to-rank task considering feature instances. Our model jointly learns the query and document pair along with the associated relevance label in the latent topic space.

Our document retrieval learning framework is also closely related to some works in posterior regularization. The objective of the posterior regularization framework is to restrict the space of the model parameters on unlabeled data as a way to guide the model towards some desired behaviour. In [Ganchev et al., 2010], the authors proposed a framework which incorporates side-information into the parameter estimation in the form of linear constraints on posterior expectations. Recently, Zhu et al., [Zhu et al., 2014], [Zhu et al., 2012b] introduced Bayesian posterior regularization under an information theoretic formulation, and applied their framework on infinite latent `SVM`. Earlier, the same authors had extended the Zellner's view of the optimization framework described in [Zellner, 1988] to propose a regularized Bayesian regularization framework for multi-task learning problem [Zhu et al., 2011]. The authors mainly added a convex function to the optimization framework proposed by Zellner. Models such as `MedLDA` [Zhu et al., 2012a], [Zhu et al., 2009] and some of its extension are based on such frameworks [Zhu et al., 2013a], [Jiang et al., 2012].

Relational topic models, such as the one described in [Chang and Blei, 2009], incorporate side-information in the form of connections on information networks. Such connections can be social network friends as used in [Yuan et al., 2013] or scholar citation networks. In [Tang et al., 2011] the authors proposed a topic model with supervised information for advertising. These models are not designed to handle document retrieval learning which can be cast as a learning-to-rank problem. Also, in our model we incorporate the

latent topic model from the BTM model to better capture latent semantic information. The supervising signal is used in the maximum margin framework.

3 Background

We first present a brief background in this section that would help understand our proposed models described later. We start with a basic topic model known as Latent Dirichlet Allocation (LDA) [Blei et al., 2003]. We present the details of main part of the LDA model. Then we will present the optimization framework of the posterior distribution obtained from LDA. This optimization framework will be then extended to incorporate loss functions from maximum-margin classifier. We will present an example of a supervised topic model that makes use of the optimization framework of LDA by extending it to incorporate some posterior constraints in Bayesian inference leading to what is known as regularized Bayesian inference framework.

3.1 Latent Dirichlet Allocation (LDA)

Latent Dirichlet Allocation (LDA) is a generative probabilistic topic model for collections of discrete data such as text document collections. The model assumes that documents exhibit multiple latent topics. Therefore, each document is a mixture of a number of topics. In LDA, it represents a latent topic as a probability distribution of words taken from a vocabulary set. A document is denoted by $d \in \{1, \ldots, D\}$ where D is the total number of documents in the collection. Let $\boldsymbol{W} = \{\boldsymbol{w}^d\}_{d=1}^{D}$ denote all the words in all the documents in the collection where each $\boldsymbol{w}^d$ denotes the words in the document d. N^d is the number of words in the document d. w_n^d is the word at the position n in the document d. K is the total number of latent topics as specified by the user. z_n^d is the topic assignment of the word w_n^d. $\boldsymbol{Z} = \{\boldsymbol{z}^d\}_{d=1}^{D}$ are topic assignments to all the words. $\boldsymbol{\Theta} = \{\boldsymbol{\theta}^d\}_{d=1}^{D}$ are topic distributions for all documents. Let $\boldsymbol{\Phi} = \{\boldsymbol{\phi}_k\}_{k=1}^{K}$ denote the word-topic distribution. Let V denote the number of words in the vocabulary. Let $\boldsymbol{\alpha}$ be the vector denoting the hyperparameter values for the document-topic distributions. Let $\boldsymbol{\beta}$ denote the vector of hyperparameter values for the word-topic distributions.

The LDA model describes the generative procedure of each document in the collection. Each document is generated from a mixture of topics that pervades the document. Each of those topics is in turn responsible for generating the words without giving importance to the order of the occurrence of the words in those documents.

The generative process of the LDA model is written as:

1. Draw topic proportion for each document d denoted as $\boldsymbol{\theta}^d$ from **Dirichlet**($\boldsymbol{\alpha}$), $\boldsymbol{\theta}^d$ is the topic proportions for a document,
2. Draw ϕ_k for each topic k from **Dirichlet**($\boldsymbol{\beta}$),
3. For each word w_n^d in the document d,

(a) Draw a topic assignment $z_n^d|\boldsymbol{\theta}^d$ from **Multinomial**$(\boldsymbol{\theta}^d)$
(b) Draw the observed word $w_n^d|z_n^d, \boldsymbol{\Phi}$ from **Multinomial**$(\phi_{z_n^d})$

The probability of a document collection $\mathbb{D}$ in `LDA` is given as:

$$p(\mathbb{D}|\boldsymbol{\alpha}, \boldsymbol{\beta}) = \prod_{d=1}^{D} \int P(\boldsymbol{\theta}^d|\boldsymbol{\alpha}) \left(\prod_{n=1}^{N^d} \sum_{z_n^d} P(z_n^d|\boldsymbol{\theta}^d) P(w_n^d|z_n^d, \boldsymbol{\beta}) \right) d\boldsymbol{\theta}^d \quad (1)$$

The posterior distribution inferred by the `LDA` model can be written as:

$$P(\boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi}|\boldsymbol{W}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \frac{P_0(\boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi}|\boldsymbol{\alpha}, \boldsymbol{\beta}) P(\boldsymbol{W}|\boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi})}{P(\boldsymbol{W}|\boldsymbol{\alpha}, \boldsymbol{\beta})} \quad (2)$$

where $P(\boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi}|\boldsymbol{W}, \boldsymbol{\alpha}, \boldsymbol{\beta})$ is the posterior distribution of the model. Let the prior distribution represented as $P_0(\boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi}|\boldsymbol{\alpha}, \boldsymbol{\beta})$, and it is defined as:

$$P_0(\boldsymbol{\Theta}, \boldsymbol{\Phi}, \boldsymbol{Z}|\boldsymbol{\alpha}, \boldsymbol{\beta}) = \left(\prod_{d=1}^{D} P(\boldsymbol{\theta}^d|\boldsymbol{\alpha}) \prod_{n=1}^{N^d} P(z_n^d|\boldsymbol{\theta}^d) \right) \prod_{k=1}^{K} P(\boldsymbol{\phi}_k|\boldsymbol{\beta}) \quad (3)$$

$P(\boldsymbol{W}|\boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi})$ is the likelihood. $P(\boldsymbol{W}|\boldsymbol{\alpha}, \boldsymbol{\beta})$ is the marginal probability distribution.

3.2 Learning Using Bayesian Inference

Equation 2 presented in Section 3.1 can be further translated into an information theoretical optimization problem [Jiang et al., 2012], [Zhu et al., 2012a], [Zhu et al., 2013a], [Zhu et al., 2014]. An advantage of considering this paradigm is that it can be easily extended to incorporate some regularization terms on the desired posterior distribution obtained using Bayes' theorem. It can lead to a learning model where the posterior distribution obtained using the Bayes' theorem is directly regularized using a learning model which considers side-information. The regularizer can be obtained from the maximum-margin learning principle, and then can be integrated into the Bayesian learning paradigm leading to regularized Bayesian inference using maximum-margin learning. In principle, this hybrid model could achieve better prediction performance than using a topic model or a maximum-margin classifier alone because this hybrid model inherits the prediction power from both maximum margin prediction learning and topic models. It is well known that maximum margin classifiers have shown strong generalization performance [Burges, 1998], and topic models have also shown good performance on document classification task [Rubin et al., 2012], [Li and McCallum, 2006]. Therefore, we can expect that the hybrid model can inherit advantages of both of these models. When conducting posterior inference, we can directly regularize the posterior distribution, which leads to a new posterior regularized by a constraint.

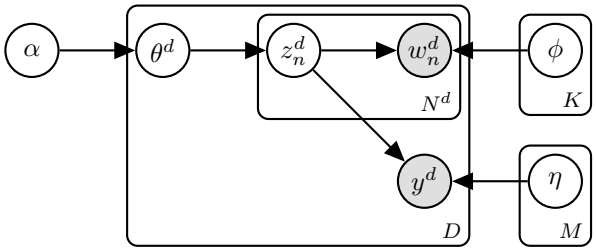

Fig. 1 Graphical representation of the `MedLDA` model.

Some supervised topic models such as `MedLDA` [Zhu et al., 2012a], Monte Carlo `MedLDA` [Jiang et al., 2012], etc. are based on this paradigm.

According to the findings described in [Zellner, 1988], Equation 2 can be transformed to an optimization problem which can be written as follows:

$$\begin{aligned} \underset{P(\boldsymbol{\Theta},\boldsymbol{Z},\boldsymbol{\Phi})\in\mathbb{P}}{\text{minimize}} \quad & \mathrm{KL}[P(\boldsymbol{\Theta},\boldsymbol{Z},\boldsymbol{\Phi}|\boldsymbol{W},\boldsymbol{\alpha},\boldsymbol{\beta})||P_0(\boldsymbol{\Theta},\boldsymbol{Z},\boldsymbol{\Phi}|\boldsymbol{\alpha},\boldsymbol{\beta})] - \mathbb{E}_P[\log \mathrm{P}(\boldsymbol{W}|\boldsymbol{Z},\boldsymbol{\Phi})] \\ \text{subject to} \quad & P(\boldsymbol{\Theta},\boldsymbol{Z},\boldsymbol{\Phi}) \in \mathbb{P}, \end{aligned} \tag{4}$$

where $\mathbb{P}$ is the probability distribution space, and $\mathrm{KL}(P||P_0)$ is the Kullback-Leibler divergence from P to P_0. The above optimization interpretation will be useful in our later discussion where we will show how this technique can be used to derive a new maximum margin learning framework using a topic model. We present how the posterior distribution can be transformed into an optimization problem depicted in Equation 4 in Appendix A.

3.3 Maximum Margin Entropy Discrimination - LDA (MedLDA)

As mentioned above, our proposed model can be regarded as a supervised topic model where the class label information is incorporated into a topic model itself. Supervised topic models have been used for both classification and regression tasks. One example of a supervised topic model is supervised `LDA` (`sLDA`) [Blei and McAuliffe, 2008] which is based on extending `LDA` via the likelihood principle. Another recent supervised topic model is `MedLDA` [Zhu et al., 2012a], [Zhu et al., 2009], [Jiang et al., 2012] whose graphical model is presented in Figure 1. Note that in this model, β is not used explicitly, but can be used as a prior to make the model fully Bayesian [Zhu et al., 2012a]. `MedLDA` combines a maximum margin learning algorithm based on Support Vector Machines (`SVM`) for label prediction, and a topic model based on `LDA` for the semantic content of the words.

The class label for the document d is denoted by y^d which takes on one of the values $\mathtt{Y} = \{1, \ldots, M\}$. Let $\overline{\boldsymbol{z}}^d$ denote a K dimensional vector with each element $\overline{z}_k^d = \frac{1}{N^d}\sum_{n=1}^{N^d} \mathbb{I}(z_n^d = k)$. $\mathbb{I}(.)$ is an indicator function which equals to 1 if the predicate holds else it is 0. $\boldsymbol{f}(y, \overline{\boldsymbol{z}}^d)$ is a MK-dimensional

vector whose elements from $(y-1)$K to yK are $\overline{\boldsymbol{z}}^d$ and the rest are all 0. Let $\boldsymbol{\eta}$ denote the parameters of the maximum margin classification model. Let C be a regularization constant, ξ^d be the slack variable, and $l^d(y)$ be the loss function for the label y; all of which are positive. $\boldsymbol{\xi}$ are the nonnegative auxiliary parameters and are usually referred to as the slack variables. Consider the Zellner's interpretation shown in Equation 4. In a regularized Bayesian framework setting a convex function is added to the optimization framework described above [Zhu et al., 2011]. One choice of such convex function is to borrow ideas from a maximum margin classifier model, and this equation can be written as:

$$\begin{aligned} \underset{P(\boldsymbol{\eta},\boldsymbol{\Theta},\boldsymbol{Z},\boldsymbol{\Phi})\in\mathbb{P},\boldsymbol{\xi}}{\text{minimize}} \quad & \mathrm{KL}[P(\boldsymbol{\Theta},\boldsymbol{Z},\boldsymbol{\Phi}|\boldsymbol{W},\boldsymbol{\alpha},\boldsymbol{\beta})||P_0(\boldsymbol{\Theta},\boldsymbol{Z},\boldsymbol{\Phi}|\boldsymbol{\alpha},\boldsymbol{\beta})] - \mathbb{E}_P[\log \mathrm{P}(\boldsymbol{W}|\boldsymbol{Z},\boldsymbol{\Phi})] + B(\boldsymbol{\xi}) \\ \text{subject to} \quad & P(\boldsymbol{\eta},\boldsymbol{\Theta},\boldsymbol{Z},\boldsymbol{\Phi}) \in \mathbb{P}(\boldsymbol{\xi}), \end{aligned} \tag{5}$$

where $B(\boldsymbol{\xi})$ is a convex function which usually refers to the hinge loss function of the maximum margin classifier. $\boldsymbol{\eta}$ denotes the parameters of the maximum margin classifier. $\mathbb{P}(\boldsymbol{\xi})$ is the subspace of probability distribution that satisfies a set of constraints. One can note that as stated in Section 3.2, we can add a loss function to the optimization view of the Bayes' theorem obtained from `LDA`. Thus the interpretation given by Zellner, can be easily used to develop supervised topic models for prediction tasks.

Considering a maximum margin based topic model for label prediction, `MedLDA`, the soft-margin for `MedLDA` can be written as:

$$\begin{aligned} \underset{p(\boldsymbol{\eta},\boldsymbol{\Theta},\boldsymbol{Z},\boldsymbol{\Phi})\in\mathbb{P},\xi}{\text{minimize}} \quad & \mathrm{KL}[P(\boldsymbol{\Theta},\boldsymbol{Z},\boldsymbol{\Phi}|\boldsymbol{W},\boldsymbol{\alpha},\boldsymbol{\beta})||P_0(\boldsymbol{\Theta},\boldsymbol{Z},\boldsymbol{\Phi}|\boldsymbol{\alpha},\boldsymbol{\beta})] - \mathbb{E}_P[\log \mathrm{P}(\boldsymbol{W}|\boldsymbol{Z},\boldsymbol{\Phi})] + \frac{C}{D}\sum_{d=1}^{D}\xi^d \\ \text{subject to} \quad & \mathbb{E}_p[\boldsymbol{\eta}^\intercal \boldsymbol{f}(y^d,\overline{\boldsymbol{z}}^d) - \boldsymbol{f}(y,\overline{\boldsymbol{z}}^d)] \geq l^d(y), \xi^d \geq 0, \forall d, \forall y, \end{aligned} \tag{6}$$

One can see from the above equation that `MedLDA` conducts regularized Bayesian inference which is of the same form as depicted in Equation 5. Therefore, `MedLDA` is a hybrid topic model which takes advantages from topic model and maximum margin learning framework. Equation 6 can also be written as:

$$\begin{aligned} \underset{P(\boldsymbol{\eta},\boldsymbol{\Theta},\boldsymbol{Z},\boldsymbol{\Phi})\in\mathbb{P},\xi}{\text{minimize}} \quad & \mathrm{KL}[P(\boldsymbol{\Theta},\boldsymbol{Z},\boldsymbol{\Phi}|\boldsymbol{W},\boldsymbol{\alpha},\boldsymbol{\beta})||P_0(\boldsymbol{\Theta},\boldsymbol{Z},\boldsymbol{\Phi}|\boldsymbol{\alpha},\boldsymbol{\beta})] - \mathbb{E}_P[\log \mathrm{P}(\boldsymbol{W}|\boldsymbol{Z},\boldsymbol{\Phi})] + \\ & \frac{C}{D}\sum_d \mathrm{argmax}_y(l^d(y)) - \mathbb{E}_P[\boldsymbol{\eta}^\intercal(\boldsymbol{f}(y^d,\overline{\boldsymbol{z}}^d) - \boldsymbol{f}(y,\overline{\boldsymbol{z}}^d))] \end{aligned}$$

The component $\frac{1}{D}\sum_d \mathrm{argmax}_y(l^d(y)) - \mathbb{E}_p[\boldsymbol{\eta}^\intercal(\boldsymbol{f}(y^d,\overline{\boldsymbol{z}}^d) - \boldsymbol{f}(y,\overline{\boldsymbol{z}}^d))]$ is the hinge loss which is defined as an upper bound of the prediction error on the training data.

One characteristic of `MedLDA` is to conduct posterior regularization where the posterior distribution obtained using a topic model is regularized with maximum margin constraints. This leads to a posterior which is mainly helpful

in classifying those points which lie on the margin of the classifier or are mis-classified. The latent topic information supplied by the topic model helps classify such hard instances, for which the maximum margin classifier would find it difficult to accomplish. This mechanism makes this model different from those two stage approaches where one can compute the latent topic information using a topic model, and then use that latent topic information as an added feature in the classification task. Two stage approach for prediction might involve error propagation from one stage to another, which can be mitigated in such single stage models as `MedLDA`.

4 Supervised Topic Model with Word Order for Document Classification

4.1 Model Description

We propose a document classification model based on a latent topic model that integrates the class label information and the word order structure into the

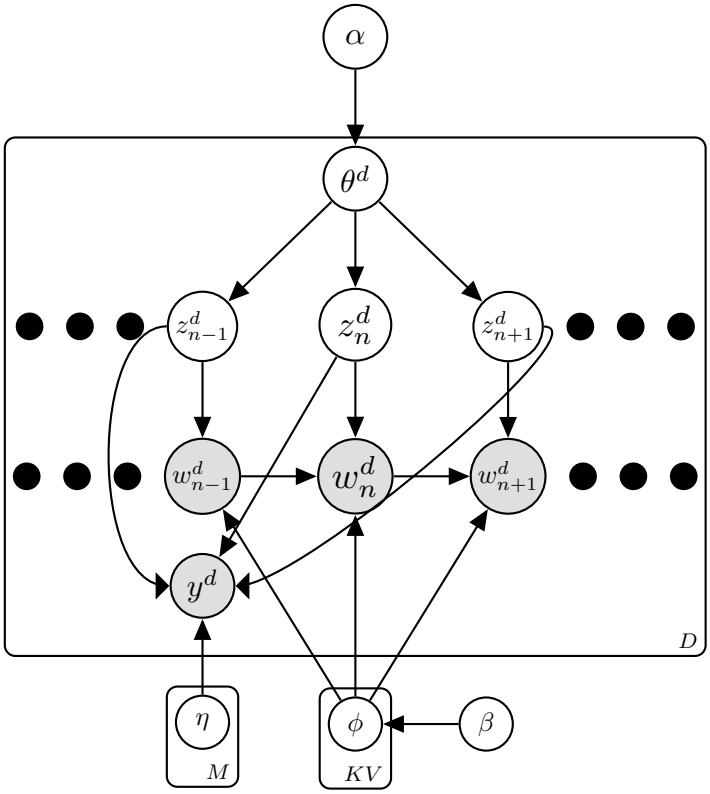

Fig. 2 Graphical representation of our proposed document classification model.

topic model itself. It enables interaction among such information for more effective modeling for document classification. There are two main components. One component is a topic model with word order. The other component is the maximum margin model. One fundamental difference between `MedLDA` and our proposed model is that our model exploits the word order structure of a document. The design of the above two components leads to latent topic representation that is more discriminative, and also advantageous for supervised document classification learning problem.

The document content modeling component of our model is primarily a bigram topic model which captures dependencies between the words in sequence. Each topic is characterized by a distribution of bigrams. The goal of our model is to generate a latent topic representation that is suitable for classification task. We adopt the same notation from Section 3. In our model, word generation is defined by the conditional distribution $P(w_n^d|w_{n-1}^d, z_n^d)$. The word-topic distribution denoted by $\boldsymbol{\Phi}$ is different from `MedLDA`. $\boldsymbol{\Phi} = \{\boldsymbol{\phi}_{kv}\}_{v,k=1}^{V,K}$ are word-topic distribution. We depict the graphical model of our model in Figure 2. Note that we show the hyperparameter β explicitly in the graphical model. The generative process of our model is depicted below:

1. Draw **Multinomial** distribution ϕ_{zw} from a **Dirichlet** prior β for each topic z and each word w,
2. For each document d
   (a) Draw a topic proportion $\boldsymbol{\theta}^d$ for the document d from **Dirichlet** $(\boldsymbol{\alpha})$, where **Dirichlet** $(\boldsymbol{\alpha})$ is the Dirichlet distribution with the parameter $\boldsymbol{\alpha}$,
   (b) For each word w_n^d,
      i. Draw a topic z_n^d from **Multinomial** $(\boldsymbol{\theta}^d)$
      ii. Draw a word w_n^d from the distribution over words for the context defined by the topic z_n^d and the previous word w_{n-1}^d from **Multinomial** $(\phi_{w_{n-1}^d z_n^d})$
3. Draw the class label parameter $\boldsymbol{\eta}$ from **Normal** $(0, \boldsymbol{\eta}_0)$, where $\boldsymbol{\eta}_0$ is the hyperparameter for $\boldsymbol{\eta}$ and is sampled M times, where M is the number of classes considered in the classification problem,
4. Draw a class label $y^d|(\boldsymbol{z}^d, \boldsymbol{\eta})$ according to Equations 8 to 10.

Let $\boldsymbol{b}^d$ denote $\{b_{n,n+1}^d\}_{n=1}^{N^d-1}$, where $b_{n,n+1}^d$ denotes the words at the positions n and $n+1$ in the document d written as $b_{n,n+1}^d = (w_n^d, w_{n+1}^d)$. $\boldsymbol{W} = \{\boldsymbol{b}^d\}_{d=1}^D$ is the word order information. The prior distribution defined in the model is expressed as:

$$P_0(\boldsymbol{\Theta}, \boldsymbol{\Phi}, \boldsymbol{Z}) = \left(\prod_{d=1}^{D} P(\boldsymbol{\theta}^d|\boldsymbol{\alpha}) \prod_{n}^{N^d} P(z_n^d|\boldsymbol{\theta}^d) \right) \prod_{k=1}^{K} \prod_{v=1}^{V} P(\boldsymbol{\phi}_{kv}|\boldsymbol{\beta}) \tag{7}$$

In our model, the objective is to infer the joint distribution $P(\boldsymbol{\eta}, \boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi}|\boldsymbol{W}, \boldsymbol{\alpha}, \boldsymbol{\beta})$, where $\boldsymbol{\eta}$ is a random variable representing the parameter of the classification

model. In addition, the discriminant function is defined as:

$$F(y, \boldsymbol{\eta}, \boldsymbol{z}; \boldsymbol{b}^d) = \boldsymbol{\eta}^\intercal \boldsymbol{f}(y; \overline{\boldsymbol{z}}^d) \tag{8}$$

The above latent function cannot be directly used for prediction tasks for an observed input document as it involves random variables. Therefore, we take the expectation and define the effective discriminant function as follows:

$$F(y; \boldsymbol{b}^d) = \mathbb{E}_{p(\boldsymbol{\eta}, \boldsymbol{z}|\boldsymbol{b}^d)}[F(y, \boldsymbol{\eta}, \boldsymbol{z}; \boldsymbol{b}^d)] \tag{9}$$

The prediction rule incorporating the word order structure in the classification task is:

$$\hat{y} = \underset{y}{\text{argmax}} \quad F(y; \boldsymbol{b}^d) \tag{10}$$

Let C be a regularization constant, ξ^d be the slack variable and $l^d(y)$ be the loss function for the label y; all of which are positive. The soft-margin framework for our model can be written as:

$$\begin{aligned} \underset{P(\boldsymbol{\eta}, \boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi}) \in \mathbb{P}, \boldsymbol{\xi}}{\text{minimize}} \quad & \text{KL}[P(\boldsymbol{\eta}, \boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi}|\boldsymbol{W}, \boldsymbol{\alpha}, \boldsymbol{\beta})||P_0(\boldsymbol{\eta}, \boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi}|\boldsymbol{\alpha}, \boldsymbol{\beta})] - \mathbb{E}_q[\log \text{P}(\boldsymbol{W}|\boldsymbol{Z}, \boldsymbol{\Phi})] + \\ & \frac{C}{D} \sum_d \text{argmax}_y(l^d(y)) - \mathbb{E}_P[\boldsymbol{\eta}^\intercal(\boldsymbol{f}(y^d, \overline{\boldsymbol{z}}^d) - \boldsymbol{f}(y, \overline{\boldsymbol{z}}^d))] \\ \text{subject to} \quad & \mathbb{E}_P[\boldsymbol{\eta}^\intercal(\boldsymbol{f}(y^d, \overline{\boldsymbol{z}}^d) - \boldsymbol{f}(y, \overline{\boldsymbol{z}}^d))] \geq l^d(y) - \xi^d, \xi^d \geq 0, \forall d, \forall y, \end{aligned} \tag{11}$$

4.2 Posterior Inference

We use Collapsed Gibbs sampling for computing the posterior inference considering the word order structure in the document. Collapsed Gibbs sampler collapses out the nuisance parameters, and speeds up the posterior inference [Shafiei and Milios, 2006]. Equation 11 can be solved in two steps in alternate manner. The first step is to estimate $P(\boldsymbol{\eta})$ given $P(\boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi})$. In the second step, we need to estimate $P(\boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi})$ given $P(\boldsymbol{\eta})$. We can estimate $P(\boldsymbol{\eta})$ from the algorithm described in [Jiang et al., 2012] where we make use of Lagrange multipliers, but our topic modeling component is different and thus the distribution $P(\boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi})$ needs to be estimated. We define $\boldsymbol{\kappa}$ as follows:

$$\boldsymbol{\kappa} = \sum_{d=1}^{D} \sum_{y^d} \lambda_{y^d}^d \boldsymbol{\Delta} \boldsymbol{f}(y^d, \mathbb{E}[\overline{\boldsymbol{z}}^d]), \tag{12}$$

where $\boldsymbol{\kappa}$ is the mean of classifier parameters $\boldsymbol{\eta}$. When we place a $*$ with κ, it denotes the optimum solution. We describe an outline for estimation of topical bigrams below.

First, we can factorize the topic model component and the maximum margin parameter component as follows:

$$P(\boldsymbol{\eta}, \boldsymbol{\Theta}, \boldsymbol{\Phi}, \boldsymbol{Z}) = P(\boldsymbol{\eta}) P(\boldsymbol{\Theta}, \boldsymbol{\Phi}, \boldsymbol{Z}) \tag{13}$$

Let $\boldsymbol{\Delta f}(y^d, \overline{\boldsymbol{z}}^d)$ be defined as follows:

$$\boldsymbol{\Delta f}(y^d, \overline{\boldsymbol{z}}^d) = \boldsymbol{f}(y^d, \overline{\boldsymbol{z}}^d) - \boldsymbol{f}(y, \overline{\boldsymbol{z}}^d) \tag{14}$$

Based on Equation 13, the formulation for the optimum solution is given as follows:

$$P(\boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi}) \propto P(\boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi}, \boldsymbol{W}) e^{\boldsymbol{\kappa}^{(*)\intercal} \sum_{d=1}^{D} \sum_{y^d} (\lambda_{y^d}^d)^* \boldsymbol{\Delta f}(y^d, \overline{\boldsymbol{z}}^d)} \tag{15}$$

where $\lambda_{y^d}^d$ is the Lagrange multiplier. The problem now is to efficiently draw samples from $P(\boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi})$ and also compute the expectation statistics of the maximum margin classifier used in our model. In order to simplify the integrals, we can take advantage of conjugate priors. We can integrate out the intermediate variables $\boldsymbol{\Theta}, \boldsymbol{\Phi}$ and build a Markov chain whose equilibrium distribution is the resulting marginal distribution $P(\boldsymbol{Z})$.

Let Z be a normalization constant. We get the following marginalized posterior distribution for our model after integrating out $\boldsymbol{\Theta}, \boldsymbol{\Phi}$:

$$P(\boldsymbol{Z}) = \frac{P(\boldsymbol{W}, \boldsymbol{Z} | \boldsymbol{\alpha}, \boldsymbol{\beta})}{Z} e^{\boldsymbol{\kappa}^{(*)\intercal} \sum_{d=1}^{D} \sum_{y} (\lambda_{y}^d)^* \boldsymbol{\Delta f}(y, \overline{\boldsymbol{z}}^d)} \tag{16}$$

The original `BTM` model proposed in [Wallach, 2006] used EM algorithm for doing the approximation. But we have used collapsed Gibbs sampler. Therefore, in order to solve the first component on the right hand side of the above equation, collapsed Gibbs sampling for the model has to be implemented. The second component can be solved using any existing `SVM` implementation with some modifications based on the formulations used in our model.

Let m_{zwv} be the number of times the word w is generated by the topic z when preceded by the word v. q_{dz} is the number of times a word is assigned to the topic z in the document d. The element $\kappa_{y^d k}$ represents the contribution of the topic k in classifying a data point to the class y^d. The transition probability along with the maximum margin constraint can be expressed as:

$$P(z_n^d | \boldsymbol{W}, \boldsymbol{Z}_{\neg n}, \alpha, \beta) = \left(\frac{\alpha_{z_n^d} + q_{dz_n^d} - 1}{\sum_{z=1}^{K} \left(\alpha_z + q_{dz}\right) - 1} \times e^{\frac{1}{N^d} \sum_{y} (\lambda_y^d)^* (\boldsymbol{\kappa}^*_{y^d k} - \boldsymbol{\kappa}^*_{y k})} \right) \times \frac{\beta_{w_n^d} + m_{z_n^d w_n^d w_{n-1}^d} - 1}{\sum_{v=1}^{V} (\beta_v + m_{z_n^d w_n^d v}) - 1} \tag{17}$$

Note that all the counts used above exclude the current case i.e., the word being visited during sampling. When we use a $\neg$ sign in the subscript of a variable, it means that the variable corresponding to the subscripted index is removed from the calculation of the count. In the above equation, -1 mainly arises from the chain rule expansion of the Gamma function. The posterior estimates of the model can be written as:

$$P(z_n^d | \boldsymbol{W}, \boldsymbol{Z}_{\neg n}, \alpha, \beta) = \left(\frac{\alpha_{z_n^d} + q_{dz_n^d}}{\sum_{z=1}^{K} \left(\alpha_z + q_{dz}\right)} \times e^{\frac{1}{N^d} \sum_{y} (\lambda_y^d)^* (\boldsymbol{\kappa}^*_{y^d k} - \boldsymbol{\kappa}^*_{y k})} \right) \times \frac{\beta_{w_n^d} + m_{z_n^d w_n^d w_{n-1}^d}}{\sum_{v=1}^{V} (\beta_v + m_{z_n^d w_n^d v})} \tag{18}$$

4.3 Prediction for Unseen Documents

Our prediction framework also follows similar strategy for unseen documents using topic models as used in many other works [Jiang et al., 2012], [Yao et al., 2009]. Let the unseen document be denoted as d^{new}. We consider the notion of word order. The input for prediction task are unlabeled test data. The output is to predict the label for the new document d^{new}. We compute the point estimate of topics obtained in the matrix $\boldsymbol{\Phi}$ from the training data. This matrix is used in the prediction task. When the unseen document is given to the model, we need to determine the latent dimensions $\boldsymbol{z}^{d^{\text{new}}}$ for this unseen document. This is computed using the MAP estimate of $\boldsymbol{\Phi}$ to obtain $\hat{\boldsymbol{\Phi}}$. Specifically, we compute the $z_n^{d^{\text{new}}}$ in each new document d^{new} as follows:

$$P(z_n^{d^{\text{new}}} | \boldsymbol{z}_{\neg n}^{d^{\text{new}}}) \propto \hat{\phi}_{(z_n^{d^{\text{new}}}, w_n^{d^{\text{new}}}, w_{n-1}^{d^{\text{new}}})} (\alpha_{z_n^{d^{\text{new}}}} + q_{dz_n^{d^{\text{new}}}}) \tag{19}$$

Expectation statistics computation can be derived in a similar manner as the classifier described in [Jiang et al., 2012].

5 Document Classification Experiments

5.1 Experimental Setup

We conduct extensive experiments on document classification using some benchmark test collections. We also compare with many related comparative methods. In addition, we present some high quality topical words showing how our model generates interpretable topical words. In all our experiments for topic models, we run the sampler for 1000 iterations[1]. We have also removed stopwords[2] and performed stemming using Porter's stemmer[3]. Text pre-processing and vector space generation was done using Gensim package[4]. Five-fold cross validation is used as in [Zhu et al., 2012a]. In each fold, the macro-average across the classes is computed. Each model is run for five times. We take the average of the results obtained for all the runs and in all the folds.

We use four datasets, namely, 20 Newsgroups dataset[5], OHSUMED-23 dataset[6], TechTC-300 Test Collection for Text Categorization[7], and Reuters 21578 text categorization collection[8]. In OHSUMED-23, as adopted in [Joachims,

[1] In [Jiang et al., 2012], the authors have found out empirically that less than 100 iterations are sufficient for convergence of the collapsed Gibbs sampler. In contrast, we have set much a higher value.

[2] http://jmlr.org/papers/volume5/lewis04a/a11-smart-stop-list/english.stop

[3] We also tested the models without performing stemming. We found that stemmed collections fared better.

[4] https://radimrehurek.com/gensim/

[5] http://qwone.com/∼jason/20Newsgroups/

[6] http://disi.unitn.it/moschitti/corpora.htm

[7] http://techtc.cs.technion.ac.il/techtc300/techtc300.html

[8] http://ai-nlp.info.uniroma2.it/moschitti/corpora/Reuters21578-Apte-90Cat.tar.gz

Table 1 Details about different datasets used in the document classification experiments.

Dataset Name	Number of Classes	Total Documents	Average Document Per Class	Average Document Length
20 Newsgroups	20	20417	1024	1638
OHSUMED-23	23	20000	923	700
TechTC-300	295	57706	47	12892
Reuters-21578	91	15437	85	1017

Table 2 Table depicting precision, recall and F-measure values for different models in the 20 Newsgroups dataset.

Models	Precision	Recall	F-Measure
Our Model	**0.880**	**0.939**	**0.875**
gMedLDA	0.869	0.869	0.868
vMedLDA	0.865	0.865	0.867
sLDA	0.805	0.812	0.809
DiscLDA	0.756	0.780	0.751
LDA	0.859	0.858	0.858
LDA+SVM	0.835	0.920	0.862
BTM	0.877	0.848	0.862
BTM+SVM	0.835	0.920	0.862
LDACOL	0.843	0.914	0.862
LDACOL+SVM	0.845	0.932	0.864
TNG	0.845	0.932	0.865
TNG+SVM	0.832	0.866	0.861
NTSeg	0.766	0.905	0.866
NTSeg+SVM	0.869	0.845	0.858
SVM	0.825	0.910	0.852

Table 3 Table depicting precision, recall and F-measure values for different models in the OHSUMED-23 dataset.

Models	Precision	Recall	F-Measure
Our Model	**0.496**	**0.910**	**0.639**
gMedLDA	0.456	0.814	0.633
vMedLDA	0.489	0.821	0.629
sLDA	0.456	0.802	0.620
DiscLDA	0.402	0.735	0.587
LDA	0.465	0.801	0.626
LDA+SVM	0.463	0.798	0.631
BTM	0.422	0.767	0.610
BTM+SVM	0.545	0.776	0.622
LDACOL	0.534	0.742	0.630
LDACOL+SVM	0.534	0.744	0.625
TNG	0.432	0.711	0.623
TNG+SVM	0.442	0.710	0.620
NTSeg	0.531	0.779	0.634
NTSeg+SVM	0.522	0.765	0.623
SVM	0.483	0.903	0.630

Table 4 Table depicting precision, recall and F-measure values for different models in the TechTC300 dataset.

Models	Precision	Recall	F-Measure
Our Model	**0.321**	**0.315**	**0.314**
gMedLDA	0.319	0.309	0.310
vMedLDA	0.319	0.309	0.310
sLDA	0.314	0.309	0.304
DiscLDA	0.311	0.308	0.303
LDA	0.303	0.304	0.301
LDA+SVM	0.302	0.305	0.305
BTM	0.304	0.305	0.304
BTM+SVM	0.304	0.304	0.301
LDACOL	0.305	0.303	0.299
LDACOL+SVM	0.304	0.305	0.299
TNG	0.304	0.306	0.302
TNG+SVM	0.304	0.301	0.296
NTSeg	0.306	0.306	0.295
NTSeg+SVM	0.308	0.304	0.298
SVM	0.314	0.311	0.309

Table 5 Table depicting precision, recall and F-measure values for different models in the Reuters dataset.

Models	Precision	Recall	F-Measure
Our Model	**0.421**	**0.414**	**0.419**
gMedLDA	0.409	0.408	0.403
vMedLDA	0.413	0.408	0.408
sLDA	0.309	0.401	0.319
DiscLDA	0.309	0.399	0.311
LDA	0.311	0.401	0.321
LDA+SVM	0.311	0.401	0.321
BTM	0.312	0.401	0.320
BTM+SVM	0.311	0.401	0.321
LDACOL	0.311	0.403	0.319
LDACOL+SVM	0.311	0.402	0.309
TNG	0.313	0.401	0.311
TNG+SVM	0.313	0.403	0.312
NTSeg	0.313	0.399	0.312
NTSeg+SVM	0.314	0.402	0.311
SVM	0.413	0.409	0.402

1998], we used the first 20,000 documents. We present the details about the datasets in Table 1. In the table, the first column presents the names of different datasets. The second column describes the total number of classes in the dataset. The third column presents the total number of documents in that entire dataset. The fourth column shows the average number of documents in the each class. The fifth column presents the average length of the documents in the entire dataset. One can see that we have used both small and large document collections.

Table 6 Table depicting the number of latent topics K obtained using the validation process, which was used in the test set for different models in different datasets.

Models	20 Newsgroups	OHSUMED-23	TechTC300	Reuters-21578
Our Model	80	70	10	20
gMedLDA	50	40	30	20
vMedLDA	30	60	50	30
sLDA	60	60	20	10
DiscLDA	70	70	30	50
LDA	50	40	40	70
LDA+SVM	50	40	20	80
BTM	80	60	30	90
BTM+SVM	80	40	60	20
LDACOL	60	50	10	50
LDACOL+SVM	70	50	20	70
TNG	70	60	20	10
TNG+SVM	60	60	20	20
NTSeg	60	40	40	50
NTSeg+SVM	60	40	90	10

In all our datasets, we used the validation set for determining the number of topics. The validation set consisted of approximately 20% of the documents. The training set comprised of approximately 60% documents and the test set consisted of approximately 20% of the documents. We use Precision, Recall and F-measure to measure the classification performance. The definitions for these metrics in the classification task can be found in [Jameel and Lam, 2013b]. We solve multiclass classification problem by decomposing into binary classification problems in each class. But this procedure also introduces the problem related to unbalanced data as stated in Nallapati [2004]. We therefore adopted the technique of under-sampling in which samples from majority class in both classes are made equal Nallapati [2004]. Empirical evidence suggests that such method generally produces better results as pointed by Zhang and Mani [2003]. We used the training set to train the model and we varied the number of topics from 10 to 100 in steps of 10 as in [Jameel and Lam, 2013b]. Then the trained model was validated on the validation set. We performed this procedure in each fold and computed the average F-measure. The number of topics which produced the best F-measure is the output of the validation process process. Then we used the test set to test the models using the number of topics obtained from the validation process. We set the loss function $(l^d(y))$ to a constant function 16 just as in [Jiang et al., 2012]. For simplicity, we assume all symmetric bigram Dirichlet prior, and we set the value of β to 0.01. The settings for other hyperparameters remain the same as in [Jiang et al., 2012] for fair comparison. As experimented in [Wang and McCallum, 2006], we also found not much variation in results with different hyperparameter values. Hyperparameter values of the other topic models (supervised and unsupervised) are the same as used in their respective works and their available publicly shared implementations. This ensures that we are using the best configurations for each of the models. In [Jiang et al., 2012], the authors

conduct extensive experimentation to find the best C value. We use the same C value for fair comparison. We also found that different values of C did not have much effect on the results.

We chose a wide range of comparative methods as follows. 1) Gibbs `MedLDA` [Zhu et al., 2013a] denoted as `gMedLDA`, 2) Variational `MedLDA` [Zhu et al., 2009] denoted as `vMedLDA`, 3) Supervised `LDA` denoted as `sLDA` [Blei and McAuliffe, 2008], 4) Discriminative `LDA` [Lacoste-Julien et al., 2008] denoted as `DiscLDA`, 5) `LDA` [Blei et al., 2003], 6) We use `LDA+SVM` in the same way as described in [Zhu et al., 2012a], 7) Bigram Topic Model `BTM` [Wallach, 2006], 8) Following procedure as adopted for `LDA+SVM`, we do the same for `BTM+SVM`, 9) `LDA`-Collocation model (`LDACOL`) [Griffiths et al., 2007], 10) `LDACOL+SVM`, 11) Topical N-gram (`TNG`) [Wang et al., 2007], 12) `TNG+SVM`, [Joachims, 1998], 13) a recently proposed model `NTSeg` [Jameel and Lam, 2013b], 14) `NTSeg+SVM`, 15) `SVM`. The features for linear `SVM` are same as that in [Zhu et al., 2013a].

5.2 Quantitative Results

We present our main classification results in Tables 2, 3, 4 and 5. We observe that our model has outperformed all the comparative methods. In all datasets, our F-measure results are statistically significant based on the sign test with a p-value < 0.05 against each of the comparative methods. By maintaining the word order and considering an extra side-information helps in improving classification results to a great extent. Since we are capturing the inherent word order semantics in the document, just like other structured unsupervised topic models, we obtained improvements over the comparative methods.

In Table 6 we present the results for the number of topics obtained during the validation process. These topics were subsequently used in the test set to compute the final results that we have depicted in Tables 2, 3, 4 and 5.

In Tables 7, 8, 9, and 10, we study the effect of document classification performance as measured by F-measure when we vary the number of topics from 10 to 100 for topic models in different datasets. As we begin from $K = 10$ in the 20 Newsgroups dataset, we see that our model does not perform very well in the beginning. Nevertheless, it still outperforms other topic models. Our model performs very well after $K \geq 70$. Similarly, in the OHSUMED-23 dataset, our model also does not perform well until $K \leq 60$. Nevertheless, it still outperforms other topic models. Then it gains good improvement as we increase the number of latent topics. Also, the unsupervised n-gram[9] topic models' performance cannot be discarded. One observation is that the recently proposed unsupervised n-gram topic model `NTSeg` has done well when compared to other unsupervised topic model in the 20 Newsgroups dataset. Similar pattern is observed in the OHSUMED-23 dataset. In the TechTC300, all the models show poor performance. This shows that the dataset has difficult examples which the topic models find difficult to classify. In Reuters too our

[9] By n-gram we mean either a unigram, a bigram, etc.

Table 7 The effect of the number of topics on document classification measured by F-measure in the 20 Newsgroups dataset.

Models	10	20	30	40	50	60	70	80	90	100
Our Model	0.783	0.843	0.845	0.856	0.859	0.865	0.874	0.875	0.875	0.874
gMedLDA	0.424	0.694	0.826	0.859	0.868	0.866	0.858	0.869	0.852	0.850
vMedLDA	0.245	0.667	0.867	0.852	0.843	0.831	0.818	0.802	0.789	0.777
sLDA	0.301	0.505	0.578	0.789	0.800	0.809	0.766	0.698	0.653	0.493
DiscLDA	0.245	0.452	0.643	0.654	0.701	0.743	0.751	0.699	0.636	0.545
LDA	0.410	0.683	0.816	0.849	0.858	0.856	0.848	0.859	0.842	0.840
LDA+SVM	0.752	0.802	0.827	0.837	0.862	0.844	0.850	0.851	0.842	0.839
BTM	0.715	0.775	0.831	0.846	0.854	0.853	0.857	0.862	0.859	0.856
BTM+SVM	0.552	0.602	0.807	0.816	0.849	0.857	0.863	0.862	0.856	0.787
LDACOL	0.601	0.633	0.701	0.699	0.843	0.862	0.854	0.833	0.765	0.799
LDACOL+SVM	0.545	0.601	0.812	0.824	0.834	0.859	0.864	0.851	0.855	0.799
TNG	0.552	0.615	0.803	0.819	0.831	0.857	0.865	0.835	0.803	0.772
TNG+SVM	0.556	0.612	0.816	0.824	0.835	0.861	0.866	0.859	0.862	0.845
NTSeg	0.601	0.612	0.654	0.670	0.840	0.866	0.845	0.756	0.722	0.626
NTSeg+SVM	0.646	0.640	0.745	0.801	0.855	0.858	0.806	0.703	0.603	0.515

Table 8 The effect of the number of topics on document classification measured by F-measure in the OHSUMED-23 dataset.

Models	10	20	30	40	50	60	70	80	90	100
Our Model	0.597	0.600	0.605	0.616	0.630	0.633	0.639	0.639	0.638	0.638
gMedLDA	0.543	0.555	0.580	0.633	0.621	0.613	0.588	0.590	0.574	0.534
vMedLDA	0.542	0.556	0.552	0.558	0.585	0.629	0.632	0.611	0.589	0.534
sLDA	0.543	0.545	0.512	0.555	0.534	0.620	0.613	0.603	0.603	0.585
DiscLDA	0.503	0.502	0.512	0.507	0.532	0.611	0.587	0.575	0.545	0.543
LDA	0.545	0.593	0.565	0.626	0.611	0.615	0.601	0.599	0.546	0.600
LDA+SVM	0.542	0.585	0.556	0.631	0.605	0.610	0.587	0.585	0.535	0.598
BTM	0.546	0.590	0.594	0.630	0.630	0.610	0.576	0.554	0.523	0.554
BTM+SVM	0.511	0.545	0.578	0.622	0.625	0.613	0.572	0.553	0.526	0.524
LDACOL	0.513	0.575	0.565	0.631	0.630	0.601	0.569	0.523	0.514	0.515
LDACOL+SVM	0.499	0.504	0.560	0.631	0.625	0.601	0.567	0.522	0.512	0.531
TNG	0.523	0.572	0.554	0.610	0.625	0.623	0.621	0.524	0.552	0.520
TNG+SVM	0.524	0.573	0.550	0.606	0.622	0.620	0.622	0.527	0.543	0.519
NTSeg	0.524	0.579	0.560	0.634	0.629	0.598	0.554	0.515	0.512	0.555
NTSeg+SVM	0.516	0.560	0.554	0.623	0.612	0.584	0.498	0.515	0.513	0.525

model shows good performance as the number of latent topics is varied from 10 to 100. It suggests that considering the word order can offer some contributions to document classification performance. Our model can outperform the other comparative methods because it inherits the advantages of both n-gram unsupervised topic models and supervised topic models. Note that as exemplified in [Jameel and Lam, 2013b] and many other works which follow word order, computational complexity of the models that follow word order is generally higher than those of their bag-of-words counterparts. Nevertheless, models incorporating word order structure have shown superior performance than the bag-of-words models [Jameel and Lam, 2013b]. Several attempts have been made recently to speed up the inference procedures for both supervised

Table 9 The effect of the number of topics on document classification measured by F-measure in the TechTC-300 dataset.

Models	10	20	30	40	50	60	70	80	90	100
Our Model	0.314	0.314	0.314	0.313	0.314	0.313	0.312	0.312	0.313	0.313
gMedLDA	0.310	0.310	0.310	0.310	0.309	0.309	0.309	0.309	0.310	0.309
vMedLDA	0.310	0.310	0.309	0.310	0.310	0.310	0.309	0.309	0.309	0.310
sLDA	0.304	0.304	0.304	0.304	0.303	0.304	0.304	0.303	0.303	0.302
DiscLDA	0.302	0.301	0.303	0.303	0.303	0.303	0.303	0.302	0.302	0.301
LDA	0.299	0.299	0.298	0.301	0.301	0.301	0.301	0.301	0.290	0.292
LDA+SVM	0.304	0.305	0.305	0.304	0.304	0.304	0.303	0.304	0.303	0.303
BTM	0.302	0.302	0.304	0.303	0.303	0.303	0.303	0.304	0.301	0.302
BTM+SVM	0.299	0.300	0.301	0.300	0.300	0.301	0.301	0.299	0.299	0.300
LDACOL	0.299	0.299	0.298	0.298	0.297	0.292	0.293	0.291	0.293	0.291
LDACOL+SVM	0.299	0.299	0.298	0.298	0.297	0.298	0.296	0.295	0.291	0.295
TNG	0.301	0.302	0.301	0.301	0.299	0.301	0.294	0.298	0.291	0.298
TNG+SVM	0.295	0.296	0.296	0.295	0.294	0.293	0.294	0.294	0.295	0.292
NTSeg	0.293	0.292	0.293	0.295	0.295	0.293	0.291	0.292	0.291	0.290
NTSeg+SVM	0.291	0.291	0.293	0.291	0.292	0.294	0.295	0.297	0.298	0.298

Table 10 The effect of the number of topics on document classification measured by F-measure in the Reuters-21578 dataset.

Models	10	20	30	40	50	60	70	80	90	100
Our Model	0.415	0.419	0.418	0.418	0.418	0.417	0.413	0.414	0.415	0.413
gMedLDA	0.401	0.403	0.403	0.401	0.402	0.401	0.403	0.402	0.402	0.401
vMedLDA	0.401	0.401	0.408	0.408	0.407	0.402	0.401	0.403	0.404	0.407
sLDA	0.319	0.315	0.312	0.312	0.310	0.310	0.310	0.309	0.310	0.306
DiscLDA	0.310	0.309	0.309	0.311	0.311	0.302	0.304	0.303	0.305	0.307
LDA	0.311	0.315	0.312	0.317	0.315	0.319	0.321	0.321	0.320	0.321
LDA+SVM	0.319	0.318	0.317	0.318	0.319	0.320	0.320	0.321	0.321	0.321
BTM	0.312	0.311	0.312	0.315	0.315	0.318	0.318	0.317	0.320	0.319
BTM+SVM	0.319	0.321	0.320	0.320	0.320	0.320	0.320	0.319	0.320	0.319
LDACOL	0.316	0.315	0.317	0.318	0.319	0.319	0.318	0.311	0.299	0.301
LDACOL+SVM	0.305	0.304	0.304	0.302	0.305	0.308	0.309	0.309	0.308	0.308
TNG	0.311	0.311	0.310	0.310	0.309	0.302	0.304	0.309	0.309	0.309
TNG+SVM	0.311	0.312	0.312	0.311	0.312	0.311	0.312	0.309	0.305	0.306
NTSeg	0.309	0.311	0.306	0.305	0.312	0.305	0.306	0.311	0.310	0.311
NTSeg+SVM	0.311	0.310	0.310	0.311	0.310	0.311	0.310	0.309	0.301	0.304

and unsupervised topic models such as [Zhu et al., 2013b], [Zhu et al., 2013c], [Porteous et al., 2008].

5.3 Examples of Topical Words

We present some high probability topical words in topics and compare our model with some related n-gram and supervised topic models, including `BTM` [Wallach, 2006], `LDACOL` [Griffiths et al., 2007], `TNG` [Wang et al., 2007], `PDLDA` [Lindsey et al., 2012], `NTSeg` [Jameel and Lam, 2013b], `MedLDA` [Zhu et al., 2012a]. We present top five most representative words from a topic describing

semantically similar theme from each model. We chose the documents from *comp.graphics* class in order to present the list of topical words in this experiment experiments as adopted in [Zhu et al., 2012a].

The objective for presenting a list of topical words for comparison is to show the words in each topic and whether they give some insight about the topic. Obviously, words which are ambiguous will not make sense to a reader about the topic, and we can then infer that the topic model is unable to generate interpretable latent topics. Note that many works related to topic models present some top-k words from some topics, but this analysis cannot be regarded as a very strong indication about the superiority of one topic model over the other. This is why quantitative analysis is very important which we have already shown, and where our model has performed better than the comparative methods.

Table 11 Top five probable words from a topic from *comp.graphics* class of 20 Newsgroups dataset.

BTM	**LDACOL**	**TNG**	**PDLDA**
compgraph path	xref	vga mode	excel digit
xref compgraph	compgraph	routine	remove
system distribution	compgraph path	pixmap	public domain
problem solving	mark	public domain	draw line
fast purpose	compgraph subject	credit	message id

Table 12 Top five probable words from a topic from *comp.graphics* class of 20 Newsgroups dataset.

NTSeg	**MedLDA**	**Our Model**
surface normal	path	bitmap draw
orient message id	routing	video memory
corporate	college	simple routing
copyright	date	color gif
make group	sender	package zip

From the results shown in Table 11 and 12, we can make two observations. First, our model generates more fine grained topical words as compared to other topic models. Second, our model generates more interpretable latent topics as compared to other topics. Words such as “video memory”, “ simple routing”, “package zip” appear to make some sense to a reader. For example, “package zip” is a bigram which might be describing about zipping the contents of a file. Overall, most of the bigrams in the topic generated by our model seem to suggest that our model has generated words which relate to the domain “computer graphics”. Other models rather generate ambiguous n-grams or they generate unigrams which do not offer much understanding to the user, for

instance, bigrams generated by the BTM model does not seem to suggest that the topic is describing about "computer graphics" as words such as "compgraph path", "xref compgraph", etc are not very insightful to a reader.

6 Topic Model for Document Retrieval Learning

6.1 Model Description

We also investigate a supervised low-dimensional latent topic model for document retrieval learning. Suppose that some relevance assessments of documents for some queries are available for training. Our goal is to learn a model that can predict the relevance of an unseen test query-document pair, and rank the documents based on the predicted relevance score. This problem setting is similar to the pointwise learning-to-rank problem. Manual relevance assessments can be modeled as a response variable in our topic model. In addition, the word order structure of the text content is also considered. The main motivation for considering the word order is to capture the semantic story inherent in the document which is supposedly lost when the order of words in the document is broken. Similar to our proposed document classification model, there are two main components in our document retrieval learning model. One component is a topic model which measures the goodness of fit of the text content of documents and queries. Queries are modeled as short documents in a similar manner as in [Wu and Zhong, 2013], [Salton et al., 1975]. Our topic model considers the word order structure in documents and queries. The second component deals with the relevance prediction within a maximum margin framework. Labels are mainly predicted using the maximum margin framework in our pointwise retrieval learning model. The dataset can be represented as $((d,q), y_{(d,q)})$ composed of query-document pairs (d,q) along with the relevance assessment label denoted by $y_{(d,q)}$ which signifies the relevance of the document d to the query q. Let $c(d,q)$ be the total number of query-document pairs in the training set. Let the number of documents in the training set be D; the number of queries in the training set be Q. As adopted in [Nallapati, 2004], the confidence scores obtained from the discriminant function is used to rank documents in our proposed model. Let the words in the document d be represented by $\boldsymbol{w}^d$ and the words in the query q be represented by $\boldsymbol{w}^q$. Let the set of topics used in the document d be represented as $\boldsymbol{z}^d$, and the set of topics in the query q be represented by $\boldsymbol{z}^q$.

There are several fundamental differences between our document retrieval learning framework with those of the previously proposed supervised topic models. In our model, each input data instance consists of a pair of document and query instead of a single document. In contrast to other supervised topic models such as [Jiang et al., 2012], [Zhu et al., 2009], [Zhu et al., 2012a], the property of the feature vector is different. In our retrieval learning model, feature vector includes different query-dependent and query-independent features which are useful for conducting the learning-to-rank task.

We first describe a new discriminant function which is suited for handling document retrieval learning problem. Therefore, the discriminant function of our model is designed as follows:

$$F(y, \boldsymbol{\eta}, (d, q)) = \boldsymbol{\eta}^{\intercal} \boldsymbol{f}(y, (d, q)) \tag{20}$$

where $\boldsymbol{\eta}$ represents the model parameters which are essentially feature weights. $\boldsymbol{f}(y, (d, q))$ is a vector of features which are designed to be useful for retrieval. The new definitions of $\boldsymbol{\eta}$ and $\boldsymbol{f}(y, (d, q))$ make it suitable to handle document retrieval task. Some examples of features are depicted in Table 13. Note that just as in LETOR learning-to-rank datasets Qin et al. [2010], these features are computed for the entire dataset $\mathbb{D}$ before generating the training, test and the validation sets. $c(w_n^d, d)$ is the number of times the word w_n^d appears in the document d. N^q is the number of words in the query q. $|.|$ denotes the size function. idf is the inverse document frequency. The first six features have also been used in [Nallapati, 2004] where readers can find the motivation behind the design of these features. Some minor refinements to some of these six features were made in [Xu and Li, 2007], [Qin et al., 2010], and we use these refined features in our experimental setup. The last feature, called topic similarity feature, is a similarity measure between the topics of the query and the document in the low-dimensional topic space generated by our model. Let $\boldsymbol{Z}^d = \{\boldsymbol{z}^d\}_{d=1}^{D}$ be topic assignments to all the words of the training documents; $\boldsymbol{Z}^q = \{\boldsymbol{z}^q\}_{q=1}^{Q}$ be topic assignments to all the words in the training queries; $\boldsymbol{\Theta}^d = \{\boldsymbol{\theta}^d\}_{d=1}^{D}$ be topic distributions for all training documents; $\boldsymbol{\Theta}^q = \{\boldsymbol{\theta}^q\}_{q=1}^{Q}$ be topic distributions for all training queries; $\boldsymbol{\Phi} = \{\boldsymbol{\phi}_{kv}\}_{v,k=1}^{V,K}$ be the word-topic distribution. In order to compute the topic similarity in the low-dimensional topic space between the document and the query, we make use of the topic-document and topic-query distributions $\boldsymbol{\Theta}^d$ and $\boldsymbol{\Theta}^q$. In each of these distributions, we consider each document or query represented as a $K \times 1$, which mainly is $P(z \in K|d)$ or $P(z \in K|q)$ where d is a document and q is a query, low-dimensional vector in the latent topic space. Each of the values in this vector can be considered as a weight for the corresponding latent topic [Hazen, 2010] or simply the contribution of a topic to a document. Consider a document d associated with a query q, and thus is also represented by its own low-dimensional latent topic vectors. Let the latent topic vector for the document d be denoted as $v^d = K_d \times 1$ and let the latent topic vector of the query q be represented as $v^q = K_q \times 1$. We compute the cosine similarity[10] between these two vectors. The intuitive idea is that if the two vectors are close to each other in the latent topic space i.e. if they are semantically related to each other even though they do not share the same words, they tend to have a high cosine similarity value in the latent topic space. In fact, works such as [Liu et al., 2009], [Maas et al., 2011] have also used cosine similarity between words and documents in the latent topic space. Other similarity metrics such as KL-Divergence could also be used.

[10] This feature is formulated as a cosine similarity of v^d and v^q denoted by cosine(v^d, v^q).

Table 13 Features used in our discriminant function in our document retrieval learning model.

Feature	**Feature**
1. $\sum_{w_n^q \in q \cap d} \log\left(c(w_n^q, d) + 1\right)$	2. $\sum_{w_n^q \in q \cap d} \log\left(1 + \frac{c(w_n^q, d)}{\lvert d \rvert}\right)$
3. $\sum_{w_n^q \in q \cap d} \log(\text{idf}(w_n^q))$	4. $\sum_{w_n^q \in q \cap d} \log\left(\frac{\lvert \mathbb{D} \rvert}{c(w_n^q, d)} + 1\right)$
5. $\sum_{w_n^q \in q \cap d} \log\left(1 + \frac{c(w_n^q, d)}{\lvert d \rvert} \text{idf}(w_n^q)\right)$	6. $\sum_{w_n^q \in q \cap d} \log\left(1 + \frac{c(w_n^q, d)}{\lvert d \rvert} \frac{\lvert \mathbb{D} \rvert}{c(w_n^q, \mathbb{D})}\right)$
7. Topic Similarity Feature - cosine(v^d, v^q)	

Unlike the classification model where we took the expectation, the effective discriminant function which is obtained from Equation 20 as follows:

$$F(y, (d, q)) = [F(y, \boldsymbol{\eta}, (d, q))] \tag{21}$$

The prediction rule is given in Equation 22, where our objective is to find a label is as follows:

$$\hat{y} = \underset{y}{\text{argmax}} \quad F(y, (d, q)) \tag{22}$$

The following maximum margin constraints are imposed:

$$F(y_{(d,q)}, (d, q)) - F(y, (d, q)) \geq l_{(d,q)}(y) - \xi_{(d,q)}, \forall y \in Y, \forall (d, q) \tag{23}$$

where $l_{(d,q)}(y)$ is a non-negative loss function. $\xi_{(d,q)}$ are non-negative slack variables which are meant for inseparable data instances. C is a positive regularization constant. The soft-margin framework for our model is described below:

$$\begin{aligned} \underset{P(\boldsymbol{\Theta}^d, \boldsymbol{\Theta}^q, \boldsymbol{Z}^d, \boldsymbol{Z}^q, \boldsymbol{\Phi}) \in \mathbb{P}, \xi, \boldsymbol{\eta}}{\text{minimize}} \; & \text{KL}\left[P(\boldsymbol{\Theta}^d, \boldsymbol{\Theta}^q, \boldsymbol{Z}^d, \boldsymbol{Z}^q, \boldsymbol{\Phi}) || P_0(\boldsymbol{\Theta}^d, \boldsymbol{\Theta}^q, \boldsymbol{Z}^d, \boldsymbol{Z}^q, \boldsymbol{\Phi})\right] - \\ & \mathbb{E}_P[\log P(\boldsymbol{W}^d, \boldsymbol{W}^q | \boldsymbol{\Theta}^d, \boldsymbol{\Theta}^q, \boldsymbol{Z}^d, \boldsymbol{Z}^q, \boldsymbol{\Phi})] + \frac{C}{c(d, q)} \sum_{(d,q)} \xi_{(d,q)} \\ \text{subject to} \quad & [\boldsymbol{\eta}^\intercal (\boldsymbol{f}(y_{(d,q)}, d, q) - \boldsymbol{f}(y, d, q,))] \geq l_{(d,q)}(y) - \xi_{(d,q)}, \xi_{(d,q)} \geq 0, \forall (d, q), \forall y \end{aligned} \tag{24}$$

6.2 Posterior Inference

In order to proceed with the derivation of the collapsed Gibbs sampling, we need to define a joint distribution for words and the topics along with the regularization effects due to the maximum margin posterior constraints. In this model too we need to alternatively find the optimal solution using maximum margin classifier and solve the topic model component. But unlike the posterior inference of the classification model, we can directly adopt implementation from existing `SVM` algorithm to find the optimum solution of the classifier. Let

$\boldsymbol{\eta}^{(*)}$ denote the optimum parameter weights. This joint distribution is written as:

$$P(\boldsymbol{Z}^d, \boldsymbol{W}^d, \boldsymbol{Z}^q, \boldsymbol{W}^q|\boldsymbol{\alpha}, \boldsymbol{\beta}) = P(\boldsymbol{W}^d|\boldsymbol{Z}^d, \boldsymbol{\beta}) \times P(\boldsymbol{W}^q|\boldsymbol{Z}^q, \boldsymbol{\beta}) \times P(\boldsymbol{Z}^d|\boldsymbol{\alpha}) \times P(\boldsymbol{Z}^q|\boldsymbol{\alpha}) \times e^{\boldsymbol{\eta}^{(*)\intercal} \sum_{(d,q)} \sum_{y=1}^{M} (\lambda^{y}_{(d,q)})^* (\boldsymbol{f}(y_{(d,q)},(d,q)) - \boldsymbol{f}(y,(d,q)))} \tag{25}$$

After some manipulations, we can come up with the following update equation:

$$P(z_n^d, z_n^q|\mathbf{W}^d, \mathbf{W}^q, \mathbf{Z}^d_{\neg n}, \mathbf{Z}^q_{\neg n}, \alpha, \beta) = \left(\frac{\alpha_{z_n^d} + m_{z_n^d w_n^d} - 1}{\sum_{z=1}^{K} \left(\alpha_z + m_z\right) - 1} \times \frac{\alpha_{z_n^q} + m_{z_n^q w_n^q} - 1}{\sum_{z=1}^{K} \left(\alpha_z + m_z\right) - 1} \times e^{\frac{1}{(N^d + N^q)} \sum_{y=1}^{M} (\lambda^{y}_{(d,q)})^* (\boldsymbol{f}(y_{(d,q)},(d,q)) - \boldsymbol{f}(y,(d,q)))^*} \right) \times \frac{\beta_{w_n^d} + m_{z_n^d w_n^d w_{n-1}^d} - 1}{\sum_{v=1}^{V} (\beta_v + m_{z_n^d w_n^d v}) - 1} \times \frac{\beta_{w_n^q} + m_{z_n^q w_n^q w_{n-1}^q} - 1}{\sum_{v=1}^{V} (\beta_v + m_{z_n^q w_n^q v}) - 1} \tag{26}$$

where m_{zwv} is the number of times the word w is generated by the topic z when preceded by the word v and is applicable to a document and a query when super-scripted by d or q respectively. m_{zw} is the number of times a word w in the document has been sampled in the topic z, and is applicable to a document and query when super-scripted by d or q respectively.

One can argue that asymmetric priors may work better especially on short documents such as queries. Many previous works for short documents have assumed asymmetric priors in their topic models such as [Yan et al., 2013], [Hasler et al., 2014]. Our model is flexible enough to accommodate asymmetric priors, but in this paper we only test our model using symmetric priors for simplicity. In [Nallapati, 2004] the author discussed some shortcomings in discriminative models for IR, in particular, the out-of-vocabulary words. The author has also suggested a few ways of dealing with those shortcomings. We also follow those strategies in this paper.

6.3 Ranking Unseen Query-Document Pairs

The prediction task on test data using the prediction rule given in Equation 22 can be realized as follows. Let $(q^{\text{new}}, d^{\text{new}})$ be an unseen test query-document pair for which we need to predict the relevance label. The task is to compute the latent topic representations of q^{new} and d^{new} using the topic space that has been learned from the training data. These latent components for the unseen query and the document can be obtained from $\hat{\boldsymbol{\Phi}}$ which is the maximum aposteriori estimate of $P(\boldsymbol{\Phi})$ computed during the training process. Suppose there are J samples from a proposal distribution, $\hat{\boldsymbol{\Phi}}$ is obtained using the

samples from the following equation:

$$\hat{\phi}_{zwv} \propto \frac{1}{J}\sum_{j=1}^{J}(\beta_{w_n^d} + m^{(j)}_{z_n^d w_n^d w_{n-1}^d}) \times (\beta_{w_n^q} + m^{(j)}_{z_n^d w_n^d w_{n-1}^d}) \tag{27}$$

where the counts are assigned in the j^{th} sample. The latent components for the unseen document and the query can be computed as follows.

$$P(z_n^{d^{\text{new}}}, z_n^{q^{\text{new}}} | \mathbf{W}^{d^{\text{new}}}, \mathbf{W}^{q^{\text{new}}}, \mathbf{Z}^{d^{\text{new}}}_{\neg n}, \mathbf{Z}^{q^{\text{new}}}_{\neg n}, \alpha, \beta) \propto \hat{\phi}_{z_n^{d^{\text{new}}} w_n^{d^{\text{new}}} w_{n-1}^{d^{\text{new}}}} (\alpha_{z_n^{d^{\text{new}}}} + m_{z_n^{d^{\text{new}}}}) \times \hat{\phi}_{z_n^{q^{\text{new}}} w_n^{q^{\text{new}}} w_{n-1}^{d^{\text{new}}}} (\alpha_{z_n^{q^{\text{new}}}} + m_{z_n^{q^{\text{new}}}}) \tag{28}$$

where the count for the word being sampled is excluded. We compute the similarity between the query and the document in the latent topic space. Note that $y_{(d,q)}$ can be dropped during the prediction step. The maximum margin prediction of labels for unseen vectors follows the standard maximum margin formulation Yu and Kim [2012]. Note that this formalism is different from the expectation based maximum margin classifier discussed previously for document classification. When the task of computing the similarity score is accomplished, it can be used in Equation 20 to compute the prediction score. Documents can be ranked based on this confidence score.

7 Retrieval Learning Experiments

7.1 Experimental Setup

We conduct document retrieval learning experiments using benchmark text collections. We will show the performance of our model by conducting extensive quantitative analysis. In addition, we will also present some high probability topical words from topics, and show how our model is able to generate better topical words leading to more interpretable topics. In all our experiments, we run the Gibbs sampler of our model for 1000 iterations. We removed stopwords, and performed stemming using Porter's stemmer.

We use four test collections for our experiments. We used a benchmark OHSUMED test collection (latest version[11]) from the LETOR [Qin et al., 2010] dataset. This dataset consists of 45 comprehensive features along with query-document pairs with their relevance judgments. It has been used extensively in evaluating several learning-to-rank algorithms. We obtained raw documents and queries of this dataset from the web[12] in order to get the word order. This dataset contains the document-id along with the list of features, which will help us relate which set of features in LETOR OHSUMED is associated with which document. Our proposed feature i.e. the topic similarity feature is treated as

[11] Minka et al., [Minka and Robertson, 2008] and some other researchers had pointed out few shortcomings in the earlier LETOR releases.

[12] http://ir.ohsu.edu/ohsumed/

Table 14 Number of features in each dataset used in document retrieval learning experiments.

Dataset	Number of Features
LETOR OHSUMED	45
AQUAINT	6
WT2G	6
ClueWeb09-English	91

one feature, in addition to the existing 45 features. It has approximately 60% query-document pairs in the training set, 20% in the validation set, and the rest in the test set in each of the five folds. For a particular fold, the queries involved in the training, the validation, and the test set are different. Validation set is used by the comparative learning-to-rank models for parameter tuning and determining the number of iterations. Our second collection is AQUAINT used in TREC HARD[13]. Basic details about this dataset can be found in [Allan, 2005]. Note that we only consider document-level relevance assessments in AQUAINT, and leave out the passage-level judgments. The third dataset is WT2G[14], along with the standard relevance judgments and topics (401 - 450) obtained from the TREC site. The fourth dataset is the Category B English documents from ClueWeb09 collection. This dataset has been obtained from the authors of [Asadi and Lin, 2013]. In order to create the training, test and validation datasets for AQUAINT and WT2G, we adopted the strategies popularly used in the learning-to-rank problems. We chose the same percentage of query-document pairs in the training, test and validation set in each fold as in LETOR OHSUMED dataset. The features used for AQUAINT and WT2G datasets are given in Table 13. Note that only the number of features differ in the datasets that we generated (WT2G and AQUAINT) when compared to LETOR OHSUMED. We present the number of features used in the document retrieval learning experiments in Table 14. Based on our proposed model, we also investigate another variant, called `Variant 1`, which we will test empirically and show its performance. In this model we ignore the word order structure in queries, but maintain the word order structure in documents. The reason is that queries are mostly short, and the role of word order might not be very significant. In addition, we also compare with another variant of our model and name it `Variant 2` where word order is not maintained in both queries and the documents. We use NDCG@5 and NDCG@10 as our metrics, similar to the metrics used in [Cai et al., 2011]. NDCG is well suited for our task because it is defined by an explicit position discount factor and it can leverage the judgments in terms of multiple ordered categories [Järvelin and Kekäläinen, 2002].

In order to determine the number of topics K, the parameter C, and the constant loss function $l_{(d,q)}(y)$ in our model, we use the validation set. We first

[13] http://ciir.cs.umass.edu/research/hard/guidelines2003.html
[14] http://ir.dcs.gla.ac.uk/test_collections/access_to_data.html

Table 15 Values for different parameters obtained using the validation set for our model in Fold 1.

Datasets	**Topics** (K)		C	$l_{(d,q)}(y)$
	NDCG@5	**NDCG@10**		
LETOR OHSUMED	110	110	60	1
AQUAINT	250	190	70	5
WT2G	250	170	50	4
ClueWeb-2009 Category B English	170	190	90	2

Table 16 Values for different parameters obtained using the validation set for our model in Fold 2.

Datasets	**Topics** (K)		C	$l_{(d,q)}(y)$
	NDCG@5	**NDCG@10**		
LETOR OHSUMED	120	130	60	1
AQUAINT	200	250	80	2
WT2G	70	150	50	2
ClueWeb-2009 Category B English	120	140	90	2

Table 17 Values for different parameters obtained using the validation set for our model in Fold 3.

Datasets	**Topics** (K)		C	$l_{(d,q)}(y)$
	NDCG@5	**NDCG@10**		
LETOR OHSUMED	110	140	60	2
AQUAINT	180	300	80	1
WT2G	90	140	50	2
ClueWeb-2009 Category B English	150	190	90	3

train our model on the training set, and measure NDCG@5 and NDCG@10 performance on the validation set. The number of topics and the model parameters can be automatically determined from the validation process. We then test our model using the test set. We varied the number of topics from 50 to 300 in steps of 10. We varied the values of C in multiples of 10. We vary $l_{(d,q)}(y)$ from 1 to 20 in steps of 1. We have again set a weak β prior which is 0.01. We have use symmetric Dirichlet priors for our model. We also found that varying the value of the hyperparameter does not drastically affect the results and this finding is consistent with [Wang and McCallum, 2006]. We also found out experimentally that different values of C does not significantly change the performance of the model. The experimental results are averaged over five folds for all the models. Each model is run only one time in each fold.

We compare the performance of our model with a range of comparative methods including popular learning-to-rank models in RankLib[15] such as `MART` [Friedman, 2001], `RankNet` [Burges et al., 2005], `AdaRank` [Xu and Li, 2007], `Coordinate Ascent` [Metzler and Croft, 2007], `LambdaRank` [Quoc and Le,

[15] http://people.cs.umass.edu/∼vdang/ranklib.html

Table 18 Values for different parameters obtained using the validation set for our model in Fold 4.

Datasets	**Topics** (K)		C	$l_{(d,q)}(y)$
	NDCG@5	**NDCG@10**		
LETOR OHSUMED	150	160	60	1
AQUAINT	200	190	80	2
WT2G	210	190	50	2
ClueWeb-2009 Category B English	120	120	90	3

Table 19 Values for different parameters obtained using the validation set for our model in Fold 5.

Datasets	**Topics** (K)		C	$l_{(d,q)}(y)$
	NDCG@5	**NDCG@10**		
LETOR OHSUMED	150	200	60	2
AQUAINT	220	230	80	3
WT2G	180	250	40	1
ClueWeb-2009 Category B English	200	190	90	2

Table 20 Values for different parameters obtained using the validation set for `Variant 1` in Fold 1.

Datasets	**Topics** (K)		C	$l_{(d,q)}(y)$
	NDCG@5	**NDCG@10**		
LETOR OHSUMED	90	210	70	1
AQUAINT	210	210	90	2
WT2G	270	120	60	3
ClueWeb-2009 Category B English	200	150	40	2

Table 21 Values for different parameters obtained using the validation set for `Variant 1` in Fold 2.

Datasets	**Topics** (K)		C	$l_{(d,q)}(y)$
	NDCG@5	**NDCG@10**		
LETOR OHSUMED	160	160	50	2
AQUAINT	190	250	70	1
WT2G	120	160	30	2
ClueWeb-2009 Category B English	150	200	60	3

2007], `LambdaMART` [Wu et al., 2010], `ListNet` [Cao et al., 2007b], `Random Forests` [Breiman, 2001] which is a popular pointwise learning-to-rank model. In addition, we used `Ranking SVM` [Joachims, 2002][16] and `SVM`^{MAP} [Yue et al., 2007][17]. The list of first six features in Table 13 are also used in these comparative methods as in [Nallapati, 2004] for learning (first 45 features in case of LETOR OHSUMED). Note that the seventh feature (or 46$^{\text{th}}$ in case of

[16] http://olivier.chapelle.cc/primal/ranksvm.m

[17] http://projects.yisongyue.com/svmmap/

Table 22 Values for different parameters obtained using the validation set for `Variant 1` in Fold 3.

Datasets	**Topics** (K)		C	$l_{(d,q)}(y)$
	NDCG@5	**NDCG@10**		
LETOR OHSUMED	150	120	50	1
AQUAINT	220	120	20	1
WT2G	120	160	10	2
ClueWeb-2009 Category B English	150	200	40	2

Table 23 Values for different parameters obtained using the validation set for `Variant 1` in Fold 4.

Datasets	**Topics** (K)		C	$l_{(d,q)}(y)$
	NDCG@5	**NDCG@10**		
LETOR OHSUMED	140	180	20	1
AQUAINT	240	190	30	4
WT2G	120	130	20	5
ClueWeb-2009 Category B English	200	150	20	3

Table 24 Values for different parameters obtained using the validation set for `Variant 1` in Fold 5.

Datasets	**Topics** (K)		C	$l_{(d,q)}(y)$
	NDCG@5	**NDCG@10**		
LETOR OHSUMED	120	210	40	3
AQUAINT	220	230	20	4
WT2G	120	240	30	5
ClueWeb-2009 Category B English	220	200	20	2

Table 25 Values for different parameters obtained using the validation set for `Variant 2` in Fold 1.

Datasets	**Topics** (K)		C	$l_{(d,q)}(y)$
	NDCG@5	**NDCG@10**		
LETOR OHSUMED	100	250	40	1
AQUAINT	240	250	60	4
WT2G	220	220	50	5
ClueWeb-2009 Category B English	210	250	30	2

LETOR OHSUMED) involves latent topic information which cannot be used in the comparative methods. In order to conduct the experiments for the comparative learning-to-rank models, we followed standard learning-to-rank experimental procedures for each comparative method. Some models have standard

Table 26 Values for different parameters obtained using the validation set for `Variant 2` in Fold 2.

Datasets	**Topics** (K)		C	$l_{(d,q)}(y)$
	NDCG@5	**NDCG@10**		
LETOR OHSUMED	180	190	30	2
AQUAINT	200	220	50	1
WT2G	180	160	20	3
ClueWeb-2009 Category B English	150	240	40	4

Table 27 Values for different parameters obtained using the validation set for `Variant 2` in Fold 3.

Datasets	**Topics** (K)		C	$l_{(d,q)}(y)$
	NDCG@5	**NDCG@10**		
LETOR OHSUMED	250	200	40	2
AQUAINT	210	150	20	3
WT2G	220	170	20	2
ClueWeb-2009 Category B English	140	200	40	2

Table 28 Values for different parameters obtained using the validation set for `Variant 2` in Fold 4.

Datasets	**Topics** (K)		C	$l_{(d,q)}(y)$
	NDCG@5	**NDCG@10**		
LETOR OHSUMED	180	120	20	1
AQUAINT	250	180	30	2
WT2G	130	230	20	2
ClueWeb-2009 Category B English	220	210	20	1

Table 29 Values for different parameters obtained using the validation set for `Variant 2` in Fold 5.

Datasets	**Topics** (K)		C	$l_{(d,q)}(y)$
	NDCG@5	**NDCG@10**		
LETOR OHSUMED	150	210	40	2
AQUAINT	210	220	20	2
WT2G	220	130	30	1
ClueWeb-2009 Category B English	180	160	20	2

published parameter values, for example, for LETOR OHSUMED, the values for `Ranking SVM`[18] and `SVM`^{MAP}[19] are online.

We present detailed parameter settings obtained from the validation dataset in each fold for our model in Tables 15, 16, 17, 18, 19. In addition, we also

[18] http://research.microsoft.com/en-us/um/beijing/projects/letor/baselines/ranksvm-primal.html

[19] http://www.yisongyue.com/results/svmmap_letor3/details.html

Table 30 NDCG@5 and NDCG@10 values for different models in the LETOR OHSUMED dataset.

Models	Performance Comparison	
	NDCG@5	**NDCG@10**
Our Model	**0.483**	**0.461**
Variant 1	0.479	0.460
Variant 2	0.478	0.460
MART	0.420	0.403
RankNet	0.471	0.455
RankBoost	0.454	0.446
AdaRank	0.469	0.445
Coordinate Ascent	0.472	0.455
LambdaRank	0.454	0.451
ListNet	0.443	0.441
Random Forests	0.434	0.431
Ranking SVM	0.461	0.454
LambdaMART	0.447	0.437
SVM-MAP	0.475	0.454

present parameter settings for our Variant 1 and Variant 2 models in Tables 20, 21, 22, 23, 24, and Tables 25, 26, 27, 28, and 29, respectively.

Note that we do not choose any unsupervised topic model for comparison primarily because they cannot make use of relevance judgment information during the training process. Thus they are always at disadvantages when compared with the learning-to-rank methods and our model, which explicitly uses the information of relevance labels during the training process. Also, supervised topic models such as sLDA cannot be directly used for comparison as one needs to make significant changes to this model to handle the document retrieval learning problem. In addition, the learning-to-rank models have already shown state-of-the-art results in this task, and thus they can be regarded as strong comparative methods. Our model does not directly use word proximity features in the learning setup [MacDonald et al., 2013]. What our model does is to use word order for finding the best model to fit the data as it has been shown in the literature that topic models with word order improve model selection [Jameel and Lam, 2013b], [Kawamae, 2014]. Such proximity features have indeed helped improve the learning-to-rank performance, but in this work our objective is to present the robustness of our model.

7.2 Quantitative Results

We present results obtained from all the test collections in Tables 30, 31, 32, and 33. From the results, we can see that our model outperforms all the comparative methods. The improvements that we obtain are statistically significant according to Wilcoxon signed rank test (with 95% confidence) against each of the comparative methods in on all the datasets except NDCG@5 in ClueWeb-2009 dataset where Variant 2 has also done better. Our results show

Table 31 NDCG@5 and NDCG@10 values for different models in the AQUAINT dataset.

Models	Performance Comparison	
	NDCG@5	**NDCG@10**
`Our Model`	**0.454**	**0.460**
`Variant 1`	0.450	0.452
`Variant 2`	0.451	0.455
`MART`	0.400	0.405
`RankNet`	0.444	0.451
`RankBoost`	0.431	0.438
`AdaRank`	0.443	0.449
`Coordinate Ascent`	0.442	0.448
`LambdaRank`	0.431	0.438
`ListNet`	0.443	0.445
`Random Forests`	0.415	0.421
`Ranking SVM`	0.434	0.433
`LambdaMART`	0.428	0.425
`SVM-MAP`	0.448	0.451

Table 32 NDCG@5 and NDCG@10 values for different models in the WT2G dataset.

Models	Performance Comparison	
	NDCG@5	**NDCG@10**
`Our Model`	**0.311**	**0.311**
`Variant 1`	0.309	0.306
`Variant 2`	0.310	0.307
`MART`	0.303	0.303
`RankNet`	0.305	0.308
`RankBoost`	0.304	0.306
`AdaRank`	0.308	0.307
`Coordinate Ascent`	0.301	0.305
`LambdaRank`	0.303	0.304
`ListNet`	0.306	0.304
`Random Forests`	0.303	0.301
`Ranking SVM`	0.304	0.305
`LambdaMART`	0.302	0.303
`SVM-MAP`	0.308	0.308

that the latent topic information generated by our model which is then used to compute query-document similarity plays a significant role. Word order too plays a role where we are able to detect better topics than unigram models.

In the OHSUMED collection, we find that our main proposed model in which word order is maintained in both queries and documents performs better than other models. Looking closely at NDCG@5 results, we can see that our model performs considerably better with statistically significant results than comparative models. `Variant 2` does not perform better than `Variant 1` at NDCG@5, thereby bringing out the importance of word order in retrieval learning task. However, models such as `SVM-MAP` and `RankNet` also do better in this collection. The reason is mainly due to the mechanism of these models, which optimize a different objective function. `Coordinate Ascent` model

Table 33 NDCG@5 and NDCG@10 values for different models in the ClueWeb-2009 Category B English dataset.

Models	Performance Comparison	
	NDCG@5	NDCG@10
Our Model	**0.369**	**0.360**
Variant 1	0.366	0.356
Variant 2	**0.369**	0.359
MART	0.334	0.341
RankNet	0.366	0.356
RankBoost	0.358	0.354
AdaRank	0.354	0.351
Coordinate Ascent	0.350	0.352
LambdaRank	0.359	0.354
ListNet	0.367	0.356
Random Forests	0.353	0.351
Ranking SVM	0.359	0.352
LambdaMART	0.350	0.352
SVM-MAP	0.367	0.358

also performs better, but does not outperform our main proposed model. At NDCG@10, we see improvement in Variant 1 and Variant 2 models where we can see that the performance gap has narrowed, but they still do not outperform out model. However, the improvement of our model is still statistically significant. Other models such as Ranking SVM, Coordinate Ascent, RankNet, and SVM-MAP also perform better in this dataset. In AQUAINT collection, we notice consistent superior performance of our model when compared with comparative models, with improvements that are statistically significant. We also find that gap between the performance of our model when compared with Variant 2 especially at NDCG@5 is also reduced. Models such as SVM-MAP and RankNet also perform better in this dataset. Also, we can see that the difference between Variant 1 and Variant 2 is not much in this dataset. We see some interesting results in WT2G dataset. Many models do better in this dataset and are quite close in performance when compared with our model especially at NDCG@5. At NDCG@10, our model consistently does better. But in ClueWeb-2009 dataset, we can see that Variant 2 matches the performance of our model. Even at NDCG@10, many models are close to our model in performance. This suggests that spam and noisy pages have some impact on our model. Also, we can conclude that maintaining word order may not be a good way to model those collections which have noisy documents. The bag-of-words model can also do better in noisy collections.

We have seen from the results obtained in this experiments that considering order of words in both queries and documents simultaneously, helps improve the performance of document retrieval learning using topic models, and relaxing the order of words either queries or documents does not help in improving the results. The reason for good performance is primarily because our model is able to capture the semantic dependencies in text and matches words based on word proximity. We also found that noise has an impact on our model.

Table 34 NDCG@5 (denoted as N@5), and NDCG@10 (denoted as N@10) results obtained from our model when we vary the number of topics from 50 to 290.

	OHSUMED		**AQUAINT**		**WT2G**		**ClueWeb**	
Topics (K)	N@5	N@10	N@5	N@10	N@5	N@10	N@5	N@10
50	0.480	0.460	0.450	0.454	0.310	0.309	0.365	0.359
70	0.480	0.461	0.451	0.455	0.310	0.308	0.364	0.354
90	0.482	0.461	0.451	0.455	0.311	0.310	0.365	0.358
110	0.483	0.461	0.452	0.458	0.310	0.310	0.366	0.353
130	0.483	0.461	0.451	0.457	0.311	0.309	0.368	0.358
150	0.483	0.461	0.453	0.455	0.310	0.310	0.369	0.359
170	0.482	0.461	0.453	0.456	0.311	0.311	0.369	0.360
190	0.481	0.461	0.452	0.458	0.311	0.311	0.369	0.360
210	0.481	0.460	0.454	0.459	0.311	0.311	0.369	0.360
230	0.481	0.461	0.454	0.460	0.310	0.309	0.368	0.359
270	0.480	0.461	0.453	0.460	0.310	0.310	0.369	0.359
290	0.482	0.460	0.451	0.459	0.311	0.311	0.368	0.360

Therefore, it can be concluded that in collections which are very noisy and contain many spam pages, the bag-of-words model can also be adopted.

One interesting facet to consider is to study the effect of the number of topics in the document retrieval learning experiment for our models. In order to study the effect on the number of topics, we varied the number of topics in the training set in each fold. We used the same set of parameters obtained in each fold in each dataset as we have shown earlier except the number of topics which we specify manually in this set of experiments. After training the model on the training set, we used the test set directly to find the effect of the number of topics. We present results by averaging results obtained from all five folds. In Table 34, we vary the number of topics from 50 to 290 in steps of 20 and present the results therein for our model. In the OHSUMED dataset we can see that as we increase the number of topics, the results improve until certain number of topics and begin to deteriorate again as we keep on increasing the number of topics. This gives us an insight about the dependence between the number of topics and the retrieval learning results for our models. But we do not find any noticeable pattern when the number of topics is varied. What we do observe is that the effect when the number of topics is varied is not huge. Most of the values appear very close to each other in all datasets.

In addition, we also present results obtained from `Variant 1` in Table 35 in different datasets. We can observe that effect of topics is not very noticeable in this model also. We have similar observation in Table 36.

It is quite interesting to see that our model outperforms some of the powerful learning-to-rank models. Our model can perform consistently well with more (in LETOR OHSUMED) and less number of features (in WT2G and AQUAINT). This shows that the generalization ability of our proposed model is very robust. The results suggest that incorporating topic similarity helps improve document retrieval performance. One reason why topic models help improve document retrieval performance as we compare the similarity between the document and the query based on latent factors rather than just the words

Table 35 NDCG@5 (denoted as N@5), and NDCG@10 (denoted as N@10) results obtained from `Variant 1` when we vary the number of topics from 50 to 290.

	OHSUMED		**AQUAINT**		**WT2G**		**ClueWeb**	
Topics (K)	N@5	N@10	N@5	N@10	N@5	N@10	N@5	N@10
50	0.479	0.460	0.444	0.451	0.306	0.304	0.360	0.352
70	0.479	0.459	0.440	0.452	0.308	0.305	0.362	0.354
90	0.478	0.459	0.445	0.450	0.309	0.304	0.363	0.353
110	0.478	0.459	0.450	0.448	0.309	0.305	0.364	0.352
130	0.479	0.460	0.448	0.451	0.309	0.306	0.365	0.354
150	0.479	0.460	0.449	0.450	0.309	0.306	0.366	0.354
170	0.479	0.460	0.448	0.451	0.308	0.306	0.366	0.356
190	0.478	0.460	0.450	0.452	0.307	0.305	0.366	0.356
210	0.478	0.459	0.450	0.452	0.308	0.306	0.366	0.356
230	0.478	0.459	0.450	0.452	0.306	0.306	0.366	0.356
270	0.479	0.460	0.446	0.451	0.309	0.304	0.365	0.355
290	0.479	0.458	0.448	0.451	0.308	0.305	0.366	0.354

Table 36 NDCG@5 (denoted as N@5), and NDCG@10 (denoted as N@10) results obtained from `Variant 2` when we vary the number of topics from 50 to 290.

	OHSUMED		**AQUAINT**		**WT2G**		**ClueWeb**	
Topics (K)	N@5	N@10	N@5	N@10	N@5	N@10	N@5	N@10
50	0.475	0.455	0.446	0.451	0.309	0.306	0.365	0.358
70	0.476	0.456	0.451	0.451	0.310	0.305	0.364	0.359
90	0.470	0.458	0.450	0.453	0.308	0.306	0.365	0.356
110	0.471	0.456	0.451	0.454	0.310	0.306	0.366	0.358
130	0.473	0.455	0.450	0.455	0.309	0.306	0.368	0.359
150	0.475	0.458	0.449	0.455	0.310	0.305	0.369	0.359
170	0.478	0.460	0.451	0.453	0.309	0.304	0.369	0.356
190	0.478	0.460	0.450	0.454	0.310	0.306	0.368	0.358
210	0.478	0.460	0.451	0.455	0.310	0.304	0.368	0.355
230	0.473	0.458	0.449	0.455	0.309	0.305	0.369	0.359
270	0.475	0.460	0.449	0.454	0.309	0.306	0.369	0.354
290	0.470	0.460	0.450	0.455	0.308	0.304	0.368	0.356

[Wei and Croft, 2006], [Sordoni et al., 2013]. Hence, this feature which our model computes is extremely important for document retrieval learning task.

7.3 Investigation on Topic Enhancements for Comparative Models

In this section, we present results where we add the latent topic feature as one of the features in addition to the existing list of features in a two stage approach. Our motivation is to study where latent topic feature obtained either from `LDA` or `BTM` can help improve the performance of the comparative models. Results of our model and its variants will remain the same as shown the previous experiment described in Section 7.2.

Table 37 NDCG@5 and NDCG@10 values for different models in the LETOR OHSUMED dataset when the comparative models are enhanced with latent topic feature obtained from the `LDA` model.

Models	Performance Comparison	
	NDCG@5	**NDCG@10**
`Our Model`	**0.483**	**0.461**
`Variant 1`	0.479	0.460
`Variant 2`	0.478	0.460
`MART`	0.423	0.406
`RankNet`	0.476	0.458
`RankBoost`	0.459	0.451
`AdaRank`	0.471	0.453
`Coordinate Ascent`	0.472	0.459
`LambdaRank`	0.458	0.455
`ListNet`	0.462	0.455
`Random Forests`	0.442	0.439
`Ranking SVM`	0.462	0.456
`LambdaMART`	0.458	0.446
`SVM-MAP`	0.478	0.456

7.3.1 Employing LDA

In this set of experiments, for all the comparative methods, we manually append a latent topic similarity feature. The procedure is to first conduct latent topic modeling using the `LDA` model on the set of documents used in the learning-to-rank experiments. Then we use an existing method described in [Wei and Croft, 2006] to compute the query-document topic similarity. We obtain a score for each number of latent topic (K) which we vary from 10 to 100. Then we create the training, test and validation datasets based on the same split as used in the previous experiment. We use the validation set to train the parameters of the comparative models. We obtain the best topic K from the validation set which gives the best NDCG@5 and NDCG@10 across all topics in the validation set.

We present results for this set of experiments on different datasets in Tables 37, 38, 39 and 40. This topic enhanced setting is used in the comparative methods only.

Our results show that even by manually adding the latent topic feature computed externally, the comparative methods cannot outperform our proposed model. From the results in all datasets, we can make a conclusion that in majority of the cases the results of the comparative methods have improved by adding the latent topic similarity feature. But the results could not outperform our proposed document retrieval learning model. The reason lies in the inherent design of the model where it is embedded with the latent topic model and maximum margin prediction. Even the closest learning-to-rank model `Ranking SVM` could not outperform our model.

The improvements that we obtain are statistically significant according to Wilcoxon signed rank test (with 95% confidence) against each of the compara-

Table 38 NDCG@5 and NDCG@10 values for different models in the AQUAINT dataset when the comparative models are enhanced with latent topic feature obtained from the LDA model.

Models	Performance Comparison	
	NDCG@5	**NDCG@10**
Our Model	**0.454**	**0.460**
Variant 1	0.450	0.452
Variant 2	0.451	0.455
MART	0.421	0.418
RankNet	0.448	0.451
RankBoost	0.439	0.443
AdaRank	0.445	0.449
Coordinate Ascent	0.449	0.448
LambdaRank	0.439	0.441
ListNet	0.446	0.448
Random Forests	0.434	0.429
Ranking SVM	0.435	0.433
LambdaMART	0.428	0.424
SVM-MAP	0.450	0.452

Table 39 NDCG@5 and NDCG@10 values for different models in the WT2G dataset when the comparative models are enhanced with latent topic feature obtained from the LDA model.

Models	Performance Comparison	
	NDCG@5	**NDCG@10**
Our Model	**0.311**	**0.311**
Variant 1	0.309	0.306
Variant 2	0.310	0.307
MART	0.303	0.304
RankNet	0.307	0.309
RankBoost	0.305	0.306
AdaRank	0.309	0.307
Coordinate Ascent	0.303	0.305
LambdaRank	0.306	0.303
ListNet	0.305	0.305
Random Forests	0.305	0.305
Ranking SVM	0.305	0.306
LambdaMART	0.302	0.304
SVM-MAP	0.309	0.309

tive methods in all the datasets except NDCG@5 in ClueWeb-2009 dataset. We can notice from that the comparative methods have improved when the latent topic feature is added. In terms of performance, the gap between the comparative methods and our model has also reduced. In LETOR OHSUMED dataset, SVM-MAP and Coordinate Ascent models perform better. In ClueWeb-2009 dataset, most of the models are able to narrow the performance gap, but our model still remains competitive.

Another interesting note is the length of the query and the performance of our model. We have noticed that for longer queries our model performs

Table 40 NDCG@5 and NDCG@10 values for different models in the ClueWeb-2009 Category B English dataset when the comparative models are enhanced with latent topic feature obtained from the `LDA` model.

Models	Performance Comparison	
	NDCG@5	**NDCG@10**
`Our Model`	**0.369**	**0.360**
`Variant 1`	0.366	0.356
`Variant 2`	**0.369**	0.359
`MART`	0.336	0.345
`RankNet`	0.368	0.358
`RankBoost`	0.360	0.356
`AdaRank`	0.356	0.351
`Coordinate Ascent`	0.354	0.354
`LambdaRank`	0.360	0.355
`ListNet`	0.368	0.359
`Random Forests`	0.354	0.353
`Ranking SVM`	0.360	0.355
`LambdaMART`	0.351	0.353
`SVM-MAP`	0.368	0.359

relatively better as compared to shorter queries. The reason may be due to the fact that the word order can convey more information to our model for longer queries as compared to shorter queries.

7.3.2 Employing BTM

In this set of experiments, instead of using the `LDA` model, we use the `BTM` model which considers word order. The procedure for adding latent topic information is similar to that described in Section 7.3.1, except that the retrieval formulation using language modeling technique needs to be changed a bit in order to incorporate word order. We present the retrieval formulations below.

The query likelihood model scoring for each document d is done by calculating the likelihood of its model in generating a query q. This can be written as $P_{\mathrm{LM}}(q|d)$. Under the bag-of-words assumption, we can write the following likelihood function:

$$P_{\mathrm{LM}}(q|d) = \prod_{i=1}^{N^q} P(q_i|d) \tag{29}$$

The above equation (Equation 29) is specified by a document model where we can consider Dirichlet smoothing [Zhai and Lafferty, 2004]. Therefore, Equation 29 can be expressed as:

$$P_{\mathrm{LM}}(q|d) = \frac{N^d}{N^d+\mu} P_{\mathrm{ML}}(q|d) + \left(1 - \frac{N^d}{N^d+\mu}\right) P_{\mathrm{ML}}(q|\mathbb{D}) \tag{30}$$

where $P_{\mathrm{LM}}(q|d)$ is the maximum likelihood estimate for the query q generated in the document d. $P_{\mathrm{ML}}(q|\mathbb{D})$ is the maximum likelihood estimate for the query

Table 41 NDCG@5 and NDCG@10 values for different models in the LETOR OHSUMED dataset when the comparative models are enhanced with latent topic feature obtained from the BTM model.

Models	Performance Comparison	
	NDCG@5	**NDCG@10**
Our Model	**0.483**	**0.461**
Variant 1	0.479	0.460
Variant 2	0.478	0.460
MART	0.431	0.409
RankNet	0.478	0.459
RankBoost	0.462	0.458
AdaRank	0.474	0.455
Coordinate Ascent	0.476	0.460
LambdaRank	0.466	0.456
ListNet	0.460	0.455
Random Forests	0.451	0.445
Ranking SVM	0.469	0.459
LambdaMART	0.458	0.447
SVM-MAP	0.478	0.459

q generated in the entire collection $\mathbb{D}$. $\mu = 1000$ is the smoothing prior. This prior value has been adopted from the work of [Zhai and Lafferty, 2004].

In order to calculate the query likelihood for the BTM model using the language modeling framework, we need to sum over all the topic variables for each word. The posterior estimates can be used in the likelihood model. The query likelihood for the query q given the document d from BTM is written as $P_{\text{BTM}}(q|d)$. Therefore, the likelihood function can be written as:

$$P_{\text{BTM}}(q|d) = \prod_{i=1}^{N^q} P_{\text{BTM}}(q_i|q_{i-1}, d) \tag{31}$$

where $P_{\text{BTM}}(q_i|q_{i-1}, d)$ can be expressed as:

$$P_{\text{BTM}}(q_i|q_{i-1}, d) = \sum_{k_i=1}^{K} P(q_i|\boldsymbol{\Phi}_{k_i}, q_{i-1}) P(k_i|\boldsymbol{\theta}^d) \tag{32}$$

Similar to the framework described in [Wei and Croft, 2006], we can adopt the following:

$$P(q|d) = \lambda P_{\text{LM}}(q|d) + (1 - \lambda) P_{\text{BTM}}(q|d) \tag{33}$$

where λ is a weighting parameter. For consistency in the experiments performed using the LDA model in Section 7.3.1, we set the value of $\lambda = 0.7$.

We present the results obtained by adding the topic information using BTM in Tables 41, 42, 43, and 44. In all our experiments, the improvement shown by our model is statistically significant according to Wilcoxon signed rank test (with 95% confidence) against each of the comparative methods in all the datasets except NDCG@5 in ClueWeb-2009 dataset.

Table 42 NDCG@5 and NDCG@10 values for different models in the AQUAINT dataset when the comparative models are enhanced with latent topic feature obtained from the `BTM` model.

Models	Performance Comparison	
	NDCG@5	**NDCG@10**
`Our Model`	**0.454**	**0.460**
`Variant 1`	0.450	0.452
`Variant 2`	0.451	0.455
`MART`	0.418	0.423
`RankNet`	0.449	0.452
`RankBoost`	0.442	0.449
`AdaRank`	0.448	0.451
`Coordinate Ascent`	0.448	0.446
`LambdaRank`	0.440	0.441
`ListNet`	0.446	0.449
`Random Forests`	0.441	0.433
`Ranking SVM`	0.436	0.448
`LambdaMART`	0.430	0.433
`SVM-MAP`	0.450	0.453

Table 43 NDCG@5 and NDCG@10 values for different models in the WT2G dataset when the comparative models are enhanced with latent topic feature obtained from the `BTM` model.

Models	Performance Comparison	
	NDCG@5	**NDCG@10**
`Our Model`	**0.311**	**0.311**
`Variant 1`	0.309	0.306
`Variant 2`	0.310	0.307
`MART`	0.305	0.305
`RankNet`	0.308	0.309
`RankBoost`	0.308	0.308
`AdaRank`	0.309	0.307
`Coordinate Ascent`	0.306	0.308
`LambdaRank`	0.305	0.304
`ListNet`	0.308	0.307
`Random Forests`	0.306	0.306
`Ranking SVM`	0.309	0.308
`LambdaMART`	0.305	0.306
`SVM-MAP`	0.310	0.308

In the OHSUMED dataset as depicted in Table 41, we can notice that our model still remains competitive compared with other models. We achieve very good performance at NDCG@5, but the other models also do very well at NDCG@10. When compared to the results obtained using the `LDA` model as depicted in Table 37 i.e. when latent topic information obtained from the `LDA` model is used, we can see that indeed performance (when compared to the results in Table 37) of comparative models has improved when word order is maintained in the topic model, and that topic feature is used in the learning-to-rank models. Looking more closely, we notice that at NDCG@5,

Table 44 NDCG@5 and NDCG@10 values for different models in the ClueWeb-2009 Category B English dataset when the comparative models are enhanced with latent topic feature obtained from the BTM model.

Models	Performance Comparison	
	NDCG@5	NDCG@10
Our Model	**0.369**	**0.360**
Variant 1	0.366	0.356
Variant 2	**0.369**	0.359
MART	0.336	0.346
RankNet	0.367	0.358
RankBoost	0.361	0.356
AdaRank	0.355	0.350
Coordinate Ascent	0.351	0.354
LambdaRank	0.363	0.358
ListNet	0.368	0.359
Random Forests	0.356	0.359
Ranking SVM	0.363	0.356
LambdaMART	0.353	0.355
SVM-MAP	0.368	0.359

most of the comparative models have shown improved performance except LambdaMART, ListNet, and SVM-MAP. In fact, the performance of ListNet and LambdaMART have actually deteriorated to some extent suggesting that latent topic information with word order did not give much help to the model. Even at NDCG@10, ListNet could recover from its poor performance, but not SVM-MAP and LambdaMART. We also notice that at NDCG@10, in Table 41, gap between our model and comparative models has lessened. In AQUAINT as depicted in Table 42, we notice that our model has performed better than comparative models. At NDCG@5, we notice that performance of three models has deteriorated as compared to that in LDA as depicted in Table 38. These models are MART, Coordinate Ascent, and SVM-MAP. But the change in results is not very significant. At NDCG@10, for AQUAINT as depicted in Table 42, we notice that MART and SVM-MAP show an improvement when compared to LDA as depicted in Table 38. In addition, the performance of LambdaRank has deteriorated when latent topic information with word order is added to the model at NDCG@10. In WT2G as depicted in Table 43, we notice good improvement in the comparative models when compared to that in LDA as depicted in Table 39 at both NDCG@5 and NDCG@10. But the performance of these models is not good when compared with our model. LambdaRank, at NDCG@5, does not show an improvement when latent topic from BTM is added to the list of features. Similarly, RankNet shows no such improvement. In ClueWeb09 collection as depicted in Table 44, at NDCG@5, many models have in fact shown lowering of NDCG@5 results, suggesting that spam and noisy text is having some impact on the results. Models such as RankNet, AdaRank, Coordinate Ascent have in fact deteriorated when compared with results listed in Table 40. Models such as ListNet and SVM-MAP show no change in performance.

At NDCG@10, RankBoost, Coordinate Ascent, and SVM-MAP show no performance improvement. AdaRank performance has in fact deteriorated.

From the above results, in general, they reveal that by incorporating latent topic information using word order in the comparative learning-to-rank methods does help improve performance. But since the approach is two stage, the comparative models are not able to do better than our proposed model. We can conclude that word order has helped improve the performance to some extent, but it is not consistent in all our results.

7.4 Topical Words Examples

Table 45 Top five probable words from a topic from AQUAINT collection.

BTM	LDACOL	TNG
foreign beggars	today	news corp
bt anton	hebron	www
hk salem	bosnian	web
fundamental prerequisites	foreign beggars	news event
great stash	atlanta	york steaks

Table 46 Top five probable words from a topic from AQUAINT collection.

PDLDA	NTSeg	Our Model
foreign minister stevo	today	news viewership
fundamental prerequisites	atlanta	foreign minister
jewish state restarts	hk salem	president nasser
reported exceptionally	york times news service	general news
york times news service	bosnia	resistance occurred

We can see from Tables 45 and 46 that our model has generated words which appear more meaningful than the other models. From the list of top five words, it can be noted that our model is describing about "Egypt" and the news related to the revolution during that time. We have only considered words from documents in order to present results in this table. AQUAINT collection does not have documents indexed in different classes just like those we have used in classification experiments, therefore supervised topic models such as MedLDA, etc. might not generate interpretable words in topics as they cannot use an extra side-information while learning. Therefore, for this comparison, we have only considered unsupervised n-gram topic models. Our model uses query-document relevance label (during learning) for generating words. We can see that words such as "president nasser" and "foreign minister" are more insightful in comparison to the words such as "hk salem" and "today" generated by the

NTSeg model. Much research has already been done in topic models with word order where it has been shown empirically that n-gram models generate more interpretable latent topics than unigram models [Lindsey et al., 2012], [Jameel and Lam, 2013b], [Jameel and Lam, 2013c], [Wang et al., 2007], [Griffiths et al., 2007]. But what those n-gram models fail to consider side-information which can help generate even better latent topical representations. We have shown empirically that our model has generated more meaningful latent topic models than comparative models.

8 Conclusions

We have presented supervised topic models which maintain word order in the document. We first propose a bigram supervised topic model with maximum margin framework, and compare the performance of the model with comparative methods. From the empirical analysis, we demonstrate that our model outperforms many comparative methods. We then extend the supervised bigram topic model to handle document retrieval learning task. This model takes as input the query-document pairs. Relevance assessments given manually by annotators are the response variables. The experimental analysis shows that our model outperforms many popular learning-to-rank models. By presenting a list of topical words in topics we showed how our model generates better topical words than the comparative methods. Results clearly show that learning with side-information helps the model generate more interpretable topics with words that are insightful to a reader.

A Proof

From Equation 2, based on the formula of Bayes' Theorem, we can deduce that $P(\boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi}|\boldsymbol{W}, \boldsymbol{\alpha}, \boldsymbol{\beta})$ is the posterior distribution that needs to be found out. $P_0(\boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi}|\boldsymbol{\alpha}, \boldsymbol{\beta})$ is the prior distribution. $P(\boldsymbol{W}|\boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi})$ is the likelihood, and the denominator $P(\boldsymbol{W}|\boldsymbol{\alpha}, \boldsymbol{\beta})$ is the marginal distribution over data.

The Kullback-Leibler Divergence (KL) from a distribution p to a distribution q can be written as $\mathrm{KL}(q||p)$. Suppose we consider an arbitrary distribution $Q(\boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi}|\boldsymbol{W}, \boldsymbol{\alpha}, \boldsymbol{\beta})$. Our goal is to ensure that this distribution is equal to the posterior distribution $P(\boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi}|\boldsymbol{W}, \boldsymbol{\alpha}, \boldsymbol{\beta})$. As in the Bayes' rule, this posterior is obtained by iteratively updating the prior $P_0(\boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi}|\boldsymbol{\alpha}, \boldsymbol{\beta})$.

Suppose we want to minimize the divergence between the arbitrary distribution and the posterior distribution, and this is what we want to achieve so that the two distributions are as close as possible or equal to each other i.e. they overlap. We can write the statement mathematically as:

$$\underset{Q(\boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi}) \in \mathbb{P}}{\text{minimize}} \ \mathrm{KL}[Q(\boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi}|\boldsymbol{\alpha}, \boldsymbol{\beta})||P(\boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi}|\boldsymbol{\alpha}, \boldsymbol{\beta})] \tag{34}$$

We know from Equation 2 that:

$$P(\boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi}|\boldsymbol{W}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \frac{P_0(\boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi}|\boldsymbol{\alpha}, \boldsymbol{\beta}) P(\boldsymbol{W}|\boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi})}{P(\boldsymbol{W}|\boldsymbol{\alpha}, \boldsymbol{\beta})} \tag{35}$$

For Equation 34, we substitute $P(\boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi}|\boldsymbol{W}, \boldsymbol{\alpha}, \boldsymbol{\beta})$ by replacing Equation 35:

$$\underset{Q(\boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi}) \in \mathbb{P}}{\text{minimize}} \ \mathrm{KL}\left[Q(\boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi}|\boldsymbol{\alpha}, \boldsymbol{\beta})||\frac{P_0(\boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi}|\boldsymbol{\alpha}, \boldsymbol{\beta}) P(\boldsymbol{W}|\boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi})}{P(\boldsymbol{W}|\boldsymbol{\alpha}, \boldsymbol{\beta})}\right] \tag{36}$$

We know that the Kullback-Leibler distance is the expectation of the difference in logarithms of their probability density functions. In terms of expectation, Equation 36 can be equivalently can be written as:

$$\mathbb{E}_Q \left[\log \frac{Q(\boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi}|\boldsymbol{\alpha}, \boldsymbol{\beta})}{\frac{P_0(\boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi}|\boldsymbol{\alpha}, \boldsymbol{\beta}) P(\boldsymbol{W}|\boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi})}{P(\boldsymbol{W}|\boldsymbol{\alpha}, \boldsymbol{\beta})}} \right] \tag{37}$$

Equation 37 can be further written as:

$$\mathbb{E}_Q \left[\log \frac{Q(\boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi}|\boldsymbol{\alpha}, \boldsymbol{\beta})}{P_0(\boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi}|\boldsymbol{\alpha}, \boldsymbol{\beta})} - \log P(\boldsymbol{W}|\boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi}) + \log P(\boldsymbol{W}|\boldsymbol{\alpha}, \boldsymbol{\beta}) \right] \tag{38}$$

This now simplifies to:

$$\underset{Q(\boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi}) \in \mathbb{P}}{\text{minimize}} \; \text{KL}[Q(\boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi}|\boldsymbol{\alpha}, \boldsymbol{\beta})||P_0(\boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi}|\boldsymbol{\alpha}, \boldsymbol{\beta})] - \mathbb{E}_{\mathbb{Q}}[\log P(\boldsymbol{W}|\boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi})] + \log P(\boldsymbol{W}|\boldsymbol{\alpha}, \boldsymbol{\beta}) \tag{39}$$

The last term in Equation 39 can be removed because it does not depend on $\boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi}$. As a result, we get:

$$\underset{Q(\boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi}) \in \mathbb{P}}{\text{minimize}} \; \text{KL}[Q(\boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi}|\boldsymbol{\alpha}, \boldsymbol{\beta})||P_0(\boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi}|\boldsymbol{\alpha}, \boldsymbol{\beta})] - \mathbb{E}_Q[\log P(\boldsymbol{W}|\boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi})] \tag{40}$$

References

Acharya, A., Rawal, A., Mooney, R. J., and Hruschka, E. R. (2013). Using both latent and supervised shared topics for multitask learning. In *Machine Learning and Knowledge Discovery in Databases*, pages 369–384.

Aldous, D. (1985). Exchangeability and related topics. *École d'Été de Probabilités de Saint-Flour XIIIâĂŤ1983*, 1117:1–198.

Allan, J. (2005). HARD track overview in TREC 2003 High Accuracy Retrieval from Documents. Technical report, DTIC Document.

Andrzejewski, D. and Buttler, D. (2011). Latent topic feedback for Information Retrieval. In *Knowledge Discovery and Data Mining*, pages 600–608.

Asadi, N. and Lin, J. (2013). Effectiveness/efficiency tradeoffs for candidate generation in multi-stage retrieval architectures. In *Special Interest Group on Information Retrieval*, pages 997–1000.

Bai, B., Weston, J., Grangier, D., Collobert, R., Sadamasa, K., Qi, Y., Chapelle, O., and Weinberger, K. (2010). Learning to rank with (a lot of) word features. *Information Retrieval*, 13(3):291–314.

Bartlett, N., Pfau, D., and Wood, F. (2010). Forgetting counts : Constant memory inference for a dependent Hierarchical Pitman-Yor process. In *International Conference on Machine Learning*, pages 63–70.

Bicego, M., Lovato, P., Oliboni, B., and Perina, A. (2010). Expression microarray classification using topic models. In *ACM Symposium on Applied Computing*, pages 1516–1520.

Blei, D. and McAuliffe, J. (2008). Supervised topic models. In *Neural Information Processing Systems*, pages 121–128.

Blei, D. M. (2012). Probabilistic topic models. *Communications of the ACM*, 55(4):77–84.

Blei, D. M. and Lafferty, J. D. (2009). Topic models. *Text mining: classification, clustering, and applications*, 10:71.

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2001). Latent Dirichlet allocation. In *Neural Information Processing Systems*, pages 601–608.

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent Dirichlet allocation. *Journal of Machine Learning Research (JMLR)*, 3:993–1022.

Breiman, L. (2001). Random forests. *Machine Learning*, 45(1):5–32.

Brin, S. and Page, L. (1998). The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 30(1):107–117.

Broder, A. (2002). A taxonomy of Web search. In *ACM Special Interest Group on Information Retrieval Forum*, volume 36, pages 3–10.

Burges, C., Shaked, T., Renshaw, E., Lazier, A., Deeds, M., Hamilton, N., and Hullender, G. (2005). Learning to rank using gradient descent. In *International Conference on Machine Learning*, pages 89–96.

Burges, C. J. (1998). A tutorial on Support Vector Machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167.

Busa-Fekete, R., Kégl, B., Éltető, T., and Szarvas, G. (2013). Tune and mix: learning to rank using ensembles of calibrated multi-class classifiers. *Machine Learning*, 93(2-3):261–292.

Cai, P., Gao, W., Zhou, A., and Wong, K.-F. (2011). Relevant knowledge helps in choosing right teacher: active query selection for ranking adaptation. In *Special Interest Group on Information Retrieval*, pages 115–124.

Cao, J., Li, J., Zhang, Y., and Tang, S. (2007a). LDA-based retrieval framework for semantic news video retrieval. In *International Conference on Semantic Computing*, pages 155–160.

Cao, Z., Qin, T., Liu, T.-Y., Tsai, M.-F., and Li, H. (2007b). Learning to rank: from pairwise approach to listwise approach. In *International Conference on Machine Learning*, pages 129–136.

Chang, J. and Blei, D. M. (2009). Relational topic models for document networks. In *International Conference on Artificial Intelligence and Statistics*, pages 81–88.

Chen, B. (2009). Word topic models for spoken document retrieval and transcription. *ACM Transactions on Asian Language Information Processing*, 8(1):2.

Cortes, C. and Vapnik, V. (1995). Support Vector Machine. *Machine Learning*, 20(3):273–297.

Dang, V., Bendersky, M., and Croft, W. B. (2013). Two-stage learning to rank for Information Retrieval. In *European Conference on Information Retrieval*, pages 423–434.

Duan, D., Li, Y., Li, R., Zhang, R., and Wen, A. (2012). RankTopic: Ranking based topic modeling. In *International Conference on Data Mining*, pages 211–220.

Egozi, O., Markovitch, S., and Gabrilovich, E. (2011). Concept-based Information Retrieval using Explicit Semantic Analysis. *Transactions on Information Systems*, 29(2):8:1–8:34.

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, pages 1189–1232.

Ganchev, K., Graça, J., Gillenwater, J., and Taskar, B. (2010). Posterior regularization for structured latent variable models. *Journal of Machine Learning Research (JMLR)*, 11:2001–2049.

Gao, J., Toutanova, K., and Yih, W.-t. (2011). Clickthrough-based latent semantic models for Web search. In *Special Interest Group on Information Retrieval*, pages 675–684.

Gao, W. and Yang, P. (2014). Democracy is good for ranking: Towards multi-view rank learning and adaptation in web search. In *Web Search and Data Mining*, pages 63–72.

Griffiths, T., Steyvers, M., and Tenenbaum, J. (2007). Topics in semantic representation. *Psychological Review*, 114(2):211.

Hang, L. (2011). A short introduction to learning to rank. *IEICE Transactions on Information and Systems*, 94(10):1854–1862.

Hasler, E., Blunsom, P., Koehn, P., and Haddow, B. (2014). Dynamic topic adaptation for phrase-based MT. In *European Chapter of the Association for Computational Linguistics*, pages 328–337.

Hazen, T. J. (2010). Direct and latent modeling techniques for computing spoken document similarity. In *Spoken Language Technology Workshop*, pages 366–371.

Heath, D. and Sudderth, W. (1976). De Finetti's theorem on exchangeable variables. *The American Statistician*, 30(4):188–189.

Jagarlamudi, J. and Gao, J. (2013). Modeling click-through based word-pairs for Web search. In *Special Interest Group on Information Retrieval*, pages 483–492.

Jameel, S. and Lam, W. (2013a). A nonparametric n-gram topic model with interpretable latent topics. In *Asian Information Retrieval Societies Conference*, pages 74–85.

Jameel, S. and Lam, W. (2013b). An unsupervised topic segmentation model incorporating word order. In *Special Interest Group on Information Retrieval*, pages 203–212.

Jameel, S. and Lam, W. (2013c). An N-gram topic model for time-stamped documents. In *European Conference on Information Retrieval*, pages 292–304.

Jameel, S., Lam, W., and Bing, L. (2015). Nonparametric topic modeling using chinese restaurant franchise with buddy customers. In *European Conference on Information Retrieval*, volume 9022, pages 648–659.

Järvelin, K. and Kekäläinen, J. (2002). Cumulated gain-based evaluation of IR techniques. *Transactions on Information Systems*, 20(4):422–446.

Jiang, Q., Zhu, J., Sun, M., and Xing, E. P. (2012). Monte Carlo methods for maximum margin supervised topic models. In *Neural Information Processing Systems*, pages 1601–1609.

Joachims, T. (1998). Text categorization with support vector machines: Learning with many relevant features. In *European Conference on Machine Learning*, volume 1398, pages 137–142.

Joachims, T. (2002). Optimizing search engines using clickthrough data. In *Knowledge Discovery and Data Mining*, pages 133–142.

Kawamae, N. (2014). Supervised N-gram topic model. In *Web Search and Data Mining*, pages 473–482.

Lacoste-Julien, S., Sha, F., and Jordan, M. I. (2008). DiscLDA: Discriminative learning for dimensionality reduction and classification. In *Neural Information Processing Systems*, pages 897–904.

Lai, H., Pan, Y., Liu, C., Lin, L., and Wu, J. (2013). Sparse learning-to-rank via an efficient primal-dual algorithm. *IEEE Transactions on Computers,*, 62(6):1221–1233.

Lakshminarayanan, B. and Raich, R. (2011). Inference in supervised Latent Dirichlet Allocation. In *Machine Learning for Signal Processing*, pages 1–6. IEEE.

Li, H. and Xu, J. (2014). Semantic matching in search. *Foundations and Trends in Information Retrieval*, 7(5):343–469.

Li, P., Burges, C. J., Wu, Q., Platt, J., Koller, D., Singer, Y., and Roweis, S. (2007). Mcrank: Learning to rank using multiple classification and gradient boosting. In *Neural Information Processing Systems*, volume 7, pages 845–852.

Li, W. and McCallum, A. (2006). Pachinko allocation: DAG-structured mixture models of topic correlations. In *International Conference on Machine Learning*, pages 577–584.

Li, X., Ouyang, J., and Zhou, X. (2015). Supervised topic models for multi-label classification. *Neurocomputing*, 149:811–819.

Liao, R., Zhu, J., and Qin, Z. (2014). Nonparametric Bayesian upstream supervised multimodal topic models. In *Web Search and Data Mining*, pages 493–502.

Lindsey, R. V., Headden, W. P., and Stipicevic, M. J. (2012). A phrase-discovering topic model using hierarchical Pitman-Yor processes. In *Empirical Methods on Natural Language Processing*, pages 214–222.

Liu, T.-Y. (2009). Learning to rank for Information Retrieval. *Foundations and Trends in Information Retrieval*, 3(3):225–331.

Liu, Y., Niculescu-Mizil, A., and Gryc, W. (2009). Topic-link LDA: joint models of topic and author community. In *International Conference on Machine Learning*, pages 665–672.

Lu, Y., Mei, Q., and Zhai, C. (2011). Investigating task performance of probabilistic topic models: An empirical study of PLSA and LDA. *Information Retrieval*, 14(2):178–203.

Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., and Potts, C. (2011). Learning word vectors for sentiment analysis. In *Association for Computational Linguistics*, pages 142–150.

MacDonald, C., Santos, R. L., and Ounis, I. (2013). The whens and hows of learning to rank for web search. *Information retrieval*, 16(5):584–628.

Metzler, D. and Croft, W. B. (2007). Linear feature-based models for information retrieval. *Information Retrieval*, 10(3):257–274.

Minka, T. and Robertson, S. (2008). Selection bias in the LETOR datasets. In *Special Interest Group on Information Retrieval Workshop on Learning to Rank for Information Retrieval*, pages 48–51.

Nallapati, R. (2004). Discriminative models for information retrieval. In *Special Interest Group on Information Retrieval*, pages 64–71.

Niu, S., Lan, Y., Guo, J., Cheng, X., and Geng, X. (2014). What makes data robust: a data analysis in learning to rank. In *Special Interest Group on Information Retrieval*, pages 1191–1194.

Noji, H., Mochihashi, D., and Miyao, Y. (2013). Improvements to the Bayesian topic n-gram models. In *Empirical Methods on Natural Language Processing*, pages 1180–1190.

Park, L. A. and Ramamohanarao, K. (2009). The sensitivity of Latent Dirichlet Allocation for Information Retrieval. In *Machine Learning and Knowledge Discovery in Databases*, pages 176–188.

Perotte, A. J., Wood, F., Elhadad, N., and Bartlett, N. (2011). Hierarchically supervised Latent Dirichlet Allocation. In *Neural Information Processing Systems*, pages 2609–2617.

Pinoli, P., Chicco, D., and Masseroli, M. (2014). Latent Dirichlet allocation based on Gibbs sampling for gene function prediction. In *Computational Intelligence in Bioinformatics and Computational Biology*, pages 1–8.

Pitman, J. and Yor, M. (1997). The two-parameter Poisson-Dirichlet distribution derived from a stable subordinator. *The Annals of Probability*, 25(2):855–900.

Porteous, I., Newman, D., Ihler, A., Asuncion, A., Smyth, P., and Welling, M. (2008). Fast collapsed Gibbs sampling for latent Dirichlet allocation. In *Knowledge Discovery and Data Mining*, pages 569–577.

Qin, T., Liu, T.-Y., Xu, J., and Li, H. (2010). LETOR: A benchmark collection for research on learning to rank for Information Retrieval. *Information Retrieval*, 13(4):346–374.

Quoc, C. and Le, V. (2007). Learning to rank with nonsmooth cost functions. *Neural Information Processing Systems*, 19:193–200.

Ramage, D., Hall, D., Nallapati, R., and Manning, C. D. (2009). Labeled LDA: A supervised topic model for credit attribution in multi-labeled corpora. In *Empirical Methods on Natural Language Processing*, pages 248–256.

Rubin, T. N., Chambers, A., Smyth, P., and Steyvers, M. (2012). Statistical topic models for multi-label document classification. *Machine Learning*, 88(1-2):157–208.

Salton, G., Wong, A., and Yang, C.-S. (1975). A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620.

Shafiei, M. M. and Milios, E. E. (2006). Latent Dirichlet co-clustering. In *International Conference on Data Mining*, pages 542–551.

Shao, Q.-M. and Ibrahim, J. G. (2000). *Monte Carlo methods in Bayesian computation*. Springer Series in Statistics, New York.

Sordoni, A., He, J., and Nie, J.-Y. (2013). Modeling latent topic interactions using quantum interference for information retrieval. In *Conference on Information and Knowledge Management*, pages 1197–1200.

Steyvers, M. and Griffiths, T. (2007). Probabilistic topic models. *Handbook of Latent Semantic Analysis*, 427(7):424–440.

Storkey, A. J. and Dai, A. (2014). The supervised Hierarchical Dirichlet Process. *Transactions on Pattern Analysis and Machine Intelligence*, 37(2):243–255.

Sun, Y., Deng, H., and Han, J. (2012). Probabilistic models for text mining. In *Mining Text Data*, pages 259–295.

Tan, M., Xia, T., Guo, L., and Wang, S. (2013). Direct optimization of ranking measures for learning to rank models. In *Knowledge Discovery and Data Mining*, pages 856–864. ACM.

Tang, J., Liu, N., Yan, J., Shen, Y., Guo, S., Gao, B., Yan, S., and Zhang, M. (2011). Learning to rank audience for behavioral targeting in display ads. In *Conference on Information and Knowledge Management*, pages 605–610.

Vapnik, V. (2000). *The nature of statistical learning theory*. springer.

Wallach, H. M. (2006). Topic modeling: Beyond bag-of-words. In *International Conference on Machine Learning*, pages 977–984.

Wallach, H. M. (2008). *Structured topic models for language*. PhD thesis.

Wallach, H. M., Mimno, D. M., and McCallum, A. (2009). Rethinking LDA: Why priors matter. In *Neural Information Processing Systems*, volume 22, pages 1973–1981.

Wang, C., Blei, D., and Li, F.-F. (2009). Simultaneous image classification and annotation. In *Conference on Computer Vision and Pattern Recognition*, pages 1903–1910.

Wang, L., Lin, J., Metzler, D., and Han, J. (2014). Learning to efficiently rank on big data. In *World Wide Web Conference*, pages 209–210.

Wang, Q., Xu, J., Li, H., and Craswell, N. (2011). Regularized latent semantic indexing. In *Special Interest Group on Information Retrieval*, pages 685–694.

Wang, Q., Xu, J., Li, H., and Craswell, N. (2013a). Regularized latent semantic indexing: A new approach to large-scale topic modeling. *Transactions on Information Systems*, 31(1):5.

Wang, S., Li, F., and Zhang, M. (2013b). Supervised topic model with consideration of user and item. In *Association for the Advancement of Artificial Intelligence*.

Wang, X. and McCallum, A. (2005). A note on topical n-grams. Technical report, DTIC Document.

Wang, X. and McCallum, A. (2006). Topics over time: A non-Markov continuous-time model of topical trends. In *Knowledge Discovery and Data Mining*, pages 424–433.

Wang, X., McCallum, A., and Wei, X. (2007). Topical N-grams: Phrase and topic discovery, with an application to Information Retrieval. In *International Conference on Data Mining*, pages 697–702.

Wei, X. and Croft, W. B. (2006). LDA-based document models for ad-hoc retrieval. In *Special Interest Group on Information Retrieval*, pages 178–185.

Wu, Q., Burges, C. J., Svore, K. M., and Gao, J. (2010). Adapting boosting for Information Retrieval measures. *Information Retrieval*, 13(3):254–270.

Wu, W. and Zhong, T. (2013). Searching the deep web using proactive phrase queries. In *World Wide Web Conference Companion*, pages 137–138.

Xie, B. and Passonneau, R. J. (2012). Supervised HDP using prior knowledge. In *Natural Language Processing and Information Systems*, pages 197–202. Springer.

Xu, J. and Li, H. (2007). AdaRank: a boosting algorithm for information retrieval. In *Special Interest Group on Information Retrieval*, pages 391–398.

Yan, X., Guo, J., Lan, Y., and Cheng, X. (2013). A biterm topic model for short texts. In *World Wide Web Conference*, pages 1445–1456.

Yao, L., Mimno, D., and McCallum, A. (2009). Efficient methods for topic model inference on streaming document collections. In *Knowledge Discovery and Data Mining*, pages 937–946.

Yi, X. and Allan, J. (2008). Evaluating topic models for Information Retrieval. In *Conference on Information and Knowledge Management*, pages 1431–1432.

Yi, X. and Allan, J. (2009). A comparative study of utilizing topic models for information retrieval. In *European Conference on Information Retrieval*, pages 29–41.

Yu, H. and Kim, S. (2012). SVM tutorial-classification, regression and ranking. In *Handbook of Natural Computing*, pages 479–506. Springer.

Yu, Z., Wu, F., Zhang, Y., Tang, S., Shao, J., and Zhuang, Y. (2014). Hashing with list-wise learning to rank. In *Special Interest Group on Information Retrieval*, pages 999–1002.

Yuan, N. J., Zhang, F., Lian, D., Zheng, K., Yu, S., and Xie, X. (2013). We know how you live: exploring the spectrum of urban lifestyles. In *Online Social Network*, pages 3–14.

Yue, Y., Finley, T., Radlinski, F., and Joachims, T. (2007). A support vector method for optimizing average precision. In *Special Interest Group on Information Retrieval*, pages 271–278.

Zellner, A. (1988). Optimal information processing and Bayes's theorem. *The American Statistician*, 42(4):278–280.

Zhai, C. and Lafferty, J. (2004). A study of smoothing methods for language models applied to information retrieval. *Transactions on Information Systems*, 22(2):179–214.

Zhang, C., Ek, C. H., Gratal, X., Pokorny, F. T., and Kjellström, H. (2013). Supervised Hierarchical Dirichlet Processes with variational inference. In *ICCV Workshop: Inference for Probabilistic Graphical Models*, pages 254–261.

Zhang, J. and Mani, I. (2003). kNN approach to unbalanced data distributions: a case study involving information extraction. In *Proceedings of Workshop on Learning from Imbalanced Datasets*.

Zhu, J., Ahmed, A., and Xing, E. P. (2009). MedLDA: Maximum margin supervised topic models for regression and classification. In *International Conference on Machine Learning*, pages 1257–1264.

Zhu, J., Ahmed, A., and Xing, E. P. (2012a). MedLDA: Maximum margin supervised topic models. *Journal of Machine Learning Research (JMLR)*, 13:2237–2278.

Zhu, J., Chen, N., Perkins, H., and Zhang, B. (2013a). Gibbs max-margin topic models with fast sampling algorithms. In *International Conference on Machine Learning*, pages 124–132.

Zhu, J., Chen, N., and Xing, E. P. (2011). Infinite latent SVM for classification and multi-task learning. In *Neural Information Processing Systems*, pages 1620–1628.

Zhu, J., Chen, N., and Xing, E. P. (2012b). Bayesian inference with posterior regularization and infinite latent support vector machines. *CoRR*, abs/1210.1766.

Zhu, J., Chen, N., and Xing, E. P. (2014). Bayesian inference with posterior regularization and applications to infinite latent SVMs. *Journal of Machine Learning Research (JMLR)*, 15:1799–1847.

Zhu, J., Zheng, X., and Zhang, B. (2013b). Improved Bayesian logistic supervised topic models with data augmentation. In *Association for Computational Linguistics*, pages 187–195.

Zhu, J., Zheng, X., Zhou, L., and Zhang, B. (2013c). Scalable inference in max-margin topic models. In *Knowledge Discovery and Data Mining*, pages 964–972.

Zong, W. and Huang, G.-B. (2014). Learning to rank with extreme learning machine. *Neural Processing Letters*, 39(2):155–166.