

A Statistical Model for Topic Segmentation and Clustering

M. Mahdi Shafiei and Evangelos E. Milios

Faculty of Computer Science, Dalhousie University
shafiei@cs.dal.ca, eem@cs.dal.ca
<http://www.cs.dal.ca/~shafiei>

Abstract. This paper presents a statistical model for discovering topical clusters of words in unstructured text. The model uses a hierarchical Bayesian structure and it is also able to identify segments of text which are topically coherent. The model is able to assign each segment to a particular topic and thus categorizes the corresponding document to potentially multiple topics. We present some initial results indicating that the word topics discovered by the proposed model are more consistent compared to other models. Our early experiments show that our model clustering performance compares well with other clustering models on a real text corpus, although they do not provide topic segmentation. Segmentation performance of our model is also comparable to a recently proposed segmentation model which does not provide document clustering.

1 Introduction

Using statistical models for modeling text corpora has received a lot of attention in recent years. These models can provide a compact description of documents in a corpus, which has been one of the main goals of the research community. Availability of such descriptions will make processing of increasingly large collections of text more efficient while preserving the essential statistical properties of the collection. The output will then be useful for basic tasks such as classification, novelty detection, summarization, and similarity and relevance judgements.

Statistical topic models are generative models for text. The basic idea behind all proposed topic models [4] is that a document is a mixture of several topics where each topic is some distribution over words. Each topic model is a generative model which specifies a simple probabilistic process by which the words in a document are being generated on the basis of a small number of latent variables.

Using standard statistical techniques, one can invert the process and infer the set of latent variables responsible for generating a given set of documents [16]. Assuming a model for generating the data, the goal of fitting this generative model is to find the optimal set of latent variables that can explain the observed data (i.e., observed words in documents). These latent variables capture the correlations between words and are referred to as topics. The direct output of these models, from an application point of view, is a set of overlapping clusters of words. Each of these clusters can be visualized by the the most probable words from their corresponding probability distribution. Clustering documents can be viewed only as a byproduct of the model fitting process and not as a direct output of topic models.

Latent Dirichlet Allocation (LDA) [4] is one of the highly cited works in topic modeling. In LDA, documents are assumed to be sampled from a random mixture over latent topics, where each topic is characterized by a distribution over words. Furthermore, the mixture coefficients are also assumed to be random and by assuming a prior probability on them, LDA provides a complete generative model for the documents [4].

The LDA model has been criticized for its inability to capture correlations between word topics which are common in natural text. A document about environment is more likely to be also about health than religion. In the LDA model, the topic proportions are derived from a Dirichlet distribution and hence are nearly independent. Several models have been proposed to capture the correlation between topics [3,8,14]. In the latter two models [8,14], the concept of topics is extended to include not only the distributions over words, but also distributions over topics. By assuming this, we allow some topics to be mixtures of other topics, thereby capturing the correlation between them. Using this idea, in our previous work [14], we proposed a hierarchical Bayesian model capable of clustering words and documents simultaneously and capturing correlation between word topics.

Splitting a text stream into coherent and meaningful segments is referred to as topic segmentation. In text segmentation, we are looking for the points in text at which focus shifts from one topic to another. For example, a news broadcast usually covers several stories or articles and therefore can be divided naturally into several pieces, each topically different. Topic segmentation is a preprocessing step for several other problems, such as topic detection and tracking in unstructured text. Information produced by a topic segmentation system can be used in summarization, browsing and facilitating the process of retrieving information buried in text data. The results can also be used to provide more informative responses for a search query by using the segmentation output for better navigation and scanning of the results by the user.

In this paper, we propose a statistical model for topic modeling and segmentation. The main contributions of the paper can be summarized as follows:

1. We propose a generative model which is able to segment text data into topically coherent segments while discovering the topic distributions over words.
2. The proposed model using a hierarchical structure is able to capture correlations between word topics.
3. The proposed model provides overlapping clustering of the documents.

The rest of this paper is organized as follows. In section 2, we briefly discuss the previous approaches to topic segmentation and modeling problems. We present our proposed model in section 3 and explain inference and parameter estimation algorithms for the model. In section 4, a set of experiments are provided to show the performance of the proposed model and compare it to related approaches. Finally, we conclude the paper with a review of the paper and discussion on future works.

2 Topic Segmentation and Identification

“Topic models” and “topic segmentation” are two closely related problems. Nevertheless, they have been often approached independently. In this section, we examine the possibility for treating these two problems in a single framework.

LDA is heavily dependent on the “bag-of-words” assumption. In models based on this assumption, the order of words and therefore the information implicit in that ordering is ignored for the sake of simplifying the model and avoiding computational complexity. Several recent works have tried to overcome this limitation [18,17]. This assumption is on the very finest level of a document structure, namely words. Two consecutive words are assumed to be topically independent whereas in reality the contrary is true. The topical dependency is also true for higher levels of text structure such as sentences and paragraphs. Actually, the dependencies in the higher levels originate from the dependencies in the word level. Therefore, one way of going beyond the “bag-of-words” assumption without complicating our model is to model these higher level structural dependencies.

This means that one can assume that a text document is composed of some topically correlated segments where each of these segments is a sequence of words. The “bag-of-words” assumption is still considered valid for the words in each of these segments but one hopes that, by capturing the higher level correlations (among segments), some of the finer level correlations (among words) are also captured. This idea makes topic segmentation a closely related and relevant problem. It suggests that tackling topic modeling and topic segmentation in a single framework provides a solution for going beyond the “bag-of-words” assumption.

Many of the existing topic segmentation algorithms are based on the idea that topic segments tend to be lexically cohesive. In lexical cohesion models, it is assumed that a shift in term distribution indicates a shift in topic. The most notable algorithm based on this assumption [7] uses a sliding window over text and uses a vector space representation of the text under the window. At each step, the term distribution for the text under the window is compared to the left and right regions of the window. The algorithm assigns a score to each topic boundary candidate based on a similarity measure between chunks of words appearing to the left and right of the candidate. Topic boundaries are then represented by the local minima points in the curve formed by these scores. These points are then adjusted to coincide with known paragraph boundaries.

3 Hierarchical Topic Segmentation and Detection Model

The driving idea for the proposed model is that human generated text seems to be composed of topically coherent segments put together. Each of the segments specifically is concerned with a more or less general topic. This topic can be modeled using statistical topic modeling approaches. One usually expects consecutive segments to be topically correlated. It means that considering a topically coherent segment, the next segment should convey a closely related topic as its predecessor. This is an observation similar to the one for words which questions the validity of “bag-of-words” assumption. Although a principal assumption for many statistical models of language, it is not a realistic one. Instead, we assume that the “bag-of words” assumption within each segment is fairly realistic, unlike for the whole document.

In this work, a model is proposed which is able to detect the boundaries of these segments. Each segment is assigned to a topic from a predefined number of topics which are referred to as “document-topics” or “supertopics” hereafter. Then, each segment is

modeled based on its word content similar to most probabilistic topic models. These learnt topics on words which are referred to as “word-topics” or simply topics are used to represent document topics. Each document-topic is assumed to be a mixture of word-topics where the mixture coefficients uniquely specify the document-topic.

Our work follows our previous work [14] on clustering documents and words simultaneously. To model the relation between topics of consecutive sentences or paragraphs, we assume a Markov structure on the distribution over document-topics. We assume that it is very likely for a sentence (or a paragraph) to have the same distribution over document-topics as its previous sentence. Otherwise, we sample a new distribution for the document-topic of this sentence. Our model also reduces to the model proposed in [13] if the text segments considered to be speech discourse. Moreover, our model is able to capture the correlations between higher level topics which is not seen in the model proposed in [13].

3.1 The Proposed Hierarchical Bayesian Model

Each document consists of different structural components or units like words, sentences and paragraphs. The proposed model can work with any of these structural components. For this section, we assume that this component is chosen to be sentences of the document. We order sentences of each document and assume a Markov structure on the topic distributions of sentences: with high probability, the topic for sentence i is the same as for sentence $i - 1$; otherwise we sample a new topic for it. We call the topics assigned to sentences “document-topics” or “super-topics”. We consider a switching binary variable for the topic of each sentence, indicating whether the topic for the current sentence is the same as the one for its predecessor. If we consider the states for all these switching variables, they will define a segmentation for the given document. We can achieve different levels of granularity for segmentation by choosing different types of structural units (i.e. words, sentences or paragraphs).

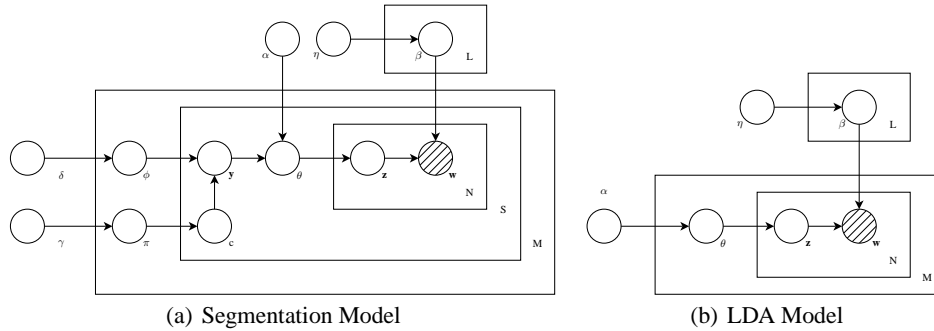


Fig. 1. Segmentation Model compared to the LDA Model

The proposed generative probabilistic model is shown as a graphical model in Fig. 1.a. Plate notation [5] is a standard and convenient way of illustrating probabilistic models with repeated sampling steps. In this graphical notation, shaded and unshaded variables indicate observed and latent (i.e., unobserved) variables respectively.

The segmentation model assumes the following generative process for each document d in a corpus D (intuitive explanations of model parameters are given in the text following the overview of the generative process.):

1. Choose $S \sim Poisson(\mu)$: number of sentences in the document
2. Choose $\phi \sim Dir(\delta)$
3. For each of the S sentences s
 - (a) Choose the same supertopic y_s for s as its previous sentence y_{s-1} with probability $p(c_s = 1) = \pi$
 - (b) Otherwise, choose a supertopic for the sentence $y_s \sim Multinomial(\phi)$
 - (c) Choose $N_s \sim Poisson(\varepsilon)$: number of words in the sentence
 - (d) Choose $\theta_s \sim Dir(\alpha, y_s)$
 - (e) For each of the N_s words w_{sn}
 - i. Choose a topic $z_{sn} \sim Multinomial(\theta_s)$: we call these topics “word-topics”
 - ii. Choose a word w_{sn} from $P(w_{sn}|z_{sn}, \beta)$, a Multinomial probability conditioned on the topic z_{sn}

Because we are interested in changes of topics, c_s indicates whether a change in topic occurs for sentence s . If $c_s = 0$, then $y_s = y_{s-1}$. Otherwise, y_s is drawn from a Multinomial distribution with parameter ϕ . The distribution is given in Eq. 1:

$$p(y_s|c_s, \phi, y_{s-1}) = \begin{cases} \delta(y_s, y_{s-1}) & c_s = 0 \\ Multinomial(\phi) & c_s = 1 \end{cases} \quad (1)$$

This distribution is not well-defined for the first sentence. Therefore, like in [13], we set $c_1 = 1$ and draw the first sentence topic from $Multinomial(\phi)$.

The word probabilities are modeled conditioned on the topics with a $L \times V$ matrix β where $\beta_{ij} = p(w^j = 1|z^i = 1)$. We assume a Dirichlet prior for drawing the parameters of word distribution. ϕ represents the mixing proportion of document-topics in a document. It specifies the parameters of the K -dimensional Multinomial distribution from which the model draws samples for document topics. θ_s is a sample from the Dirichlet distribution and specifies the mixing proportion of word-topics in the text segment s . Note that this mixing proportion depends on the supertopic that the current text segment is generated from. The model assumes that each document-topic is a mixture of several word-topics and this fact is modeled through the matrix of hyperparameters α that will be estimated in the learning phase.

3.2 Inference and Parameter Estimation

The inference problem is to compute the posterior probability of hidden variables given the input parameters $\alpha, \eta, \delta, \gamma$ and observations \mathbf{w} :

$$p(\pi, \mathbf{c}, \phi, \mathbf{y}, \theta, \mathbf{z}|\mathbf{w}, \alpha, \eta, \delta, \gamma) = \frac{p(\pi, \mathbf{c}, \phi, \mathbf{y}, \theta, \mathbf{z}, \mathbf{w}|\alpha, \eta, \delta, \gamma)}{p(\mathbf{w}|\alpha, \eta, \delta, \gamma)} \quad (2)$$

For the models in the LDA family, exact inference is intractable. Therefore, approximation methods have been proposed to do the inference. Model parameters can be

theoretically estimated using EM based algorithms but these algorithms often face local optima problems for models in this family. Therefore, by adopting a Bayesian approach, we use methods in which some of the hidden parameters are integrated out instead of being explicitly estimated. By using conjugate priors on the model parameters, this task becomes much more mechanical and straightforward. Integrating out some parameters also simplifies the sampling process, explained next. In our model, we need to integrate out the parameters β , ϕ and δ .

Gibbs sampling like other members of the Markov chain Monte Carlo (MCMC) algorithms family is an iterative method used to draw samples from complex and usually high dimensional distributions. Each iteration of the algorithm gives a sample from the target distribution in the long run. In each iteration of the Gibbs sampling method, variables are divided into blocks and each block is sampled from its conditional distribution conditioned on the current values of all other random variables of the target distribution. This process is performed sequentially and continues until the sampled values approximate the target distribution.

For our model, the target distribution is the posterior distribution of word-topics, document-topics and topic-switching variables given the collection of documents. This is an intractable distribution and sampling from it is difficult. By using Gibbs sampling, in each iteration, we sample from the conditional distribution of a single word-topic in a document conditioned on the topic assignment for all other words and sentences in all documents except the current word. We also sample from the conditional distribution of a single document-topic for a text segment and its corresponding switching variable given that the topic assignments of all other words not in the current sentence, topic assignments of all other sentences and all other switching variables values are known.

We order the documents in the corpus randomly and each document is given an index according to its position in this list. We represent the corpus with three list of indices: word indices wl , sentence indices pl and document indices dl (As mentioned earlier, one can use paragraphs or any other well-defined structural unit of text instead). wl_i denotes the index of the i th word in the sequence of words (if we assume the whole corpus as a sequence of words fed to the algorithm). dl_i is the document index and pl_i represent the sentence index of the corresponding word respectively. Note that the purpose of the model is putting together these structural segments of the text (chosen by user preference) and forming sequences of topically coherent segments. These lists will then be fed to the Gibbs Sampling algorithm. For each word token, the Gibbs sampling algorithm estimates the probability of assigning the current word to word-topics given assignment of all other words to word-topics from the corresponding conditional distribution that we will derive shortly. Then the current word would be assigned to a word-topic and this assignment will be stored for reference when the Gibbs sampling algorithm works on other words.

While scanning the list of words, we watch for new sentences (or the structural unit chosen by the user) as they start. For each such new structural segment, the Gibbs sampling algorithm decides whether this segment should have the same topic as the preceding topic or it should be assigned to a new topic. In the latter case, the Gibbs sampler estimates the probability of assigning this sentence to document-topics given assignments of all other sentences to document-topics. These probabilities are com-

puted from the corresponding conditional distribution for a sentence given all other topic assignments to every other sentence and all words not in this sentence. Then the new sentence would be assigned to a document-topic.

In our case we need to compute the two conditional distributions $p(z_{dsn}|z_{-dsn}, c, y, w)$ and $p(y_{ds}, c_{ds}|z, y_{-ds}, c_{-ds}, w)$, where z_{dsn} represents the word-topic assignment for word w_{dsn} (word n in document d and sentence s) and z_{-dsn} denotes the word-topic assignments for all other words except the current word w_{dsn} . y_{ds} denotes the document-topic assignment for sentence p_{ds} in document d and y_{-ds} represents the document-topic assignments for all sentences except the current sentence p_{ds} . Beginning with the joint probability of a dataset, and using the chain rule, we can obtain the conditional probabilities conveniently. For our model, we obtain equations 3 and 4.

$$p(z_{dsn}|z_{-dsn}, c, y, w) = \frac{(\alpha_{y_{ds}z_{dsn}} + n_{z_{dsn}}^{(ds)})}{\sum_{l=1}^L (\alpha_{y_{ds}l} + n_l^{(ds)})} \times \frac{n_{z_{dsn}w_{dsn}} + \eta_{w_{dsn}} - 1}{\sum_{v=1}^V n_{z_{dsn}v} + \eta_v - 1} \quad (3)$$

$$p(y_{ds}, c_{ds}|z, y_{-ds}, c_{-ds}, w) = \quad (4)$$

$$\begin{cases} \frac{n_{d_0} + \gamma}{N_d + 2\gamma} \times \frac{\Gamma(\sum_{l=1}^L \alpha_{y_{ds}l})}{\prod_{l=1}^L \Gamma(\alpha_{y_{ds}l})} \times \frac{\prod_{l=1}^L \Gamma(\alpha_{y_{ds}l} + n_l^{(ds)})}{\Gamma(\sum_{l=1}^L \alpha_{y_{ds}l} + n_l^{(ds)})} & \text{if } c_{ds}=0 \text{ \& } s>1 \text{ \& } y_{ds}=y_{d(s-1)} \\ \frac{n_{d_1} + \gamma}{N_d + 2\gamma} \times \frac{\delta_k + n_{y_{ds}}^d}{\sum_{k=1}^K (\delta_k + n_k^d)} \times \frac{\Gamma(\sum_{l=1}^L \alpha_{y_{ds}l})}{\prod_{l=1}^L \Gamma(\alpha_{y_{ds}l})} \times \frac{\prod_{l=1}^L \Gamma(\alpha_{y_{ds}l} + n_l^{(ds)})}{\Gamma(\sum_{l=1}^L \alpha_{y_{ds}l} + n_l^{(ds)})} & \text{if } c_{ds} = 1 \\ 0 & \text{otherwise} \end{cases}$$

where $n_l^{(ds)}$ represents how many times a word in sentence s of document d has been assigned to topic l . $n_{lw_{dsn}}$ represents the total number of times that the word w_{dsn} has been assigned to topic l . n_k^d is the number of times a sentence in document d has been assigned to document-topic k . n_{d_0} and n_{d_1} are the number of times that the switching variable c is set to be 0 and 1 respectively.

In most of the statistical topic models inspired by the LDA model, the Dirichlet parameters α are assumed to be given and fixed, which still gives reasonable results. But for the proposed model, as in [14,8], these parameters are a very important part of the model. These parameters determine how the correlations between different word topics through their participation in document topics are captured by the model. For estimating parameters of a Dirichlet distribution, a family of approaches based on maximum likelihood or maximum a posteriori estimation of parameters has been proposed in the literature [11]. There is no closed-form solution for these methods and one should use iterative methods to learn the parameters. But these iterative methods are often computationally expensive and other methods like moment matching [11] are used to approximate the parameters of the Dirichlet prior α . We will also use this approach for our model. This means that in each iteration of Gibbs sampling, we update

$$\begin{aligned} var_{kl} &= \frac{1}{N_k} \sum_{s \in S_k} \left(\frac{n_l^{(s)}}{n^{(s)}} - mean_{kl} \right)^2, & mean_{kl} &= \frac{1}{N_k} \sum_{s \in S_k} \frac{n_l^{(s)}}{n^{(s)}}, & \alpha_{kl} &\propto mean_{kl} \\ m_{kl} &= \frac{mean_{kl}(1 - mean_{kl})}{var_{kl}} - 1, & \sum_{l=1}^L \alpha_{kl} &= \exp\left(\frac{\sum_{l=1}^L \log(m_{kl})}{L - 1}\right) \end{aligned} \quad (5)$$

where S_k represents the set of sentences assigned to document-topic k and N_k is the number of sentences assigned to document-topic k . $n_l^{(s)}$ represents the number of times a word in sentence s has been assigned to word-topic l . $n^{(s)}$ is the number of words in sentence s . Note that for $mean_{kl}$ and var_{kl} , we only consider the sentences assigned to document-topic k . For each document-topic k , we first compute sample mean $mean_{kl}$ and sample variance var_{kl} . They are computed over all sentences assigned to document-topic k .

Algorithm 1 shows the pseudocode for the Gibbs sampling process for our model.

Algorithm 1: LDCC Segmentation Gibbs Sampling

Input: $\gamma, \delta, \alpha, \eta, L, K, \text{Corpus}, \text{MaxIteration}$

Output: topic assignments for all words and sentences in the Corpus

- 1 Initialization: Randomly, initialize the word-topic assignments for all word tokens and document topic assignments and topic switch variables for all sentences
 - 2 Compute P_{dk} for all values of $k \in \{1..K\}$ and all documents
 - 3 Compute n_{lv} for all values of $l \in \{1..L\}$ and all word tokens
 - 4 Compute $n_l^{(ds)}$ for all values of $l \in \{1..L\}$ and all documents and their sentences
 - 5 **if doing parameter estimation then**
 - 6 Initialize *alpha* parameters using Eq. 5
 - 7 Randomize the order of documents in the corpus
 - 8 Randomize the order of sentences in each document
 - 9 Randomize the order of words in each sentence
 - 10 **for** $iter \leftarrow 1$ **to** MaxIteration **do**
 - 11 **foreach** word i according to the order **do**
 - 12 Exclude word i and its assigned topic l from variables $n_l^{(ds)}$ and n_{li}
 - 13 $newl =$ sample new word-topic for word i using Eq. 3
 - 14 Update variables $n_l^{(ds)}$ and n_{li} using the new word-topic $newl$ for word i
 - 15 **if entered a new sentence** j **then**
 - 16 Exclude sentence j and its assigned topic k from variable P_{dk}
 - 17 $(newk, newc) =$ sample new document-topic and the switching variable for sentence j using Eq. 4
 - 18 **if** $newc == 1$ **then**
 - 19 Assign $newk$ as the new document-topic for sentence j ;
 - 20 Update variable P_{dk} using the new document-topic $newk$ for sentence j
 - 21 **if doing parameter estimation then**
 - 22 Update *alpha* parameters using Eqs. 5
-

4 Experimental Results

In the following sections, we present early results, indicating that our model is able to discover topics that are more coherent compared to the LDA model for two different datasets. We also show that the proposed model outperforms our previous model which does not have the ability to detect segments in the text. We finally compare the segmentation performance of our model with a recently proposed model that has comparable performance compared to some of the well-known topic segmentation algorithms.

4.1 Datasets

We use two real-world datasets in our experiments. Our first dataset is a subset of the Wikipedia XML corpus¹ [6]. This subset contains 261 articles categorized in 5 overlapping classes, namely "Music", "Art", "Archaeology", "Christianity" and "Spirituality". Each document belongs to 2.09 classes on average. The biggest class corresponds to "Art" with 179 documents. The smallest class, "Archaeology" has 63 documents. We removed all the words which occurred in less than 5 documents from the list of final word tokens. We also used a list of standard "stopwords" and deleted all numbers, words with length less than 3 and having non-ASCII characters. We do not consider paragraphs with less than 5 words and do not include documents with less than 3 paragraphs. We do not have the tags for separating words, therefore we used all delimiting characters to separate words. There are 65978 word tokens, 2740 paragraphs and 2311 unique words after preprocessing.

Our second dataset has been used in previous studies [9,10] for evaluating the segmentation results. It consists of spoken lecture transcripts from an undergraduate physics class and a graduate artificial intelligence class. A typical 90 minutes lecture has 500 to 700 sentences and over 8500 words. The segmentation of the lecture transcripts are done manually to facilitate access to lecture recordings available on the class website for students. It is aimed to convey the high-level topical structure of the lectures. Each lecture is annotated with six segments on average. The second part of the dataset corresponding to the AI class has on average 12 segments for each lecture.

4.2 Word-Topic Examples

In this section, we show 10 and 9 word-topics derived from the Wikipedia dataset using our model which we call "LDSEG" hereafter, and the LDA model respectively. The topics are represented each with their 10 most indicative words and are presented in Figs 2 and 3. The topics were derived by assuming the number of document-topics equal to 4 and the number of word-topics equal to 50. Among the 50 word-topics discovered by each model, we are able to match 32 topics. Topics 1 and 2 in Figs 2 and 3 are two examples. Although the early inspection of the results for the examined models shows that the topics discovered by our model are more coherent. As an example, topic 1 discovered by the LDA model has some words that do not fit in the topic while for the corresponding topic by our model, all the top indicative words compose a more coherent topic. We are able to observe this in most of the topics.

Our model also seems to be able to discover some topics that the LDA model do not detect. Examples of such topics can be seen in the topics 6 and 7 of our model. Topic 3 of our model is especially interesting. It is a topic about "bigfoot" and all the indicative words are the names related to this topic. For the LDA model, we have the word "bigfoot" mixed in the topic 2 which does not make sense. Instead of these topics, the LDA model has some other words put together as a topic that do not strongly indicate a topic. One can see examples of this topic in the topics 4 and 8 discovered by the LDA model. Our model is also able to split some of the more general topics into

¹ it is available for download at <http://www-connex.lip6.fr/~denoyer/wikipediaXML> by registration

more specific ones. For example, topic 3 discovered by the LDA is about Christianity and Jesus whereas in our model, the same topic is split into two more specific topics 4 and 5.

topic 1	topic 2	topic 3	topic 4	topic 5	topic 6	topic 7	topic 8	topic 9	topic 10
alexander	women	nietzsche	church	god	radio	economic	league	computer	house
greek	sexual	bigfoot	catholic	church	day	million	football	game	parliament
ancient	family	dragon	orthodox	christ	television	government	team	system	commons
apollo	men	sasquatch	christian	jesus	year	economy	game	games	lords
gods	male	evidence	churches	baptism	abc	company	club	apple	members
hecate	female	campbell	saint	life	show	world	season	atari	act
mythology	children	verne	roman	christian	advertising	international	home	software	bill
earth	members	krantz	council	heaven	broadcast	trade	won	data	kingdom
archaeology	people	wallace	century	holy	bbc	growth	cup	commodore	ireland
hermes	gay	friedrich	religious	faith	time	development	world	disc	england

Fig. 2. Example word-topics for the Wikipedia dataset discovered by our model

topic 1	topic 2	topic 3	topic 4	topic 5	topic 6	topic 7	topic 8	topic 9
greek	women	church	lennon	league	computer	house	des	russell
alexander	sexual	god	xavier	football	game	parliament	disk	theory
graffiti	family	jesus	voting	team	games	commons	linear	nietzsche
apollo	children	christ	peel	game	apple	lords	data	human
khazar	female	orthodox	john	club	system	members	group	philosophy
gods	male	christian	godzilla	season	atari	act	lie	work
mythology	gay	baptism	contest	cup	commodore	khmer	audio	life
hecate	men	catholic	costas	world	design	bill	vector	social
khazars	feminism	churches	borda	boxing	software	government	omega	rousseau
ancient	bigfoot	saint	candidate	teams	video	rouge	space	political

Fig. 3. Example word-topics for the Wikipedia dataset discovered by the LDA model

4.3 Document Clustering Performance

The Latent Dirichlet Co-Clustering (LDCC) model proposed in our previous work [14], has been shown to have better clustering results compared to several other models including Model-based Overlapping Co-Clustering [15], Model-based Overlapping Clustering [1] and K-Means algorithm in terms of precision, recall and F-measure. We are interested to see if the proposed model in this paper can deliver as our previous work [14] in terms of clustering performance, considering the fact that it is not given any information about the segmentation.

We use the Wikipedia dataset for measuring the clustering performance. In order to compare clustering results, we use precision, recall, and F-measure calculated over pairs of points, as defined in [1]. For each pair of points that share at least one cluster in the overlapping clustering results, these measures try to estimate whether the prediction of this pair as being in the same cluster was correct with respect to the underlying true categories in the data. Precision is calculated as the fraction of pairs correctly put in the same cluster, recall is the fraction of actual pairs that were identified, and F-measure is the harmonic mean of precision and recall.

Table 1 presents the results of LDSEG versus LDCC algorithm in terms of precision, recall and F-Measure for the Wikipedia Corpus. Each reported result is an average over 50 samples. We take the average results for the number of word-topics varying

		LDSEG			LDCC		
K	L	Precision	Recall	F-Measure	Precision	Recall	F-Measure
4	10-80	0.734	0.488	0.585	0.727	0.508	0.594
6	10-80	0.718	0.435	0.538	0.717	0.423	0.526

Table 1. Clustering results of LDCC and LDSEG algorithms on Wikipedia dataset.

between 10 and 80. Table 1 contains the results for two different values for the number of super-topics, 4 and 6. The results show that although no information about segmentation is given to the LDSEG model, it is still comparable to the LDCC model in terms of precision, recall and F-Measure.

4.4 Segmentation Results

We use two standard error rate metrics for comparing the segmentation results of our model and the model proposed in [13]. P_k [2] is the probability that two segments drawn randomly from a document are incorrectly identified as belonging to the same topic. WinDiff [12] moves a sliding window across the text and counts the number of times the hypothesized and reference segment boundaries are different within the window. For both these measures, lower values indicate better agreement with a gold standard segmentation.

Segmentation performance of the model introduced in [13] is comparable with previous segmentation algorithms. Thus we compare our model with this model. We use our second dataset which has human-annotated segmentation for evaluating our model performance. Table 2 shows that our model is comparable with the model in [13] for different numbers of word-topics.

Model	LDSEG		Purver et. al.		
	L	Pk	WinDiff	Pk	Windiff
20	0.405	0.431	0.413	0.420	
30	0.407	0.432	0.416	0.421	
40	0.419	0.450	0.416	0.421	

Table 2. Segmentation results of LDSEG and Purver’s [13] algorithms on our second dataset.

4.5 Conclusions and Future Work

We have proposed a hierarchical Bayesian model that combines topic identification and segmentation in text document collections. The proposed model is able to cluster the documents in the dataset into overlapping clusters. Extracted word-topics seem to be more coherent compared to LDA. Segmentation and clustering performance of the model is comparable to some recently introduced models, although those models are simpler and lack some of the features of our model.

We plan to do more experiments using other datasets to compare our model with other topic segmentation and clustering algorithms. We also like to try our model as an early preprocessing part for some other tasks including text summarization and translation. For learning parameter α in our model, we currently use moment-matching

method. We plan to take a fuller Bayesian approach by assuming the parameter α a random variable. Assuming a prior on α allows the model to automatically select values for this hyper-parameter .

References

1. A. Banerjee, C. Krumpelman, S. Basu, R. Mooney, and J. Ghosh. Model based overlapping clustering. In *International Conference on Knowledge Discovery and Data Mining (KDD)*, August 2005.
2. Doug Beeferman, Adam Berger, and John Lafferty. Statistical models for text segmentation. *Machine Learning*, 34(1-3):177–210, 1999.
3. David Blei and John Lafferty. Correlated topic models. In Y. Weiss, B. Schölkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems 18*, pages 147–154. 2006.
4. David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.
5. Wray L. Buntine. Operations for learning with graphical models. *Journal of Artificial Intelligence Research (JAIR)*, 2:159–225, 1994.
6. Ludovic Denoyer and Patrick Gallinari. The Wikipedia XML Corpus. *SIGIR Forum*, 2006.
7. Marti A. Hearst. Texttiling: segmenting text into multi-paragraph subtopic passages. *Comput. Linguist.*, 23(1):33–64, 1997.
8. Wei Li and Andrew McCallum. Pachinko allocation: Dag-structured mixture models of topic correlations. In *ICML '06: 23rd International Conference on Machine Learning*, Pittsburgh, Pennsylvania, USA, June 2006.
9. Igor Malioutov and Regina Barzilay. Minimum cut model for spoken lecture segmentation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 25–32, July 2006.
10. Igor Malioutov, Alex Park, Regina Barzilay, and James Glass. Making sense of sound: Unsupervised topic segmentation over acoustic input. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 504–511, June 2007.
11. T. P. Minka. Estimating a Dirichlet distribution. Technical report, MIT, 2000.
12. Lev Pevzner and Marti A. Hearst. A critique and improvement of an evaluation metric for text segmentation. *Comput. Linguist.*, 28(1):19–36, 2002.
13. Matthew Purver, Konrad Kording, Thomas Griffiths, and Joshua Tenenbaum. Unsupervised topic modelling for multi-party spoken discourse. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics (COLING-ACL)*, pages 17–24, July 2006.
14. M. Mahdi Shafiei and Evangelos E. Milios. Latent dirichlet co-clustering. In *ICDM '06: Proceedings of the Sixth International Conference on Data Mining*, pages 542–551, 2006.
15. Mahdi Shafiei and Evangelos Milios. Model-based overlapping co-clustering. In *Proceedings of the Fourth Workshop on Text Mining, Sixth SIAM International Conference on Data Mining*, Bethesda, Maryland, April 22 2006.
16. M. Steyvers and T. Griffiths. Probabilistic topic models. In T. Landauer, D. Mcnamara, S. Dennis, and W. Kintsch, editors, *Latent Semantic Analysis: A Road to Meaning*. Laurence Erlbaum, 2005.
17. Hanna M. Wallach. Topic modeling: beyond bag-of-words. In *ICML '06: Proceedings of the 23rd international conference on Machine learning*, pages 977–984, 2006.
18. Xuerui Wang and Andrew McCallum. A note on topical n-grams. Technical Report UM-CS-2005-071, University of Massachusetts Amherst, December 2005.