Unsupervised Segmentation of Bibliographic Elements with Latent Permutations

Tomonari Masada[1], Yuichiro Shibata[1] and Kiyoshi Oguri[1]

Nagasaki University, 1-14 Bunkyo-machi, Nagasaki, 852-8521 Japan
{masada,shibata,oguri}@nagasaki-u.ac.jp
http://www.cis.nagasaki-u.ac.jp/~masada/

Abstract. This paper introduces a novel approach for large-scale unsupervised segmentation of bibliographic elements. Our problem is to segment a word token sequence representing a citation into subsequences each corresponding to a different bibliographic element, e.g. authors, paper title, journal name, publication year, etc. Obviously, each bibliographic element should be represented by *contiguous* word tokens. We call this constraint *contiguity constraint*. Therefore, we should infer a sequence of assignments of word tokens to bibliographic elements so that this constraint is satisfied. Many HMM-based methods solve this problem by prescribing fixed transition patterns among bibliographic elements. In this paper, we use generalized Mallows models (GMM) in a Bayesian multi-topic model, effectively applied to document structure learning by Chen et al. [4], and infer a *permutation* of latent topics each of which can be interpreted as one among the bibliographic elements. According to the inferred permutation, we arrange the order of the draws from a multinomial distribution defined over topics. In this manner, we can obtain an ordered sequence of topic assignments satisfying contiguity constraint. We do not need to prescribe any transition patterns among bibliographic elements. We only need to specify the number of bibliographic elements. However, the method proposed by Chen et al. works for our problem only after introducing modification. The main contribution of this paper is to propose strategies to make their method work also for our problem.

1 Introduction

Multi-topic modeling, inaugurated by the proposal of latent Dirichlet allocation (LDA) [2], provides successful solutions to many applications. In this paper, we use multi-topic modeling for clustering word tokens so that the same cluster (i.e., the same topic) correspond to the same real-world category.

In this paper, we consider segmentation of bibliographic elements. It is assumed that each citation data is represented as a sequence of untagged word tokens. Our problem is to assign each word token to a topic so that the word tokens assigned to the same topic refer to the same bibliographic element, e.g. authors, paper title, journal name, publication year, etc. We solve this problem in an *unsupervised* manner. We use no knowledge about transition patterns

Fig. 1. An example of segmentation our method provides for DBLP dataset (cf. Table 1). Each line corresponds to a different citation, and long lines are cut off at the right side to present more citations with larger fonts. Each subsequence separated by ♦ corresponds to the assignment to a different topic inferred by our method. In our experiment, the number of topics is set to be larger than the number of bibliographic elements by one. In this example, the number of topics is five, because the number of bibliographic elements is four. Those four elements are: author names, paper title, conference name (or journal name), and publication year.

among bibliographic elements. Further, bibliographic elements do not need to be identified beforehand. We only assume that the number of different bibliographic elements is known. The number of topics can be set to be larger than that of different bibliographic elements, because we can identify multiple topics with the same bibliographic elements when we interpret the topic assignments provided by our method. Figure 1 gives an example of segmentation obtained by our method in the experiment whose details will be explained later.

Our target data is a set of citations obtained, for example, after an OCR processing of the reference section of printed papers. While correction of OCR errors is important and may be realized by introducing extensions to our model as Takasu [10] did for HMM, we regard it as future work. In this paper, we concentrate on segmentation of bibliographic elements by assuming that OCR errors are already corrected. Further, publication data presented on the Web by researchers can also be regarded as our target data, because most of such data are presented not as a segmented data, e.g. in BibTeX format, but just as a sequence of untagged word tokens.

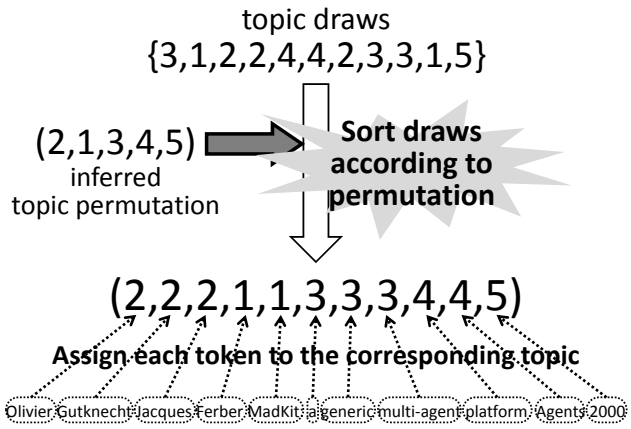

Fig. 2. How to obtain a sequence of topic assignments satisfying contiguity constraint by inferring a topic permutation. In the resulting topic sequence, we can interpret, for example, topic 2 as representing author names, topic 5 as publication year, etc.

In any solution to our problem, each bibliographic element should be referred to by *contiguous* word tokens. In other words, the word tokens referring to the same bibliographic element should not be separated by the word tokens referring to other elements. We call this constraint *contiguity constraint*. Many HMM-based methods put contiguity constraint by prescribing fixed transition patterns among bibliographic elements [3, 8, 10]. In contrast, we provide a more flexible answer where we infer a *permutation* of topics in multi-topic modeling. By inferring a topic permutation for each citation, we arrange the order of the topic draws according to the inferred permutation, where the number of the topic draws is the same with the word tokens included in each citation. In this manner, we obtain a sequence of topic draws satisfying contiguity constraint (see Figure 2). For the resulting topic sequences, we interpret each topic as one among the bibliographic elements to obtain a segmentation of bibliographic elements.

This paper shows that we can use a Bayesian probabilistic model proposed by Chen et al. [4] to solve our problem in the manner described above. We call their model *CBBK* by taking the initials of the authors' last names. While LDA [2] is a standard model for Bayesian multi-topic modeling, we cannot use LDA, because LDA gives topic assignments not satisfying contiguity constraint. CBBK can successfully put contiguity constraint on topic assignments by incorporating *generalized Mallows model* (GMM) [6] that defines a probability distribution over all permutations of topics. We can infer a topic permutation as a draw from this distribution whose parameters are fitted to the input data.

However, CBBK is devised by Chen et al. for *document structure learning*, a problem widely different from ours. In document structure learning, we are given a set of documents, each of which is regarded as a sequence of untagged *paragraphs*, and infer a semantic structure of each document by assigning each

paragraph to a topic so that contiguity constraint is satisfied. Then, document structure is recovered as a sequence of topics, where each topic is represented by a set of contiguous paragraphs. For example, the semantic structure of every academic paper can be recovered as an ordered set of sections, and each section is a set of contiguous paragraphs related to the same semantic content.

However, CBBK works for our problem only after introducing modification. In this paper, we regard each word token in citation data as a paragraph consisting of only one token. That is, we assign each word token to a topic as in LDA. Consequently, the unit for topic assignment loses richness in its semantic content, because the assignment unit is now a single word token. Chen et al. did not have a provision for using CBBK in this manner. They only considered the assignment of paragraphs, a semantic unit far larger than word tokens, to topics. Actually, Chen et al. only used documents whose paragraphs consist of tens of word tokens in their evaluation experiment. Therefore, we cannot know whether CBBK works for our problem based only on their results.

The main contribution of this paper is to show that CBBK can be applied to our problem, i.e., unsupervised segmentation of bibliographic elements, by using the following two strategies to modify the settings with which CBBK is applied:

1. We realize a *dense* topic distribution for each document (i.e., for each citation) by choosing an appropriate topic Dirichlet prior distribution.
2. We use *large-scale* datasets for capturing topical relatedness *across* documents and recoup the loss of content richness of topic assignment unit.

The latter strategy aims to fully utilize the advantage of CBBK, i.e., the advantage that topical relatedness across documents can be effectively captured, for example, when compared with BayesSeg [5] that processes each document only separately. In our case, each document is just a sequence of tens of untagged word tokens and gives almost no clue to segmentation when processed separately. Therefore, it is highly favorable that CBBK is applied to large-scale datasets for utilizing that advantage. We will discuss the former strategy after giving the details of Markov-chain Monte Carlo (MCMC) inference for our proposal.

The rest of the paper is organized as follows. Section 2 shows how we modify CBBK to realize segmentation of bibliographic elements. Section 3 gives the details of MCMC inference. Section 4 includes the settings and the results of our experiment. Section 5 concludes the paper with discussions and future work.

2 Model

2.1 Generalized Mallows Model

The key technology in CBBK is *generalized Mallows model* (GMM) [6]. GMM defines a probability distribution over all permutations of a fixed number of items. In CBBK, for each document, we draw as many topics as the paragraphs from a multinomial distribution defined over a fixed set of K topics $\{1, \ldots, K\}$[1].

[1] In this paper, we identify each topic with its ID number.

Further, we draw a permutation of these K topics from GMM. Then, by arranging the drawn topics according to the drawn permutation, we can obtain an ordered multiset of topics satisfying contiguity constraint. For example, when we draw topics as $\{3, 1, 2, 2, 4, 4, 2, 3, 3, 1, 5\}$ and draw a permutation $(2, 1, 3, 4, 5)$, we obtain an ordered topic multiset $(2, 2, 2, 1, 1, 3, 3, 3, 4, 4, 5)$ (see Figure 2). The obtained ordered multiset induces topic assignments of paragraphs. For example, the first paragraph is assigned to topic 2, the fifth paragraph to topic 1, etc.

When we use GMM, every permutation of K topics is represented as a $(K-1)$-dimensional vector $\mathbf{v} = (v_1, \ldots, v_{K-1})$ whose entries are non-negative integers called *inversion counts*. Each inversion count v_k satisfies $0 \leq v_k \leq K - k$ and tells how many topics appear before topic k among the $K - k$ topics larger than k, i.e., among $\{k+1, \ldots, K\}$. Note that each inversion count vector corresponds to a unique permutation. To the permutation represented as an inversion count vector $\mathbf{v} = (v_1, \ldots, v_{K-1})$, GMM gives the following probability:

$$p(\mathbf{v}|\rho) = \prod_{k=1}^{K-1} \frac{\exp(-\rho_k v_k)}{\psi_k(\rho_k)} \ , \tag{1}$$

where each ρ_k is a non-negative real parameter of GMM, and each $\psi_k(\rho_k)$ is a normalization constant obtained as

$$\psi_k(\rho_k) = \frac{1 - \exp\{-(K-k+1)\rho_k\}}{1 - \exp(-\rho_k)} \ , \tag{2}$$

which is a sum of a geometric series. The probability distribution of GMM in Eq. (1) admits the following conjugate prior distribution:

$$p(\rho_k|\gamma_k, \nu) \propto \exp\{-\gamma_k \nu \rho_k - \nu \log \psi_k(\rho_k)\} \ . \tag{3}$$

Throughout the paper, we set each hyperparameter γ_k as follows:

$$\gamma_k = \frac{1}{e^{\rho_0} - 1} - \frac{K-k+1}{e^{(K-k+1)\rho_0} - 1} \ . \tag{4}$$

This setting of each γ_k is recommended by [4] so as to fix the mode of the prior in Eq.(3) to the constant ρ_0. Further, we set $\rho_0 = 1$ and $\nu = 0.1$ as in [4].

GMM has the following special feature. As Eq. (1) shows, GMM gives a more or at least an equal probability to the case $v_k = 0$ when compared with the other strictly positive cases. In other words, GMM prefers permutations represented as an inversion count vector including many zeros. Note that the permutation represented as the zero vector is the identity permutation. Therefore, GMM is likely to give a large probability mass to the identity permutation and also to the permutations showing only a small deviation from the identity permutation. This feature of GMM is useful, because we can hope that there will be a unique canonical order of bibliographic elements in the given citation dataset as long as we fix the source of the data, e.g. the journals published by the same publisher.

2.2 Modifying CBBK

In this paper, we use CBBK for segmenting bibliographic elements. However, we have no paragraphs, because each citation is represented as a sequence of untagged word tokens. Therefore, we regard each word token as a paragraph consisting of only one word token and draw as many topics as the word tokens for each citation. Precisely, we modify CBBK as follows:

1. For each topic k, draw a word multinomial distribution $\texttt{Multi}(\phi_k)$, defined over the set of W different words, from the corpus-wide symmetric Dirichlet prior $\texttt{Dirichlet}(\beta)$.
2. Draw a GMM parameter ρ_k from the prior in Eq. (3) for each topic $k < K$ and obtain a GMM $\texttt{GMM}(\rho)$.
3. The jth citation $\mathbf{x}_j$ as a sequence of n_j word tokens $\mathbf{x}_j = (x_{j1}, \ldots, x_{jn_j})$ is generated as follows:
   (a) Draw a topic multinomial distribution $\texttt{Multi}(\theta_j)$ from the corpus-wide symmetric Dirichlet prior $\texttt{Dirichlet}(\alpha)$.
   (b) Draw n_j topics from the topic multinomial $\texttt{Multi}(\theta_j)$ and obtain an unordered multiset $\mathbf{t}_j$ of n_j topics.
   (c) Draw a permutation $\mathbf{v}_j$ of K topics from $\texttt{GMM}(\rho)$.
   (d) By ordering the topics in $\mathbf{t}_j$ according to $\mathbf{v}_j$, obtain an ordered topic multiset $\mathbf{z}_j = (z_{j1}, \ldots, z_{jn_j})$ satsifying contiguity constraint.
   (e) For each word token x_{ji}, $i = 1, \ldots, n_j$, draw a word w from the word multinomial $\texttt{Multi}(\phi_{z_{ji}})$ and set $x_{ji} = w$.

The generative process of modified CBBK now looks similar to LDA, because we assign not each paragraph but each word token to a topic. However, the topic assignment is affected by how drawn topics are ordered by the permutation drawn from GMM. Therefore, modified CBBK behaves quite differently from LDA in how word tokens are assigned to topics.

However, LDA and CBBK have an important common feature. Both models can intensively capture topical relatedness *across* documents. Both in LDA and in CBBK, per-topic word multinomials $\texttt{Multi}(\phi_1), \ldots, \texttt{Multi}(\phi_K)$ are shared by all documents. The two Dirichlet priors, $\texttt{Dirichlet}(\alpha)$ and $\texttt{Dirichlet}(\beta)$, are also shared. This feature differentiates CBBK from BayesSeg [5], which processes each document separately and thus captures no relatedness across documents.

3 MCMC Inference

We use MCMC inference described in [4] to infer posterior distributions. Each iteration consists of updates of GMM parameters, updates of inversion counts, and updates of topic draws. Further, we optimize the hyperparameter β of the word Dirichlet prior once per every iteration, though this optimization is not considered in [4]. The details of each part of the iteration are given below.

We draw GMM parameter ρ_k from the following conditional distribution:

$$p(\rho_k | \ldots) \propto \exp\left\{ -\left(\gamma_k \nu + \sum_j v_{jk}\right)\rho_k - (N + \nu) \log \psi_k(\rho_k) \right\}, \qquad (5)$$

where N is the number of documents and v_{jk} is the kth inversion count for the jth citation. We cannot analytically obtain the normalization constant for the distribution in Eq. (5). Therefore, a slice sampling is conducted. While Chen et al. [4] used MATLAB blackbox sampler, we implemented a customized sampler to achieve computational efficiency, because we target large-scale datasets.

For the jth citation, we draw each of the $K-1$ inversion counts $v_{j1}, \ldots, v_{jK-1}$ from the following conditional distribution:

$$p(v_{jk}^{\text{new}}|\ldots) \propto \frac{\exp(-\rho_k v_{jk}^{\text{new}})}{\psi_k(\rho_k)} \cdot p(\mathbf{x}_j|\mathbf{z}_j^{\text{new}}, \mathbf{x}_{-j}, \mathbf{z}_{-j}, \beta) \ . \tag{6}$$

The first half of the right hand side of Eq. (6) is the probability of a new inversion count coming from Eq. (1). The latter half is the conditional probability of the observed word token sequence $\mathbf{x}_j$ in the jth citation, where $\mathbf{x}_{-j}$ (resp. $\mathbf{z}_{-j}$) means the set of the observed word token sequences (resp. the set of the latent ordered topic multisets) for all citations except the jth citation. Further, $\mathbf{z}_j^{\text{new}}$ refers to the ordered topic multiset obtained after updating v_{jk} for the jth citation. Note that, by updating an inversion count, topic assignments may be altered with respect to more than one word tokens simultaneously. Therefore, the latter half of the right hand side of Eq. (6) reflects possible changes of topic assignments for multiple word tokens and is written as

$$p(\mathbf{x}_j|\mathbf{z}_j^{\text{new}}, \mathbf{x}_{-j}, \mathbf{z}_{-j}, \beta) = \prod_k \frac{\Gamma(n_k^{\neg j} + W\beta) \prod_w \Gamma(n_{kw}^{\text{new}} + \beta)}{\Gamma(n_k^{\text{new}} + W\beta) \prod_w \Gamma(n_{kw}^{\neg j} + \beta)} \ , \tag{7}$$

where n_{kw}^{new} means how many tokens of the word w are assigned to the the topic k after an update of an inversion count. In Eq. (7), $n_{kw}^{\neg j}$ means how many tokens of the word w are assigned to the topic k except the word tokens in the jth citation, n_k^{new} is defined to be $\sum_w n_{kw}^{\text{new}}$, and $n_k^{\neg j}$ is defined to be $\sum_w n_{kw}^{\neg j}$.

For each citation, we update topic draws as many times as the number of the word tokens in the citation. The probability that topic k is drawn as the ith topic draw for the jth citation can be written as follows:

$$p(t_{ji} = k|\ldots) \propto (n_{jk}^{\neg j} + \alpha) \cdot \prod_k \frac{\Gamma(n_k^{\neg j} + W\beta) \prod_w \Gamma(n_{kw}^{\text{new}} + \beta)}{\Gamma(n_k^{\text{new}} + W\beta) \prod_w \Gamma(n_{kw}^{\neg j} + \beta)} \ , \tag{8}$$

where $n_{jk}^{\neg j}$ means how many word tokens in the jth document are assigned to topic k except the ith topic draw. Note that more than one topic assignments can be altered even when we change only one topic draw. Therefore, Eq. (8) is more complicated than the equation used for LDA [7].

While Chen et al. [4] set the hyperparameters of the symmetric Dirichlet priors to constants, we use an empirical Bayes method proposed by Minka [9] and reestimate the Dirichlet hyperparameters once per each iteration. However, many trials in a preliminary experiment reveal that the reestimation works only for β. In constrast, for α, our preliminary experiment simply shows that a larger value leads to a better result. This observation is in contrast with [4] where

a small value is recommended to encourage a *sparse* topic distribution for each document. Consequently, we set $\alpha \to \infty$ and encourage *dense* topic distributions. This corresponds to the case where we replace Eq. (8) by

$$p(t_{ji} = k| \ldots) \propto \prod_k \frac{\Gamma(n_k^{\neg j} + W\beta) \prod_w \Gamma(n_{kw}^{new} + \beta)}{\Gamma(n_k^{new} + W\beta) \prod_w \Gamma(n_{kw}^{\neg j} + \beta)} \ . \tag{9}$$

That is, we drop the term related to α.

We can guess the reason why dense topic distributions are favorable for our problem as follows. In our case, each paragraph contains only one word token. Therefore, different paragraphs in the same document (i.e., in the same citation) do not show a meaningful divergence in word frequencies. However, by combining statistics of topic assignments *across* many citations, we can capture topical differences as differences in word frequencies. To combine statistics of topic assignments across many citations, we make topic distributions dense for every citation, because dense topic distributions can make many topics shared by different citations and thus can establish many "bonds" connecting the citations. Along such bonds, the statistics from many different citations can be summarized. Consequently, we can have meaningful differences in word frequencies. These differences may lead to an effective topic differentiation.

4 Experiment

4.1 Evaluation Settings

To obtain the datasets for our evaluation experiment, we used DBLP citation database[2] and MEDLINE/PUBMED database[3]. With respect to DBLP database, we used the XML file `dblp.xml` distributed on February 8, 2010 and composed three datasets D0, D20, and D50 as follows:

1. We collect the citations whose publication year ranges from 2000 to 2009. The number of citations amounts to 944,755. The number of different words is 685,799. Further, the number of word tokens is 17,408,876, which is larger by 35 times than the "Cities" corpus used in the experiment of [4].
2. We extract the five bibliographic elements: authors, article title, booktitle, journal, and year. However, we identify booktitle with journal, because not a few citations have completely the same content for both elements. As a result, we have the following *four* bibiliographic elements: authors, article title, booktitle, and year.
3. We fix the canonical order of the four bibliographic elements as follows: authors, article title, booktitle, and year. Note that the canonical order is not used in MCMC inference as an input. We do not need to specify anything other than the number of topics. We first sort the bibliographic elements in this order for all citations and compose three datasets D0, D20, and D50 as follows:

[2] `http://dblp.uni-trier.de/xml/`
[3] MEDLINE®/PUBMED®, a database of the U.S. National Library of Medicine.

Table 1. Dataset specifications.

DBLP datasets			MEDLINE datasets		
citations	word tokens	different words	citations	word tokens	different words
944,755	17,408,876	685,799	3,001,207	87,085,708	2,168,061

(a) By erasing the information about bibliographic elements, we make each citation into a sequence of untagged word tokens, i.e., into a raw text. We denote the set of these citations by D0. Since bibliographic elements are sorted in the same order for all citations, D0 provides an "ideal" problem to be solved.
(b) Before erasing the information about bibliographic elements, we randomly select 20% of the citations and randomly shuffle the order of bibliographic elements. We do not change the order of word tokens in each bibliographic element. For example, we do not change the ordering of the word tokens giving each paper title. After this random shuffling of the order of bibliographic elements, we erase the information about bibliographic elements. We denote this dataset by D20.
(c) We randomly shuffle the order of bibiliographic elements for the randomly selected 50% of the citations and erase the information about bibliographic elements. We denote the resulting dataset by D50. In D50, bibliographic elements are sorted in the canonical order for a far smaller subset of citations than in D20.

These three datasets, i.e., D0, D20, and D50, are called DBLP datasets in the discussions below.

From MEDLINE/PUBMED database, 100 files whose names range from `medline09n0400.xml` to `medline09n0499.xml` were used. In these 100 files, we could find 3,001,207 citations and 87,085,708 word tokens. We applied the procedure described above also to these files and composed three datasets M0, M20, and M50 in the same manner as D0, D20, and D50, respectively. We extracted the following *five* bibliographic elements: authors, publication year, article title, journal title, and pages. Further, this order was used as the canonical order. The three datasets, i.e., M0, M20, and M50, are called MEDLINE datasets.

For all six datasets, we applied no preprocessing like stemming, punctuation removal, and stop word elimination, because we wanted to compose datasets including citations similar to those obtained after an OCR processing of the reference part of printed papers or from the researchers' Web sites. However, for MEDLINE datasets, we eliminated the parentheses '[' and ']' appearing at the head and the tail of each article title, because they are artifacts which will not appear in any real data. Consequently, the number of different words is 2,168,061 for MEDLINE datasets. Table 1 summarizes dataset specifications.

We implemented MCMC inference for CBBK in `gcc` on Linux PC from scratch. The soundness of our implementation was checked by using the dataset in [4]. Our implementation gives intermediate sampling results per a fixed number of iterations. We combine all intermediate results to obtain a final answer as

follows: We assign each word token to the topic to which the word token is most frequently assigned among all intermediate results. The answer obtained in this manner leads to a better evaluation score than the sampling result available at the final iteration of MCMC.

Our problem is to obtain clusters of word tokens so that the word tokens in the same cluster refers to the same bibliographic element. Therefore, we evaluate the results by precision, recall, and F-score, which are standard evaluation measures for clustering. We adopt the definitions of these measures given by [4].

4.2 Preliminary Experiment

We first conducted a preliminary experiment on DBLP datasets and tested various settings for evaluation. Consequently, we obtained the following observations:

- While Chen et al. [4] set the number of MCMC iterations to 10,000, we needed at most 1,000 iterations to achieve a good enough result. This may be because our datasets, far larger than those used in [4], include redundancy.
- We parallelized MCMC with OpenMP library and obtained almost the same evaluation results as when we implemented no parallelization. We ran eight threads on Intel Core i7 920 CPU and made each thread process a non-overlapping subset of citations. Several types of statistics should be shared among the threads. Therefore, we made write operations to the variables holding such statistics mutually exclusive.
- We achieved better results for a smaller K, i.e., a smaller number of topics. However, when we made K equal to the true number of bibliographic elements, we could not obtain any good results. Therefore, we set K to the number greater by one than the true number of bibliographic elements. That is, we set $K = 5$ for DBLP datasets and set $K = 6$ for MEDLINE datasets.

Our experiment settings were fixed based on these preliminary observations.

4.3 Evaluation Results for DBLP Datasets

Figure 3 summarizes the results of the experiment conducted on DBLP datasets. Each solid bar represents an F-score averaged over 15 results of different MCMC inferences, and each error bar indicates plus and minus one standard deviation. Each of the 15 results, corresponding to a different MCMC inference, is obtained by combining 20 intermediate results. These 20 intermediate results are given by MCMC per 50 iterations from the 50th to the 1,000th iteration. The wall-clock time of 1,000 MCMC iterations was 11 hours on a PC equipped with Intel Core i7 920 and 12 Gbytes main memory.

The solid bars labaled as D0, D20, and D50 give F-scores for each of the three datasets, D0, D20, and D50, respectively. The dark gray solid bars labeled as "Opt. β" in the legend show F-scores obtained by optimizing β with Minka's method [9]. On the other hand, the light gray bars labeled as "$\beta = 0.05$" show F-scores obtained when β is fixed to 0.05. For both cases, we set α is set to

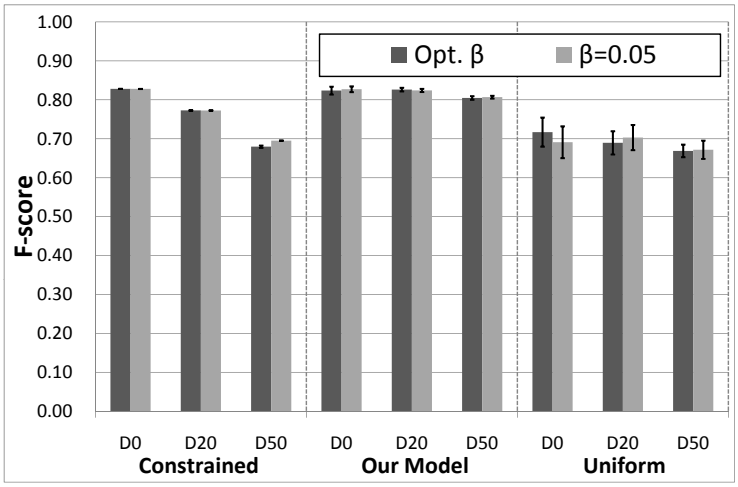

Fig. 3. Comparing F-scores obtained by applying CBBK and its two variants to DBLP datasets, i.e., the datasets D0, D20, and D50.

∞. We can observe that β set to 0.05 gave almost the same results with the optimized β for all cases.

The six solid bars in the leftmost area of Figure 3 represent F-scores achieved by the *constrained* model, one among the two variants of CBBK described in [4]. The constrained model forces the topic permutation distribution to give all its probability to one among $K!$ permutations and is implemented by fixing all inversion counts to zero. On the other hand, the six solid bars in the rightmost area represent F-scores achieved by the *uniform* model, another variant of CBBK. The uniform model gives the same probability to all $K!$ permutations and is implemented by setting $\rho_k = 0$ for all k. The six F-scores in the middle area are achieved by the CBBK with no restriction on permutation distributions. As Figure 3 shows, the unrestricted CBBK can provide better segmentation results than its two variants having a restriction on permutation distributions.

While the constrained model (see the leftmost area of Figure 3) leads to the results comparable with the unrestricted CBBK (see the middle area of Figure 3) only for D0 dataset, this may be because the bibliographic elements in all citations are sorted in the same order for D0. This situation poses no difficulty for the constrained model. However, when bibliographic elements appear in a non-canonical order for not a few citations, both of the constrained model and the uniform model are not effective.

Figure 3 also shows that the difficulty of our problem is not increased even when we randomly rearrange the order of bibliographic elements for 20% of the citations, as long as we use the CBBK with no restriction on permutation distributions. Further, even when we introduce a random rearrangement into 50% of the citations, F-score decreases only by two or three percent points. Therefore, we can say that GMM effectively infers the order of bibliographic

elements even when not a few citations include bibliographic elements in some non-canonical order.

We only consider the CBBK with no restriction on permutation distributions from now on and conduct no experiments related to the two variants of CBBK for MEDLINE datasets.

Finally, we add the following fact with respect to DBLP datasets: LDA only gives F-scores around 0.290 for all of D0, D20, and D50. Obviously, LDA gives almost the same F-scores for all of D0, D20, and D50, because LDA does not model any topic orderings and thus cannot make distinction between D0, D20, and D50. As The F-scores given by LDA are disastrously bad, we can say that contiguity constraint is mandatory for an effective segmentation of bibliographic elements with multi-topic modeling approaches.

4.4 Evaluation Results for MEDLINE Datasets

We next discuss the evaluation experiment conducted on MEDLINE datasets. The evaluation results are given in Table 2, which includes not only F-scores, but also precisions and recalls for revealing more details.

The number of word tokens of each MEDLINE dataset, M0, M20, and M50, is larger by 177 times than the "Cities" dataset in [4]. Therefore, we achieved an efficiency in computational time by reducing the number of MCMC iterations to 150. This number of iterations was determined based on an observation that topic assignments were not largely modified by MCMC after around this number of iterations. We think that the redundancy may be more common in MEDLINE datasets than in DBLP datasets. Therefore, we could reduce the number of MCMC iterations. The wall-clock time of 150 MCMC iterations was 17 hours on a PC equipped with Intel Core i7 920 and 12 GBytes main memory. We modified our implementation to output intermediate results per 10 iterations. Consequently, we obtained 15 intermediate results in total from the 10th to the 150th iteration for each execution of MCMC. Table 2 gives the precision, recall, and F-score averaged over 10 MCMC executions. Each averaged value is accompanied with the corresponding standard deviation.

With respect to β, we only show the results for the optimized β in Table 2, though, as in case of DBLP datasets, we could obtain almost the same results when $\beta = 0.05$. Instead, we show the results when we optimize α. The optimization is realized with Minka's method [9] as in case of β. The right half of Table 2 shows the results for the optimized α. Obviously, the optimized α gave segmentations of lower quality when compared with the results for $\alpha \to \infty$, which are given in the left half of Table 2, with respect to all three measures, i.e., precision, recall, and F-score. It can be said that we should make topic distributions *dense* by setting $\alpha \to \infty$. We only discuss this case from now on.

In Table 2, the F-score for M20 is less than that for M0, though the F-score for D20 is almost the same with that for D0 in Figure 3. Further, the difference of F-scores between M20 and M50 is larger than the difference between D20 and D50 presented in Figure 3. These results can be explained in the following manner. Recall that the number of bibliographic elements is five in MEDLINE datasets

Table 2. Comparing precisions, recalls, and F-scores obtained for MEDLINE datasets.

	Fixing α to ∞			Optimizing α		
	precision	recall	F-score	precision	recall	F-score
M0	0.870±0.001	0.828±0.001	0.849±0.001	0.469±0.001	0.798±0.001	0.591±0.001
M20	0.855±0.002	0.803±0.001	0.828±0.001	0.652±0.007	0.664±0.004	0.658±0.004
M50	0.791±0.002	0.726±0.001	0.757±0.002	0.718±0.004	0.618±0.003	0.664±0.002

and is four in DBLP datasets. Consequently, M20 and M50 includes 5!=120 variations of orderings of bibliographic elements. This number is larger than the number of ordering variations in D20 and D50, i.e., 4!= 24. Therefore, the segmentation of bibliographic elements for M20 and M50 becomes more difficult than that for D20 and D50. However, even when the number of bibliographic elements is large, our method can give a fairly good segmentation as long as the proportion of the noisy citations, i.e., the citations including bibliographic elements in a non-canonical order, is small.

5 Conclusions and Future Work

This paper provides a new method for segmentation of bibliographic elements by modifying CBBK, a probabilistic model proposed by Chen et al. [4]. We propose two strategies to solve the difficulties caused by regarding each word token as a paragraph and make CBBK applicable to our problem. Our two strategies, i.e., a special treatment of topic Dirichlet prior and a usage of large datasets, are aimed at intensively capturing topical relatedness *across* citations under a situation where we assign quite small units (i.e., word tokens) to topics. The evaluation experiment shows that our strategies realize an effective segmentation of bibliographic elements.

In a more realistic situation, OCR errors may be included in the citation data obtained from scanned articles. Therefore, it is an important future work to incorporate correction of OCR errors into our model as Takasu did for HMM [10]. With respect to the citations after this error correction, and also with respect to the citations downloaded from the Web pages, it is a possible direction to improve the quality of segmentation by controlling word probability distributions for each topic, as is proposed in [1], where we can use some external knowledge related to each bibliographic element.

We know that existing successful citation databases mainly adopt HMM-based approaches. However, such databases achieve their efficiency not only with an HMM modeling but also with other practical fine-tuning techniques. While our approach also requires many additional fine-tuning techniques, we think that our unsupervised approach can be an alternative to HMM as the basis for obtaining a new style of segmentation of bibliographic elements.

Acknowledgement

This work was supported in part by Japan Society for the Promotion of Science (JSPS) Grant-in-Aid for Young Scientists (B) 60413928 and also by Nagasaki University Strategy for Fostering Young Scientists with funding provided by Special Coordination Funds for Promoting Science and Technology of the Ministry of Education, Culture, Sports, Science and Technology (MEXT).

References

1. Andrzejewski, D., Zhu, X., and Craven, M.: Incorporating Domain Knowedge into Topic Modeling via Dirichlet Forest Priors. Proc. of ICML (2009)
2. Blei, D. M., Ng, A. Y., Jordan, M. I.: Latent Dirichlet Allocation. JMLR, Vol.3, pp. 993–1022 (2003)
3. Connan, J., Omlin, C. W.: Bibliography Extraction with Hidden Markov Models. Tech. Rep. US-CS-TR-00-6, Univ. of Stellenbosch (2000)
4. Chen, H., Branavan, S. R. K., Barzilay, R., Karger, D. R.: Global Models of Document Structure Using Latent Permutations. Proc. of ACL, pp. 371–379 (2009)
5. Eisenstein, J., Barzilay, R.: Bayesian Unsupervised Topic Segmentation. Proc. of EMNLP, pp. 334–343 (2008)
6. Fligner, M. A., Verducci, J. S.: Distance Based Ranking Models. J. R. Statist. Soc. B, Vol.48, No.3, pp. 359–369 (1986)
7. Griffiths, T. L., Steyvers, M.: Finding Scientific Topics. Proc. of Natl. Acad. Sci., Vol.101, Suppl.1, pp. 5228–5235 (2004)
8. Hetzner, E.: A Simple Method for Citation Metadata Extraction Using Hidden Markov Models. Proc. of JCDL, pp. 280–284 (2008)
9. Minka., T.: Estimating a Dirichlet Distribution. `http://research.microsoft.com/%7Eminka/papers/dirichlet/` (2000)
10. Takasu, A.: Bibliographic Attribute Extraction from Erroneous References Based on a Statistical Model. Proc. of JCDL, pp. 49–60 (2003)
11. Yin, P., Zhang, M., Deng, Z.-H., Yang, D.-Q.: Metadata Extraction from Bibliographies Using Bigram HMM. Proc. of ICADL, pp. 1–14 (2004)