An Unsupervised Topic Segmentation Model Incorporating Word Order*

Shoaib Jameel and Wai Lam
Department of Systems Engineering and Engineering Management,
The Chinese University of Hong Kong,
Hong Kong.
{msjameel, wlam}@se.cuhk.edu.hk

ABSTRACT

We present a new unsupervised topic discovery model for a collection of text documents. In contrast to the majority of the state-of-the-art topic models, our model does not break the document's structure such as paragraphs and sentences. In addition, it preserves word order in the document. As a result, it can generate two levels of topics of different granularity, namely, segment-topics and word-topics. In addition, it can generate n-gram words in each topic. We also develop an approximate inference scheme using Gibbs sampling method. We conduct extensive experiments using publicly available data from different collections and show that our model improves the quality of several text mining tasks such as the ability to support fine grained topics with n-gram words in the correlation graph, the ability to segment a document into topically coherent sections, document classification, and document likelihood estimation.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval - Clustering

Keywords

Topic Modeling, Topic Segmentation, N-gram words, Gibbs Sampling, Document Classification

1. INTRODUCTION

Simplicity may not always lead to greatness! Topic models such as Latent Dirichlet Allocation (LDA) [5] have been widely used to find topics in a document collection. Typically, a topic is a probability distribution over words. But the LDA model has been criticized for its bag-of-words assumption [32] as the model does not consider the structural information inherent in the text which could help tap extra knowledge from the text. It is well known that the bag-of-words assumption is mainly a simplifying assumption to reduce the complexity of the model [24].

Some recent topic models have demonstrated better qualitative and quantitative performance when the bag-of-words assumption is relaxed [18], [20], [1], and [23]. Maintaining the word order during the processing of documents introduces some computational overhead, but it allows us to achieve what the bag-of-words models cannot do in general [15]. In order to address the shortcoming inherent in the LDA model, the authors in [39] introduced the Topical N-gram model (TNG) to find n-gram words in topics. By n-gram we mean a word can be a unigram, a bigram, a trigram word, etc. The TNG model has the ability to decide whether to form a unigram or a bigram during the topic discovery process. The TNG model mainly extends the LDA Collocation model [14] (LDACOL) and the Bigram Topic Model [35] (BTM). All these models advocate that the word order in a document is essential. But one shortcoming of these models is that they lack the ability to consider the document's structure such as paragraphs and sentences. Thus they cannot segment a document into coherent topics. This sometimes becomes essential in tasks such as tackling the word sense disambiguation problem as shown in [15], segmenting news articles and finding topics in each segment [30], topic detection and tracking [40], and a plethora of other tasks which motivate us to explore deeper into the topic segmentation model with n-gram topic word discovery.

We propose a new unsupervised topic discovery model, called NTSeg, for a collection of text documents. NTSeg maintains the segment structure of the document such as paragraphs and sentences. In addition, it preserves the word order in the document. NTSeg can help capture topical changes in the document from one segment to another. As a result, it can generate two levels of topics of different granularity, namely, segment-topics and word-topics. In addition, it can generate n-gram words in each word-topic. We also conduct extensive experiments on publicly available datasets to demonstrate the superiority of NTSeg in comparison to the state-of-the-art models in solving several text mining tasks such as the ability to support fine grained topics with n-gram words in the correlation graph, the ability to segment

*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 Code: CUHK413510) and the Direct Grant of the Faculty of Engineering, CUHK (Project Codes: 2050476 and 2050522). 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 Xiaojun Qian for his help with the experiments and some discussions related to the technical content in the paper.

SIGIR'13, July 28–August 1, 2013, Dublin, Ireland.

ACM 978-1-4503-2034-4/13/07 ...$15.00.

a document into topically coherent sections, document classification, and document modeling.

2. RELATED WORK

Topic models: Some previous works have considered the order of the words in a document during the topic discovery process. For example, in [35], the author described the Bigram Topic Model (BTM) which incorporates the hierarchical Dirichlet language model into the unigram based topic model in order to capture the dependencies between the words in sequence. One drawback of this model is that it only generates bigram words in a topic. This limitation was addressed in the LDA Collocation model (LDACOL) [14] which introduces a new set of status variables in the model called the bigram status variable. This variable indicates whether two consecutive words form a bigram or not. One limitation of the LDACOL model is that it cannot decide whether to form a unigram or a bigram for the same two consecutive words depending on their nearby context. Another issue with the model is that only the first term in a bigram has a topic assignment. One needs to make some assumptions in order to give the topic assignment to every term in a bigram [38], and [37]. The limitations inherent in the LDACOL model have been addressed in the Topical N-gram model, TNG [39]. The TNG model allows for consecutive words in a topic model to depend on each other in that they can be selected either to come from a unigram term distribution or from a bigram distribution. However, one limitation of the Topical N-gram model is that it cannot segment a document into topically coherent units, known as topic segmentation task. There are also other limitations and they have been addressed recently in [23] where the authors gave the same topic assignment to every term in a phrase and the words share the same probability mass in the phrase by introducing the hierarchical Pitman-Yor processes (HPYP) [34] in their model named PDLDA. The PDLDA model also cannot perform topic segmentation. Incorporating the HPYP model in our NTSeg model to capture phrasal terms would make our model NTSeg overly complex leading to inefficient processing of large text corpora.

Recently in [26] the authors proposed a topic model where n-gram words are viewed as random variables. The authors manually generated n-gram words from the dataset and considered those n-grams as part of the vocabulary. This model focuses on handling discussions or debates. In [19], the author proposed another topic n-gram model which mines sentiments from text corpora. Wang et al. [36] proposed an n-gram topic model for academic retrieval. They apply an online inference algorithm and find unigrams and bigrams in a topic. In [42], the authors presented an n-gram based news thread extraction model that uses the TNG model with a background distribution. A method which has adopted a different approach to n-gram topic modeling is [20]. It combines the paradigms of frequent pattern mining and topic modeling. All the topic models proposed above do not employ topic segmentation. In [41], the authors proposed an Auto Topic Number LDA (ATNLDA) model for topic segmentation and apply the model on stem cell research literature. This model can automatically calculate the optimal topic number. A difference between ATNLDA and ours is that ATNLDA does not consider the word order.

Topic Correlations: Shafiei et al. in [31] described a latent Dirichlet co-clustering method, known as LDCC, which captures correlation between word-topics and document-topics (or super-topics). The LDCC model is a hierarchical topic model where unigram words are assigned to the word-topics and paragraphs are assigned to the document-topics. The model can also find correlations between the word-topics and the document-topics. In [3], the authors proposed correlated topic model (CTM) that can capture the evolution of topics over time without considering the word order. In [22], the authors proposed Pachinko Allocation Model (PAM) where the concept of topic is extended to not only including distributions over words, but also distributions over topics. This model assumes the structure of an arbitrary DAG in which each leaf is associated with a word and each non-leaf node is a distribution over its children. The interior nodes are distributions over topics called super-topics. Recently, in [6], the authors presented a new model to find correlation among topics in a corpus using the Generalized Dirichlet distribution model instead of the Dirichlet distribution.

One similarity between our work and the topic correlation models is that our work also introduces two levels of topic assignments. For example, word-topics and document-topics as described in Shafiei et al. [31] share the same notion as the word-topics and segment-topics described in our proposed approach. We adopt the name segment-topics because known text segments such as paragraphs or sentences are assigned to the segment-topics. However, all the correlation topic models mentioned above assume exchangeability among the words in a document. The importance of capturing n-gram words is that it reduces the ambiguity in the mind of the reader as to what the word is referring to in the correlation graph. For example, presenting the word "networks" in a topic is ambiguous especially for a person who is not a domain expert. In contrast, showing the word "neural networks" in a topic significantly reduces ambiguities. In addition, all the topic correlation models mentioned above cannot conduct topic segmentation.

Topic segmentation: In [16], the author presented the TextTiling algorithm for segmenting a textual discourse into coherent segments. TextTiling is a domain-independent text segmentation technique that assigns a score to each segment boundary candidate based on a cosine similarity measure between chunks of words appearing to the left and right of the candidate. Then the segment boundaries are placed at the locations of valleys under this measure, and are then adjusted to coincide with known paragraph boundaries. TextTiling algorithm does not find topics. In [12], the authors presented a topic segmentation based approach which takes into account the lexical cohesion. The authors modeled lexical cohesion in a Bayesian context and obtained significant improvement against the existing models. They assumed that words are drawn from a multinomial language model. Our work is significantly different from the above two models in that ours is a domain-independent topic segmentation method based on a topic model. Apart from finding a low-dimensional representation of the original vector space, our model also segments a document and finds n-gram words in each topic.

In [25], the authors presented a method for topic segmentation based on topic modeling where the authors used the LDA model to segment texts into coherent topics that assume exchangeability among the words in a document. In [4], the authors described another topic segmentation method by unifying the segmenting hidden Markov model in [27] and

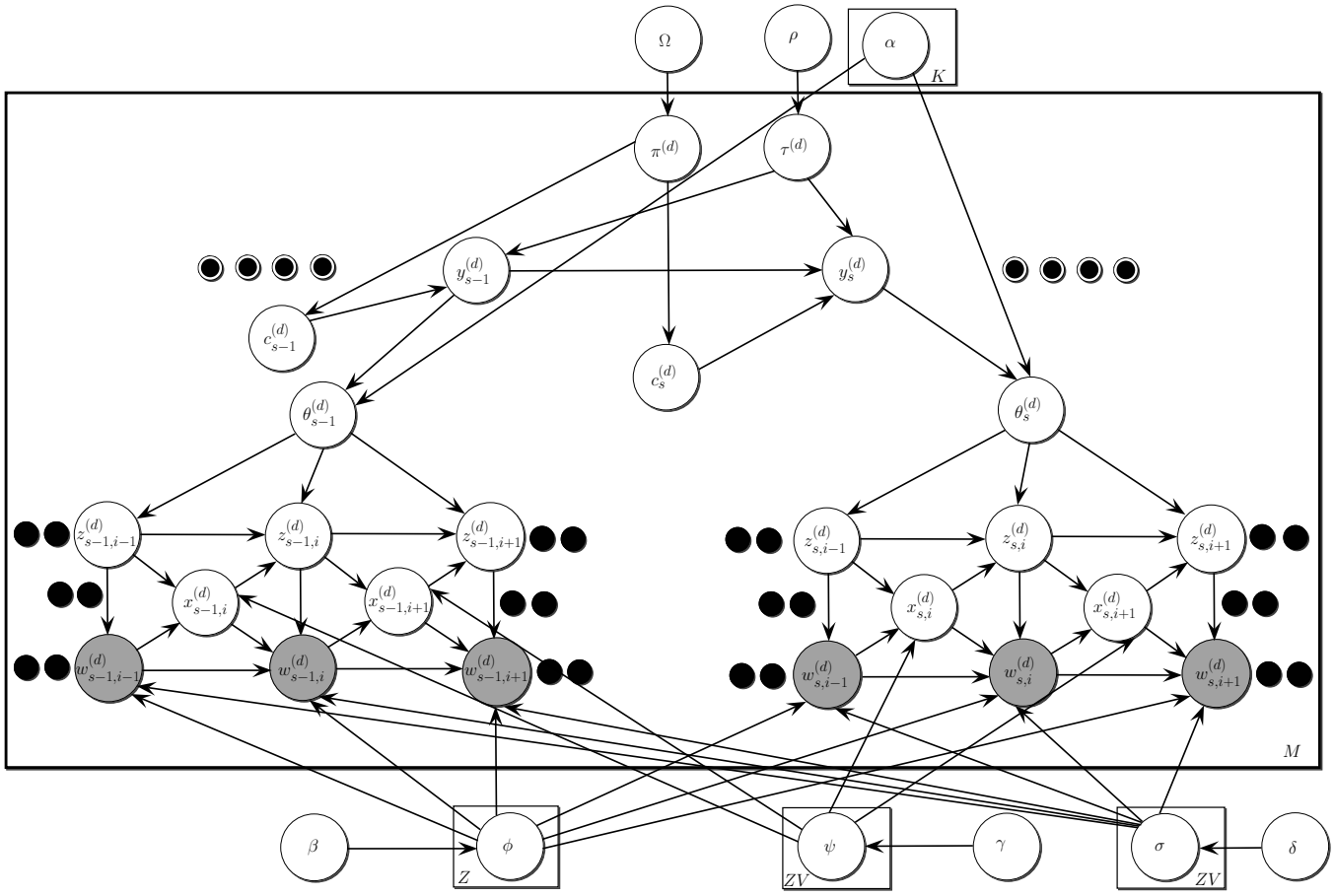

Figure 1: Our proposed model `NTSeg` in plate notation.

the aspect model in [17]. Recently, in [30], the authors presented `TopicTiling` based on `LDA`. Their algorithm is very similar to the `TextTiling` algorithm, and segments documents using the `LDA` topic model. Also, in [29], the authors presented methods in which topic models can help segmentation based methods by extending their own `TopicTiling` model. Similarly, in [32] the authors proposed a topic segmentation based topic model, known as `LDSEG`, where they assumed that the word order is not important. The authors introduced the notion of topic hierarchy where sentences are assigned to the document-topics and unigrams are assigned to the word-topics. The `LDSEG` model represents documents as a distribution over document-topics or super-topics in such a way that each segment is assigned a super-topic or document-topic, which is then used to choose the parameters of a document independent Dirichlet distribution from which the word-topics for the segment is drawn. In order for consecutive segments to have similar word-topic distributions, an additional binary variable per segment encodes whether the document-topic is forced to be the same as that of the previous segment. In [9], the authors proposed a topic model based hierarchical segmentation approach where they assumed that the word order within the segment is not important and apply variational Bayesian Expectation-Maximization procedure for computing the posterior inference. This model has been designed for segmenting the speech data. In [11], the authors proposed a collapsed Gibbs sampler for the topic segmentation problem for a faster posterior inference. They employ a hierarchical Pitman-Yor processes to handle hierarchical modeling, which our model does not incorporate. In [33], the authors presented a topic segmentation model which does not find topics in a segment. In contrast to the above models, our model does not assume exchangeability among the words in a document. In [8], the authors proposed a subsequence based topic segmentation approach which uses a suffix tree model for representing text and measures coherence between sentences based on subsequence. Their model maintains the order of the words in each segment, but the model does not find collocations in that text segment.

3. OUR PROPOSED MODEL (NTSeg)

We depict our proposed `NTSeg` model in Figure 1 using a graphical model in plate notation where shaded circles represent observed variables and unshaded ones are the latent variables. Each document, in general, is organized as atomic segments such as paragraphs or sentences. Our model preserves this structure. One characteristic of our model is that a document comprises of several topically coherent segments. Another characteristic of our model is due to the preservation of the ordering of the words it is able to capture word-collocations. Thus our proposed model is no longer invariant to the reshuffling of the words in each segment.

For a properly written discourse comprehension, documents are generally composed of coherent segments which

are semantically linked to one another so that a reader could relate the storyline as one moves forward in the discourse [21]. Our model comprises of two levels of topic of different granularity. One is the segment-topic to which atomic segments in a document are assigned and the ordering of segments as they appear in the document defines the topic change-points in the document. The other is the word-topic to which n-gram words in the segment are assigned. Segment-topics come from a predefined number of segment-topics K. Each segment-topic comprises of a mixture of several word-topics where the mixture coefficients uniquely specify the segment-topic. Word-topics come from a predefined number of word-topics Z. In general, the number of segment-topics will be less than the number of word-topics. The reason is that the number of segments in a document is less than compared to the number of words [31].

In the graphical model shown in Figure 1, M denotes the number of documents in the collection and V denotes the number of words in the vocabulary. In each atomic segment s, `NTSeg` finds n-gram words in a word-topic $\mathbf{z}$. It can also find correlations between both kinds of topics i.e. word-topics $\mathbf{z}$ and segment-topics $\mathbf{y}$. The segments of each document are assumed to follow a Markov structure on the topic distributions of each segment. We assume that there will be a high probability that the topic for the segment s in the document will be the same as that of the segment $s-1$. A segment binary switching variable $c_s^{(d)}$ for the segment-topic in the document d indicates whether there is a change of topic between the segments. The states of the switching variable correspond to the segmentation of the document into coherent topical units. Apart from the segment switching binary variable, `NTSeg` also incorporates another random variable known as the bigram status variable $\mathbf{x}$ which indicates the bigram status i.e. whether a word w at position i in the segment s in the document d, denoted as $w_{si}^{(d)}$, forms a bigram with the previous word $w_{s,i-1}^{(d)}$. The mechanism is that if $x_{si}^{(d)} = 1$, then $w_{s,i-1}^{(d)}$ and $w_{si}^{(d)}$ form a bigram else they do not.

It can be observed that the existing topical n-gram model (`TNG`) [39], `LDSEG`, [32], and `LDCC`, [31] are special cases derived from our model. For example, consider only a segment, for instance segment s, in Figure 1. Removing the segmentation scheme along with a set of arrows pointing from $z_{s,i-1}^{(d)} \rightarrow z_{si}^{(d)}$ and $x_{si}^{(d)} \rightarrow z_{si}^{(d)}$ reduces to the `TNG` model. One can observe that our model, `NTSeg`, has the capability of deciding whether to generate a unigram or a bigram in a topic and the topic assignment for the words in a bigram are the same. This aspect differentiates `NTSeg` from `TNG`. Similar to `TNG`, `NTSeg` assumes a first order Markov assumption i.e. it is mainly a bigram model, but the basic generation process produces unigram or bigram words. However, `NTSeg` has the ability to produce higher order n-grams (i.e. $n > 2$) by concatenating consecutive n-grams (unigram or bigram words) having the same topic and the bigram status variable between them is 1. In this way, the words in the n-gram share the same topic. This again contrasts `NTSeg` from `TNG` where `TNG` analyzes each n-gram post hoc as if the topic of the final word in the n-gram was the topic assignment of the entire n-gram. But it violates the principle of non-compositionality [23]. Removing the bag-of-words assumption in each segment of our proposed `NTSeg` model reduces to the `LDSEG` model. Relaxing both bag-of-words and removing the segmentation switch variable of `NTSeg` reduces to the `LDCC` model. `NTSeg` has the ability to decide whether to form a unigram or bigram based on context which the `LDSEG` model cannot achieve.

The following generative process of our model, `NTSeg`, helps better understand the graphical model shown in Figure 1:

1. Draw ϕ_z from $\mathbf{Dirichlet}(\beta)$ for each word-topic z, where ϕ_z is the word (unigrams only) distribution for the word-topic z; β is the parameter of the Dirichlet prior on the per-word-topic word (unigrams only) distribution
2. Draw ψ_{zw} from $\mathbf{Beta}(\gamma)$ for each word-topic z and each word w, where ψ_{zw} is the Bernoulli distribution for the bigram status variables for the word-topic z and the word w; γ is the parameter of the Beta prior
3. Draw σ_{zw} from $\mathbf{Dirichlet}(\delta)$ for each word-topic z, and each word w, where σ_{zw} is the bigram word distribution for bigrams; δ is the parameter of the Dirichlet prior on word-topic bigram word distribution
4. For each document d in the collection
   (a) Draw $\tau^{(d)}$ from $\mathbf{Dirichlet}(\rho)$, where $\tau^{(d)}$ is the mixing proportion of the segment-topics in the document d; ρ is the parameter of the Dirichlet prior on the segment-topics
   (b) Draw $\pi^{(d)}$ from $\mathbf{Beta}(\Omega)$, where $\pi^{(d)}$ defines the parameter of the Bernoulli distribution for the segment switch variable in the document d; Ω is the parameter of the Beta prior
   (c) For each segment s in the document d
      i. Draw $c_s^{(d)}$ from $\mathbf{Bernoulli}(\pi^{(d)})$
      ii. Draw the segment-topic $y_s^{(d)}$ for s from $\mathbf{Multinomial}(\tau^{(d)})$ if $c_s^{(d)} = 0$ else $y_s^{(d)} = y_{s-1}^{(d)}$, where $y_s^{(d)}$ is the segment-topic that is assigned to the segment s in the document d
      iii. Draw $\theta_s^{(d)}$ for s in the document d from $\mathbf{Dirichlet}(\alpha_{y_s^{(d)}z})$; α is a $K \times Z$ matrix where each row represents the mixing proportion of the word-topics in a segment-topic; $\theta_s^{(d)}$ is the mixing proportion of the word-topics in the segment s in the document d
      iv. For each of $N_s^{(d)}$ words in the segment s in the document d, where $N_s^{(d)}$ denotes the number of words in the segment s in the document d
         A. Draw $x_{si}^{(d)}$ from $\mathbf{Bernoulli}(\psi_{z_{s,i-1}^{(d)} w_{s,i-1}^{(d)}})$, where $x_{si}^{(d)}$ is the bigram status variable between words $w_{s,i-1}^{(d)}$ and $w_{si}^{(d)}$ in the segment s of the document d
         B. Draw $z_{si}^{(d)}$ from $\mathbf{Multinomial}(\theta_s^{(d)})$ if $x_{si}^{(d)} = 0$ else $z_{si}^{(d)} = z_{s,i-1}^{(d)}$, where $z_{si}^{(d)}$ is the word-topic assignment for the word $w_{si}^{(d)}$ in the segment s in the document d.
         C. Draw $w_{si}^{(d)}$ from $\mathbf{Multinomial}(\sigma_{z_{si}^{(d)} w_{s,i-1}^{(d)}})$ if $x_{si}^{(d)} = 1$ else draw $w_{si}^{(d)}$ from $\mathbf{Multinomial}(\phi_{z_{si}^{(d)}})$

$c_s^{(d)}$ indicates whether there is a change in the segment-topic between the segments $s-1$ and s in the document d. If $c_s^{(d)}=1$ then it means that $y_s^{(d)}=y_{s-1}^{(d)}$ i.e. segment-topic does not change between the segments in the document d. However, when $c_s^{(d)}=0$, then $y_s^{(d)}$ is drawn from a Multinomial distribution parameterized by $\tau^{(d)}$. The computation of the probability $P(y_s^{(d)}|c_s^{(d)},\tau^{(d)},y_{s-1}^{(d)})$ is done based on two conditions i.e. $\rho(y_s^{(d)},y_{s-1}^{(d)})$ when $c_s^{(d)}=1$ or sampling from **Multinomial** $(\tau^{(d)})$ when $c_s^{(d)}=0$.

The segment distribution $P(y_s^{(d)}|c_s^{(d)},\tau^{(d)},y_{s-1}^{(d)})$ is not properly defined for the first segment of every document. Therefore, $c_s^{(d)}=1$ is defined for the first segment which is drawn from **Multinomial**$(\tau^{(d)})$. Similarly we assume that $x_{s1}^{(d)}$ is observed and only unigram is allowed at the beginning of every segment.

4. POSTERIOR INFERENCE

The inference problem is related to computing the posterior probability of the hidden variables when the input parameters $\beta,\gamma,\delta,\rho,\Omega$ and the observed variable $\mathbf{w}$ are given. Also, an estimate of the α hyperparameter has to be made. It can be shown that computing the exact inference in our model is intractable. Hence, we need to resort to approximation techniques such as Gibbs sampling [7]. Adoption of Bayesian methods results in some hidden parameters being integrated out instead of being explicitly estimated. Assuming conjugate priors on the model parameters also eases the inference algorithm significantly. Algorithm 1 depicts the Gibbs sampling used in our approximate inference for `NTSeg`

We need to compute the two conditional distributions:

$$P(z_{si}^{(d)},x_{si}^{(d)}|z_{\neg si}^{(d)},x_{\neg si}^{(d)},\mathbf{w},\mathbf{c},\mathbf{y},\mathbf{x},\alpha,\beta,\gamma,\delta,\rho,\Omega) \quad (1)$$

$$P(y_s^{(d)},c_s^{(d)}|\mathbf{z},y_{\neg s}^{(d)},c_{\neg s}^{(d)},\mathbf{w},\mathbf{x},\alpha,\beta,\gamma,\delta,\rho,\Omega) \quad (2)$$

Note that $w_{\neg si}^{(d)}$ defines all the words in the segment except the current word $w_{si}^{(d)}$ in segment s in the document d. $z_{\neg si}^{(d)}$ is the word-topic assignment for all other words except the current word $w_{si}^{(d)}$. In Equations 3 and 4, n_{zw} is the number of times the word w is assigned to the word-topic z as a unigram. m_{wvz} is the number of times the word w appears as a second word of a bigram with a previous word v and both words in the bigram are assigned to the same word-topic z. p_{zwt} denotes the number of times the status variable $x=t$ (0 or 1) given the previous word w and the previous word's word-topic z. $h_{sz}^{(d)}$ is the number of times a word in segment s of document d is assigned to word-topic z. $\kappa_{c_s,0}^{(d)}$ and $\kappa_{c_s,1}^{(d)}$ is the number of times the switching variable $c_s^{(d)}$ is set to 0 and 1 in the document d, respectively. $\rho_{y_s^{(d)}}$ is the corresponding Dirichlet parameter for the segment-topic $y_s^{(d)}$. $b_k^{(d)}$ is the number of times a segment in the document d has been assigned to the segment-topic k. $y_{\neg s}^{(d)}$ is the segment-topic assignments for all the segments except the current segment s in the document d.

Beginning with the joint probability of a dataset, and using the chain rule, we obtain the conditional probabilities conveniently. We obtain the following equations:

$$P(z_{si}^{(d)},x_{si}^{(d)}|\mathbf{w},z_{\neg si}^{(d)},x_{\neg si}^{(d)},\mathbf{y},\mathbf{c},\alpha,\beta,\gamma,\delta,\rho,\Omega) \propto (\alpha_{y_s^{(d)}z_{si}^{(d)}}+h_{sz_{si}^{(d)}}^{(d)}-1)\times(\gamma_{x_{si}^{(d)}}+p_{z_{s,i-1}^{(d)}w_{s,i-1}^{(d)}x_{si}^{(d)}}-1) \times \begin{cases} \frac{\beta_{w_{si}^{(d)}}+n_{z_{si}^{(d)}w_{si}^{(d)}}-1}{\sum_{v=1}^{V}\left(\beta_v+n_{z_{si}^{(d)}v}\right)-1} & \text{if } x_{si}^{(d)}=0 \\ \frac{\delta_{w_{si}^{(d)}}+m_{w_{si}^{(d)}w_{s,i-1}^{(d)}z_{si}^{(d)}}-1}{\sum_{v=1}^{V}\left(\delta_v+m_{w_{s,i-1}^{(d)}vz_{si}^{(d)}}\right)-1} & \text{if } x_{si}^{(d)}=1 \;\&\; z_{si}^{(d)}=z_{s,i-1}^{(d)} \end{cases} \quad (3)$$

$$P(y_s^{(d)},c_s^{(d)}|\mathbf{z},y_{\neg s}^{(d)},c_{\neg s}^{(d)},\mathbf{w},\mathbf{x},\alpha,\beta,\gamma,\delta,\rho,\Omega) \propto \begin{cases} (\rho_{y_s^{(d)}}+b_{y_s^{(d)}}^{(d)}-1)\times(\alpha_{y_s^{(d)}z_{si}^{(d)}}+h_{sz_{si}^{(d)}}^{(d)}-1)\times \\ \left(\frac{\kappa_{c_s,0}^{(d)}+\Omega_0}{\sum_{x=0}^{1}\kappa_{c_s,x}^{(d)}+\Omega_0+\Omega_1}\right) & \text{if } c_s^{(d)}=0 \\ (\alpha_{y_s^{(d)}z_{si}^{(d)}}+h_{sz_{si}^{(d)}}^{(d)}-1)\times\left(\frac{\kappa_{c_s,1}^{(d)}+\Omega_1}{\sum_{x=0}^{1}\kappa_{c_s,x}^{(d)}+\Omega_0+\Omega_1}\right) \\ \qquad \text{if } c_s^{(d)}=1 \;\&\; s>1 \;\&\; y_s^{(d)}=y_{(s-1)}^{(d)} \end{cases} \quad (4)$$

Note that in our model the hyperparameter α captures the relationships between the segment-topics and word-topics. This hyperparameter must be estimated from the data. Although there are many ways to estimate this hyperparameter [31], we adopt the moment matching method which is computationally less expensive [31], and [22]. Therefore at each iteration of the Gibbs sampling (Line 37 of Algorithm 1), we update:

$$\overline{\lambda}_{kz}=\frac{1}{q_k}\sum_{s\in S_k}\frac{h_{sz}^{(d)}}{N_s^{(d)}} \quad (5) \qquad \overline{\nu}_{kz}=\frac{1}{q_k}\sum_{s\in S_k}\left(\frac{h_{sz}^{(d)}}{N_s^{(d)}}-\overline{\lambda}_{kz}\right)^2 \quad (6)$$

$$\alpha_{kz}\propto\overline{\lambda}_{kz} \quad (7) \qquad \lambda_{kz}=\frac{\overline{\lambda}_{kz}(1-\overline{\lambda}_{kz})}{\overline{\nu}_{kz}}-1 \quad (8)$$

$$\sum_{z=1}^{Z}\alpha_{kz}=\exp\left(\frac{\sum_{z=1}^{Z}\log(\lambda_{kz})}{Z-1}\right) \quad (9)$$

where S_k is the set of segments assigned to the segment-topic k. q_k is the number of segments assigned to the segment-topic k. $\overline{\lambda}_{kz}$ and $\overline{\nu}_{kz}$ are the sample mean and sample variance, respectively, of the number of times the word-topic z is assigned to the segment-topic k.

The posterior estimates for θ, ϕ, ψ, π, τ, σ are:

$$\hat{\theta}_{s,yz}^{(d)}=\frac{\alpha_{yz}+h_{sz}^{(d)}}{\sum_{z=1}^{Z}(\alpha_{yz}+h_{sz}^{(d)})} \quad (10) \qquad \hat{\phi}_{z,w}=\frac{\beta_w+n_{zw}}{\sum_{v=1}^{V}(\beta_v+n_{zv})} \quad (11)$$

$$\hat{\psi}_{zw,x}=\frac{\gamma_x+p_{zwx}}{\sum_{t=0}^{1}(\gamma_t+p_{zwt})} \quad (12) \qquad \hat{\sigma}_{zw,v}=\frac{\delta_w+m_{wvz}}{\sum_{v=1}^{V}(\delta_v+m_{wvz})} \quad (13)$$

$$\hat{\pi}_r^{(d)}=\frac{\Omega_r+\kappa_{c_s,r}}{\sum_{r=0}^{1}(\Omega_r+\kappa_{c_s,r})} \quad (14) \qquad \hat{\tau}_y^{(d)}=\frac{\rho_y+b_y^{(d)}}{\sum_{k=1}^{K}(\rho_k+b_k^{(d)})} \quad (15)$$

The target distribution is the posterior distribution of the word-topics, the segment-topics, the topic switching variables of the segments, and the bigram status variables. When we use the Gibbs sampling technique, at each iteration, we

sample from the conditional distribution of the word-topics in a document conditioned on the word-topic assignments for all other words except the current word (Line 23 in Algorithm 1). In addition, we also sample the bigram status variable (Line 24). We sample from the conditional distribution of a segment-topic for a segment (Line 14) and also the corresponding switching variable given the segment-topic assignments (Line 14).

At each iteration of the Gibbs sampling procedure, we only sample a subset of the variables which are directly related to the conditional probability. We perform this step repeatedly until we arrive at some approximation. A variable is sampled from the conditional distribution given that the assignments for all other variables are known which is a standard procedure in a Gibbs sampler. As the list of words is being scanned along with the bigram status variables, the sampler keeps track of any new segment being encountered. For each new segment, the sampler decides about the topic assignment of the segment i.e., whether it should assign the current segment to the same topic as the previous segment or a new segment-topic. If the segment has to be assigned to a new segment-topic, the sampler estimates the probability of assigning the segment to the segment-topic. These probabilities are computed from the conditional distribution for a segment given all other topic assignments to every other segment and all words in the segment as depicted in Algorithm 1.

Input : $\gamma, \delta, Z, K, \rho, \Omega, \beta, Corpus, MaxIteration$
Output: N-gram words derived from the word-topic assignments; assignments of segment-topics to segments in documents; an estimate of α

1 Initialize count variables in Equations 3 and 4 to 0;
2 Initialize $b_k^{(d)}$, $\kappa_{cs,0}^{(d)}$ and $\kappa_{cs,1}^{(d)}$ for all values of $k \in \{1, \cdots, K\}$ in all documents;
3 Initialize $h_{sz}^{(d)}$ for all values of $z \in \{1, \cdots, Z\}$ in all documents and their segments;
4 Initialize n_{zw} and p_{zwt} for all values of $z \in \{1, \cdots, Z\}$ and for all words in the collection;
5 Initialize m_{wvz} for all values of $z \in \{1, \cdots, Z\}$ and for all bigrams in the collection;
6 Randomly initialize word-topic assignments, segment-topic assignments, segment-topic switch variables, and bigram status variables;
7 **if** *performing parameter value estimation* **then**
8 Initialize α using Equations 5, 6, 7, 8, 9;
9 **end**
10 **for** *iter* $\leftarrow$ *1* **to** $MaxIteration$ **do**
11 **foreach** *document* $d \in [1, M]$ *in the collection* **do**
12 **foreach** *segment s in the document d* **do**
13 Exclude segment s and its assigned topic k from the count variables;
14 $(new_k, new_c) \leftarrow$ sample new segment-topic and segment switching variable for segment s using Equation 4;
15 **if** *(*$new_c == 0$*)* **then**
16 Assign new_k as the new segment-topic for segment s;
17 Update variables $b_k^{(d)}$, and $\kappa_{cs,0}^{(d)}$ using the new segment-topic new_k for segment s;
18 **end**
19 **if** *(*$new_c = 1$*)* **then**
20 Update variable $\kappa_{cs,1}^{(d)}$;
21 **end**
22 **foreach** *word i in segment s in the document d, according to order* **do**
23 Exclude word i and its assigned word-topic z from the count variables;
24 $(new_z, new_x) \leftarrow$ sample new word-topic for word i and bigram status variable using Equation 3;
25 **if** *(*$new_x == 0$*)* **then**
26 Assign new_z as the new word-topic;
27 Update n_{zw}, $h_{sz}^{(d)}$, p_{zw0} using word-topic new_z for word i;
28 **end**
29 **if** *(*$new_x == 1$*)* **then**
30 Update $h_{sz}^{(d)}$, p_{zw1}, m_{wvz} for word i;
31 **end**
32 **end**
33 Update the posterior estimate for $\theta_s^{(d)}$ for each segment using Equation 10;
34 **end**
35 Update the posterior estimates for $\pi^{(d)}$ and $\tau^{(d)}$ for each document using Equations 14, and 15;
36 **end**
37 **if** *performing parameter value estimation* **then**
38 Update α using Equations 5, 6, 7, 8, and 9;
39 **end**
40 **end**
41 Update the posterior estimates for ϕ_{zw}, ψ_{zwt} and σ_{zw} using Equations 11,12, and 13,;

Algorithm 1: Inference algorithm for NTSeg.

5. EXPERIMENTS AND RESULTS

Evaluation of topic models is a challenging task. Simply showing the highly probable n-gram words obtained from each topic may not be able to portray the underlying strengths or weaknesses of a topic model. Therefore, we evaluate our model on several text mining tasks including the ability to support fine grained topics with n-gram words in the correlation graph, the ability to segment a document into topically coherent sections, document classification, and document likelihood estimation.

In each experiment, we chose several existing closely related comparative methods for comparison purpose. We will describe those comparative methods in the subsections that follow. For our proposed framework, NTSeg, the segment granularity is basically a paragraph because topical changes typically occur at paragraph boundary and this strategy is also used in [31]. Note that NTSeg can also work at the granularity of a sentence which has also been used in one of our experiments (refer Section 5.2). In our experiments, the number of iterations for the Gibbs sampler is 1000 which is the value of the $MaxIteration$ used in Algorithm 1. We have chosen the following hyperparameter values $\beta = 0.01$, $\gamma = 0.1$, $\delta = 0.1$, $\Omega = 0.1$, and $\rho = 0.1$. Other topic models such as TNG, LDSEG etc, also assume fixed hyperparameter values. We did not perform any stemming, but removed stopwords[1] from the collection.

5.1 Correlation Graph

NTSeg produces two levels of topics, namely, segment-topics and word-topics. A word-topic is comprised of n-grams. We show the correlation graph for the purpose of depicting how our model finds correlations among various

[1]http://jmlr.csail.mit.edu/papers/volume5/lewis04a/a11-smart-stop-list/english.stop

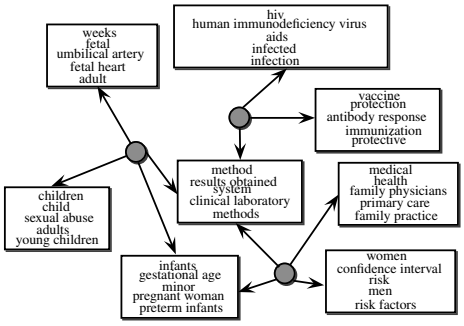

Figure 2: Correlation identified by `NTSeg` between the word-topics and the segment-topics on OHSUMED collection considering $Z = 200$ and $K = 100$. Each circle shows a segment-topic and each box corresponds to a word-topic. We can notice that a segment-topic can capture correlations between several word-topics.

segment-topics and word-topics. Details regarding constructing and interpreting such correlation graphs can be found in [22], and [31].

We have used the OHSUMED[2] collection to show the correlation graph. The collection is composed of 348,566 documents with 154,711 words in the vocabulary without stopwords.

We have experimented by varying both number of the word-topics Z and the number of the segment-topics K. Z was varied from 50 to 200 in steps of 50 whereas K was varied from 50 to 150 in steps of 50. However, we did not observe significant difference in the quality of the results. The resulting correlation graph is shown in Figure 2 which is obtained by setting $Z = 200$ and $K = 100$. Due to space constraint, we only show the graph obtained from our `NTSeg` model. Note that other models such as `PAM`, `LDCC`, `LDSEG`, `GD-LDA` [6] and `CTM`, only form unigrams in a topic leading to ambiguous interpretation. For example, presenting the unigram "confidence" will not be that insightful in a correlation graph. In contrast, presenting the term "confidence interval" is more meaningful as shown in Figure 2.

5.2 Topic Segmentation Experiment

The purpose of this experiment is to show how well `NTSeg` generates segmentation of documents corresponding to coherent topical units. The segmentation information is obtained via the segmentation switch variable $c_s^{(d)}$ which gives the segment topic change-points in the document. In our problem setting we know the segment boundaries in advance such as paragraphs or sentences, but we do not know the word and segment topics. Our purpose is thus to learn the segment and word topics from the document collection. The prediction output of the segment status variable will define the segmentation of a document. To evaluate the performance, we make use of the annotated segmentation information. We use two standard metrics, namely, Pk and WinDiff which are widely used in the topic segmentation literature [32]. As described in [32], Pk is defined as the probability that two segments drawn randomly from a document are incorrectly identified as belonging to the same topic [2]. WinDiff [28] moves a sliding window across the text and counts the number of times the hypothesized and reference segment boundaries are different from within the

[2]http://ir.ohsu.edu/ohsumed/ohsumed.html

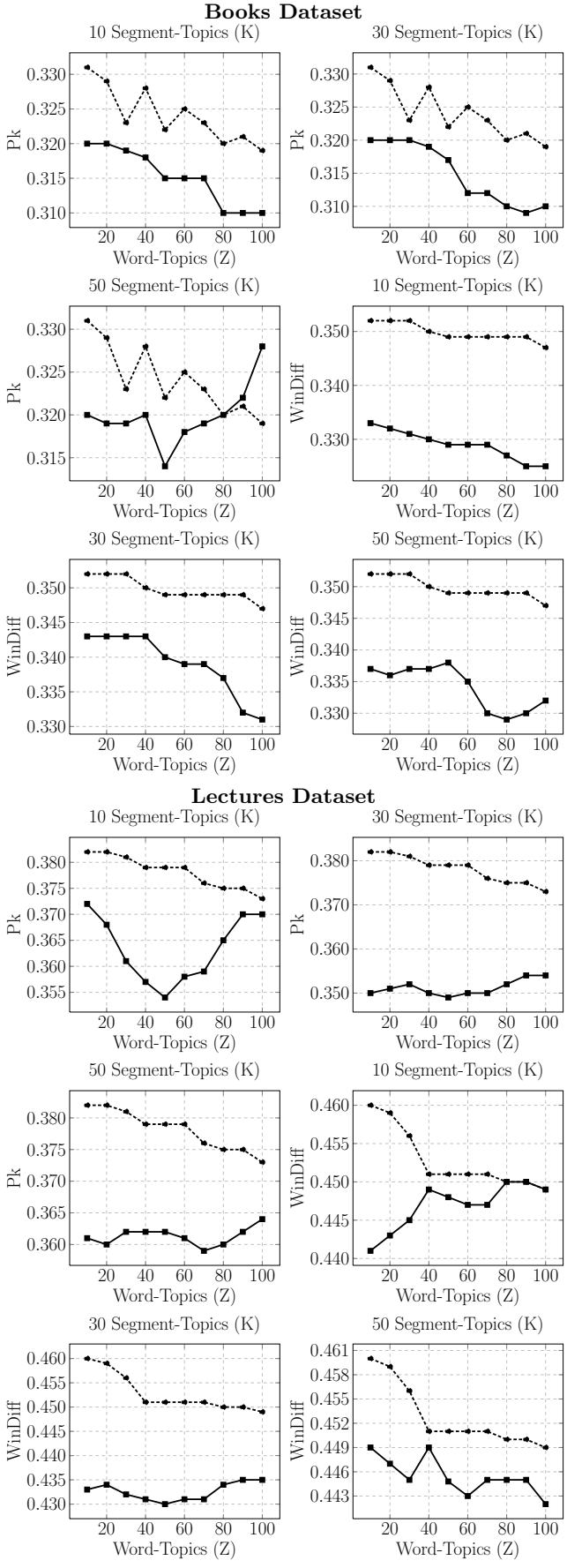

Figure 3: Comparison of `NTSeg` (depicted by ▬■▬ marker) against `TopicTiling` (depicted by - -●- - marker) on topic segmentation task.

	Precision	Recall	F-Measure
LDSEG	0.580	0.420	0.487
PAM	0.550	0.450	0.495
LDACOL	0.400	0.300	0.343
TNG	0.490	0.420	0.452
PDLDA	0.580	0.500	0.537
NTSeg	**0.640**	**0.520**	**0.574**

Table 1: Document classification results for the Computer Dataset of the 20 Newsgroups corpus.

	Precision	Recall	F-Measure
LDSEG	0.440	0.400	0.419
PAM	0.500	0.330	0.398
LDACOL	0.420	0.370	0.393
TNG	0.560	0.470	0.511
PDLDA	0.580	0.510	0.543
NTSeg	**0.620**	**0.560**	**0.588**

Table 2: Document classification results for the Science Dataset of the 20 Newsgroups corpus.

window. The lower the values obtained for these two metrics, the better is the segmentation result.

We use two publicly available datasets that contain segment boundaries corresponding to the topic changes. The first dataset, called Lectures in our experiment, consists of spoken lecture transcripts from an undergraduate physics class and a graduate artificial intelligence class. The transcripts consist of a 90 minute lecture recording and have 500 to 700 sentences with about 9000 words. Note that here the segment granularity is a sentence. More details about this dataset can be obtained from [32]. Our second dataset, called Books in our experiment, is the books[3] dataset in which each document is a chapter extracted from a medical textbook.

We chose a recently proposed topic segmentation method `TopicTiling` [30] which has outperformed many state-of-the-art text segmentation models proposed in the literature and chose the best performing variant of `TopicTiling` from [30]. Note that `TopicTiling` only has the notion of word-topics. For each of the segment and word-topics, we run the Gibbs sampler five times and take the average of the Pk and WinDiff values at the end of the fifth run.

We illustrate the segmentation results in Figure 3. From the results, we note that our model performs extremely well in both datasets compared to the state-of-the-art topic segmentation model. Using a two-tailed significance test, our results are statistically significant with $p < 0.05$ against `TopicTiling`. In the Books dataset, `NTSeg` performs reasonably better, but the improvement obtained is not very high considering both Pk and WinDiff metrics. However, good improvement is obtained in the Lectures dataset using both metrics.

5.3 Document Classification Experiment

We conduct document classification experiment using topic models. In the training phase, a topic model is learned for each class using the set of training documents in that class. In testing, to conduct document classification for a testing document, we compute the likelihood of the testing document against each trained topic model for each class. The testing document is classified to the model that produces the highest likelihood. Note that this procedure is also used in [22].

	Precision	Recall	F-Measure
LDSEG	0.390	0.320	0.352
PAM	0.540	0.490	0.514
LDACOL	0.550	0.410	0.470
TNG	0.550	0.450	0.495
PDLDA	0.590	0.410	0.484
NTSeg	**0.620**	**0.570**	**0.594**

Table 3: Document classification results for the Politics Dataset of the 20 Newsgroups corpus.

	Precision	Recall	F-Measure
LDSEG	0.330	0.320	0.325
PAM	0.368	0.360	0.363
LDACOL	0.200	0.180	0.189
TNG	0.340	0.290	0.313
PDLDA	0.380	0.210	0.271
NTSeg	**0.420**	**0.380**	**0.399**

Table 4: Document classification results for the Sports Dataset of the 20 Newsgroups corpus.

We measure the classification performance using precision, recall and F-measure. The meaning of precision for a class is the number of true positives divided by the total number of documents predicted to that class. Recall is defined as the number of true positives divided by the total number of elements that actually belong to that class in the gold standard. F-measure is the harmonic mean of precision and recall.

We use the 20 Newsgroups corpus[4] and generated four datasets. The first dataset comprises of documents related to computer technology (the "comp" directory in the dataset). It is composed of several classes such as "graphics", "windows", "hardware", etc. Each of these classes consists of 1000 documents. We split the documents in each of these classes into 75% training and 25% test documents. For each class, we trained and tested the model by varying the number of word-topics from 10 to 100 in steps of 10 and the number of segment-topics from 10 to 50 in steps 20. We compute precision and recall using the test set for each class for each word-topic and segment-topic values and then we compute the average result for one class across all word-topics and segment-topics. Similarly, we follow the same precision and recall computation for all classes. Finally we compute the average over all precision and recall values for all the classes. We then compute F-measure from the obtained precision and recall values. The experimental setup is similar for the other three datasets, namely, "sci" (called Science Dataset), "politics" (called Politics Dataset), and "sports" (called Sports Dataset).

The comparative methods include `LDSEG`, `PAM`, `LDACOL`, `TNG`, and `PDLDA`. All these models are described in Section 2. Note that some of the comparative methods such as `TNG`, `PDLDA`, and `LDACOL` have no notion of segment-topics.

The classification performance results are presented in Tables 1, 2, 3 and 4. We can observe that in all the datasets our model, `NTSeg`, has outperformed all the comparative methods. Compared to all the comparative methods, our results are also statistically significant using the sign test with $p < 0.05$. Gain obtained in the Computer and Science datasets is more when compared to the gain in Sports and

[3]http://groups.csail.mit.edu/rbg/code/bayesseg/

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

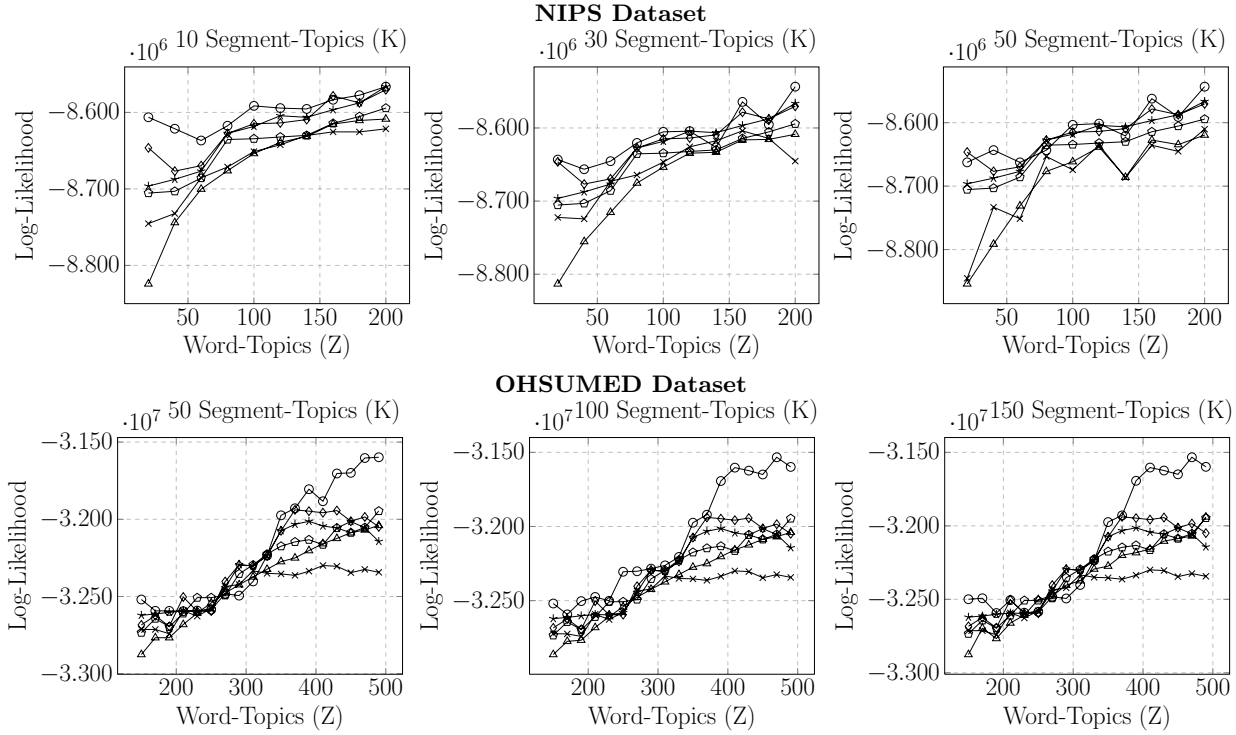

Figure 4: Performance of `NTSeg` (depicted by —○—) in terms of generalizing on the new data. We can see that our model `NTSeg` generalizes better than other comparative methods which are: `LDSEG` (depicted by —△—), `PAM` (depicted by —×—), `LDACOL` (depicted by —⬠—), `TNG` (depicted by —✶—), and `PDLDA` (depicted by —◇—).

Politics datasets. `PDLDA` also proved to be a better model in comparison to the other comparative methods.

5.4 Document Likelihood Experiment

Another evaluation scheme to compare the relative performance of topic models is to study how the models generalize on an unseen data. The entire corpus in this method is first split into training and testing set. The training set generally contains more number of documents as compared to the testing set. A model is first learned on the training data, and the testing set is used to measure the generalization performance of the topic models. In the topic modeling literature, metrics such as perplexity computation or log-likelihood have often been used. For example, `PAM` uses empirical log-likelihood [10] as an evaluation metric and so does a recently proposed method `GD-LDA` [6]. Log-likelihood has also been widely used as one of the evaluation metrics, for example in [3]. We chose log-likelihood metric for comparing the topic models. The comparative methods here are `LDSEG`, `PAM`, `LDACOL`, `TNG`, and `PDLDA`.

We use the NIPS dataset[5]. The NIPS collection is widely used in the topic modeling literature. Note that the original raw NIPS dataset consists of 17 years of conference papers. But we supplemented this dataset by including some new raw NIPS documents[6] and it has 19 years of papers in total. Our NIPS collection consists of 2741 documents comprising of 453,606,9 non-unique words and 94961 words in the vocabulary. In addition to the NIPS collection we also use the OHSUMED collection.

In order to calculate the likelihood of held-out data, we must integrate out the sampled multinomials and sum over all possible topic assignments which has no closed-form solution. Griffiths et al. [13] have used Gibbs sampling for computing such approximations. First, we randomly split each of the datasets into 80% training and 20% testing. We trained each of the topic models on the training set. We then tested the models on the testing set by running the inference algorithms five times for each word-topic and segment-topic pair. We then took average value for all five runs. We varied the number of segment-topics from 10 to 50 in steps of 20 and the number of word-topics from 20 to 200 in steps of 20 in the NIPS collection. As the OHSUMED collection is larger compared with the NIPS collection, so we varied the number of segment-topics from 50 to 150 in steps of 50 and word-topics from 150 to 490 in steps of 20.

From the results in Figure 4 we can see that in the NIPS collection, `NTSeg` performs better than the comparative methods especially when the number of segment-topics is 10. However, its performance deteriorates a bit when the number of segment-topics is increased, but still remains competitive with the comparative methods. Moreover, we notice that as the number of word-topics increases, the performance of `NTSeg` deteriorates to some extent in the NIPS collection. However, in the OHSUMED collection, `NTSeg` again performs better against the comparative methods when the number of word-topics is increased. We can observe that `NTSeg` outperforms the comparative methods considerably when the number of segment-topics is 100. The results suggest that `NTSeg` can perform very well on large document collections as large collections provide richer information about word co-occurrences.

[5]http://www.cs.nyu.edu/~roweis/data.html
[6]http://ai.stanford.edu/~gal/Data/NIPS/

6. CONCLUSIONS AND FUTURE WORK

We have developed a generative topic discovery model, known as `NTSeg`, which maintains the document's structure such as paragraphs and sentences and also keeps the order of the words in the document intact. `NTSeg` incorporates the notion of word-topics and segment-topics. We have conducted extensive experiments and shown results using both qualitative analysis where we show the n-gram words in the correlation graph and quantitative performance. Experimental results demonstrate that by relaxing the bag-of-words assumption in each segment improves the performance of the model.

Giving an arbitrary number of word-topics and segment-topics to the model is one issue that we would look into for future work. We would attempt to work towards a model which could automatically find out the desirable number of word-topics and segment-topics in the collection.

7. REFERENCES

[1] B. Adams, D. Phung, and S. Venkatesh. Discovery of latent subcommunities in a blog's readership. *ACM Trans. on the Web*, 4(3):12:1–12:30, 2010.

[2] D. Beeferman, A. Berger, and J. Lafferty. Statistical models for text segmentation. *Machine Learning*, 34:177–210, 1999.

[3] D. Blei and J. Lafferty. Correlated topic models. *Proc. of NIPS*, pages 147–155, 2006.

[4] D. M. Blei and P. J. Moreno. Topic segmentation with an aspect Hidden Markov model. In *Proc. of SIGIR*, pages 343–348, 2001.

[5] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent Dirichlet Allocation. *JMLR*, 3:993–1022, 2003.

[6] K. L. Caballero, J. Barajas, and R. Akella. The Generalized Dirichlet Distribution in enhanced topic detection. In *Proc. of CIKM*, pages 773–782, 2012.

[7] G. Casella and E. I. George. Explaining the Gibbs sampler. *The American Statistician*, 46(3):pp. 167–174, 1992.

[8] X. Chen and S. Chen. Subsequence-based text segmentation and labeling. In *Proc. of ECTS*, pages 582–587, 2009.

[9] J. Chien and C. Chueh. Topic-based hierarchical segmentation. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(1):55–66, 2012.

[10] P. Diggle and R. Gratton. Monte Carlo methods of inference for implicit statistical models. *Journal of the Royal Statistical Society*, pages 193–227, 1984.

[11] L. Du, W. Buntine, and H. Jin. A segmented topic model based on the two-parameter Poisson-Dirichlet process. *Machine Learning*, 81(1):5–19, 2010.

[12] J. Eisenstein and R. Barzilay. Bayesian unsupervised topic segmentation. In *Proc. of EMNLP*, pages 334–343, 2008.

[13] T. L. Griffiths and M. Steyvers. Finding scientific topics. *National Academy of Sciences of the United States of America*, 101(Suppl 1):5228–5235, 2004.

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

[15] A. Gruber, Y. Weiss, and M. Rosen-Zvi. Hidden topic Markov models. In *Proc. of AI and Statistics*, pages 163–170, 2007.

[16] M. A. Hearst. TextTiling: Segmenting text into multi-paragraph subtopic passages. *Comput. Linguist.*, 23(1):33–64, 1997.

[17] T. Hofmann. Unsupervised learning by Probabilistic Latent Semantic Analysis. *Machine Learning*, 42:177–196, 2001.

[18] S. Jameel and W. Lam. An N-gram topic model for time-stamped documents. In *Proc. of ECIR*, pages 292–304. 2013.

[19] N. Kawamae. Identifying sentiments over n-gram. In *Proc. of WWW*, pages 541–542, 2012.

[20] H. D. Kim, D. H. Park, Y. Lu, and C. Zhai. Enriching text representation with frequent pattern mining for probabilistic topic modeling. *Journal of American Society for Information Science and Technology*, 49(1):1–10, 2012.

[21] W. Kintsch. The role of knowledge in discourse comprehension: A construction-integration model. *Psychological Review*, 95(2):163, 1988.

[22] W. Li and A. McCallum. Pachinko allocation: DAG-structured mixture models of topic correlations. In *Proc. of ICML*, pages 577–584, 2006.

[23] R. V. Lindsey, W. P. Headden, and M. J. Stipicevic. A phrase-discovering topic model using hierarchical Pitman-Yor processes. In *Proc. of EMNLP*, pages 214–222, 2012.

[24] D. Metzler and W. B. Croft. A Markov Random Field model for term dependencies. In *Proc. of SIGIR*, pages 472–479, 2005.

[25] H. Misra, F. Yvon, J. M. Jose, and O. Cappe. Text segmentation via topic modeling: An analytical study. In *Proc. of CIKM*, pages 1553–1556, 2009.

[26] A. Mukherjee and B. Liu. Mining contentions from discussions and debates. In *Proc. of KDD*, pages 841–849, 2012.

[27] P. V. Mulbregt, I. Carp, L. Gillick, S. Lowe, and J. Yamron. Text segmentation and topic tracking on broadcast news via a hidden markov model approach. In *Proc. of ICSLP*, pages 2519–2522, 1998.

[28] L. Pevzner and M. A. Hearst. A critique and improvement of an evaluation metric for text segmentation. *Comput. Linguist.*, 28(1):19–36, 2002.

[29] M. Riedl and C. Biemann. How text segmentation algorithms gain from topic models. In *Proc. of NAACL HLT*, pages 553–557, 2012.

[30] M. Riedl and C. Biemann. TopicTiling: A text segmentation algorithm based on LDA. In *Proc. of ACL 2012 Student Research Workshop*, pages 37–42, 2012.

[31] M. Shafiei and E. Milios. Latent Dirichlet Co-Clustering. In *Proc. of ICDM*, pages 542 –551, 2006.

[32] M. Shafiei and E. Milios. A statistical model for topic segmentation and clustering. In *Proc. of Advances in Artificial Intelligence*, pages 283–295, 2008.

[33] A. Tagarelli and G. Karypis. A segment-based approach to clustering multi-topic documents. In *Proc. of SDM*, pages 1–33, 2008.

[34] Y. Teh, M. Jordan, M. Beal, and D. Blei. Hierarchical Dirichlet processes. *JASA*, 101(476):1566–1581, 2006.

[35] H. M. Wallach. Topic modeling: Beyond bag-of-words. In *Proc. of ICML*, pages 977–984, 2006.

[36] H. Wang and B. Lang. Online Ngram-enhanced topic model for academic retrieval. In *Proc. of ICDIM*, pages 137–142, 2011.

[37] L. Wang, B. Wei, and J. Yuan. Topic discovery based on LDACOL model and topic significance re-ranking. *Journal of Computers*, 6(8):1639–1647, 2011.

[38] X. Wang and A. McCallum. A note on Topical N-grams. Technical report, DTIC Document, 2005.

[39] X. Wang, A. McCallum, and X. Wei. Topical N-Grams: Phrase and topic discovery, with an application to Information Retrieval. In *Proc. of ICDM*, pages 697 –702, 2007.

[40] Y. Wang, E. Agichtein, and M. Benzi. TM-LDA: Efficient online modeling of latent topic transitions in social media. In *Proc. of KDD*, pages 123–131, 2012.

[41] Q. Wu, C. Zhang, and X. An. Topic segmentation model based on ATNLDA and co-occurrence theory and its application in stem cell field. *Journal of Information Science*, pages 1–14, 2012.

[42] Z. Yan and F. Li. News thread extraction based on Topical N-gram model with a background distribution. In *Proc. of ICONIP*, pages 416–424, 2011.