Wikipedia-Based Efficient Sampling Approach for Topic Model

T.Zhao, C.Li and M.Li

School of Software, Tsinghua University, Beijing 100084, China
e-mail: zt882001@hotmail.com; cli@tsinghua.edu.cn; imyli1024@gmail.com

Abstract

In this paper, we propose a novel approach called Wikipedia-based Collapsed Gibbs sampling (Wikipedia-based CGS) to improve the efficiency of the collapsed Gibbs sampling(CGS), which has been widely used in latent Dirichlet Allocation (LDA) model. Conventional CGS method views each word in the documents as an equal status for the topic modeling. Moreover, sampling all the words in the documents always leads to high computational complexity. Considering this crucial drawback of LDA we propose the Wikipedia-based CGS approach that commits to extracting more meaningful topics and improving the efficiency of the sampling process in LDA by distinguishing different statuses of words in the documents for sampling topics with Wikipedia as the background knowledge. The experiments on real world datasets show that our Wikipedia-based approach for collapsed Gibbs sampling can significantly improve the efficiency and have a better perplexity compared to existing approaches.

Keywords

Gibbs sampling, Latent Dirichlet Allocation, Wikipedia, Topic Model

1. Introduction

The Latent Dirichlet Allocation (LDA) model, a general probabilistic framework for topic modeling, has been widely used for topic modeling and other related fields since it was first proposed by Blei et.al, 2003. The key idea of LDA model is to assume that a document is a mixture of topics, and words in the document have a distribution over these topics. Actually, these topics are represented as a multinomial distribution over the words. Based on these assumptions, the LDA model takes Bayesian framework as learning model by executing Expectation-Maximization algorithm from data iteratively. In 2004 Griffiths and Steyvers proposed a Markov-chain Monte Carlo method called Collapsed Gibbs Sampling (CGS), which has been widely used in LDA variants. From then on CGS becomes a straight-forward approach for LDA and rapidly converged to a well known ground-truth.

Based on LDA model, further ideas and techniques have been widely applied in LDA variants and other research fields. For example Author-Topic model (Steyvers *et al.*, 2004) uses the CGS to discover author's topics among documents; Joint Sentiment/Topic model (Lin and He, 2009) combines topic model with sentiment analysis to find topics with sentiment information by using CGS. All these works

need to employ LDA or its variants to generate topics from large amounts of documents automatically. However, since CGS views each word as the same status when sampling a topic for each word in the documents during each iteration, its performance seems far from satisfaction, especially on large textual corpora.

Therefore, to speed up the estimation procedure of LDA, we propose a novel sampling approach for topic modeling called Wikipedia-based Collapsed Gibbs sampling (Wikipedia-based CGS). We use the Wikipedia concept as background knowledge to distinguish words in the documents with three different statuses according to the meaningful case of the words. Then we assign different sampling times for these statuses. Experiments on real world datasets show that our approach presents a significant improvement for efficiency and a satisfied perplexity performance. From the experiment results, we also conclude that our approach focuses more sampling times on those meaningful words and less on other words to extract meaningful topics compared to other existed approaches.

2. Related work

Previous works on optimizing or parallelizing CGS have been explored in different implementations to improve the efficiency and overcome the scalability limitation. The first implementation of LDA is GibbsLDA. This standard LDA implementation has been widely used as a baseline model. Porteous et al. proposed FastLDA (Porteous *et al.*, 2008), considering that the posterior distribution is sparse for most words w and topics z. They exploited an upper bound of the posterior distribution and divided it into segments. Thus, FastLDA could sample the topic assignment for words without computing all $p(z_i \mid w)$, which means it improved the efficiency by executing less than K operations per iteration. Yao et al. proposed a SparseLDA (Yao *et al.,* 2009) to further improve the efficiency of CGS by dividing the full conditional probability mass into three parts and using an original approximate sampling scheme for document-topic count matrix and topic-word count matrix. Both FastLDA and SparseLDA require sampling for each word in the documents. Han Xiao *et al.* assumed that the same words in a document represented partly the same topics, so they considered to reduce sampling times for the same words in one document and proposed an Efficient Collapsed Gibbs sampling strategy (ECGS) (Xiao and Stibor, 2010). This paper described two optimization strategies for the ECGS algorithm. One is shortcut-ECGS, which assumes that the same words in one document have the same topic distribution. Though the shortcut-ECGS contributes to the efficiency improvement, the perplexity performance is unsatisfied. The other strategy is Dynamic-ECGS, which introduces a sampling-time vector for the same words in documents to decide the word's sampling times per iteration. In this strategy, the sampling-time of the type is a random variable and Dynamic-ECGS draws it from a multinomial distribution with the parameter vector Γ_{di} with a damping variable γ in iterations to gradually reduce the sampling-time for the same words over iterations. The vector is updated due to the unique drawn topics in each iteration. On the other hand, some parallelization works have also been proposed due to the high computational complexity of training LDA by using CGS. Newman et al. (Newman *et al.*, 2007) presented two synchronous methods, AD-LDA and HDLDA, to process distributed CGS algorithm. By straightforwardly mapping LDA to a

distributed processor setting, AD-LDA is easy to implement and can be viewed as an approximation to Gibbs-sampled LDA. While HDLDA is a model that uses a hierarchical Bayesian extension of LDA to account for distributed data directly. This model has a theoretical guarantee of convergence but is more complex to implement. In 2009, Wang et al. (Wang *et al.*, 2009) used the map-reduce framework and MPI to implement the AD-LDA, which is called PLDA.

However, all these works we mentioned above sample topics for all words, which takes considerably computational cost. Although the ECGS algorithm has reduced the sampling times of the same words, it does not distinguish status of the words in documents which could further reduce the sampling times per iteration.

3. Wikipedia-based CGS approach

Wikipedia-based CGS Algorithm

In the rest sections of this paper, we use token to represent the occurrence of a word and use type to represent the unique words, e.g. "the cat and the dog" has five tokens but four types. The important notations in this paper are demonstrated in Table 1.

Notation	Description
N_d	Number of types in document d
N_{di}	Number of the i_{th} type in document d
w_{di}	The i_{th} type in document d
z_{di}	The topic assignment for i_{th} type in document d
S_{di}^t	The sampling rate for w_{di} in iteration t
α, β, ω	Dirichlet priors

Table 1: Notations used in this paper

As mentioned above, Dynamic-ECGS algorithm is proposed to reduce the sampling times of repetitive words. However, not all types in documents should be reduced to a lower sampling rate. For instance, given a document with eight tokens “take” and three tokens “algorithms”, we naturally care more about the topics on “algorithms” rather than “take”, so the algorithm should take a relatively higher sampling rate for type "algorithm" than that of "take". Accordingly, we consider to employ higher sampling rate on those particularly meaningful words and lower sampling rate on others due to the theoretical reason that higher sampling rate for the type contributes to a more focused topic distribution. Here the sampling rate is formally defined. Sampling rate in CGS iteration t for type w is defined as the ratio of sampling times of the type to the number of occurrence of the type in a document as follows.

$$S_{di}^t = \frac{\square_{di}}{N_{di}}$$

where $S_{di}^t \in (0,1)$. $\square_{di}$ is defined as the sampling times in t iteration for the type w_{di}. Based on these, we propose a Wikipedia-based CGS algorithm. We consider to distinguish the statuses of types in a document into three statuses (*concept type*,

meaningful type and *general type*) according to its meaningful statuses based on Wikipedia background knowledge. The concept types are the words in vocabulary that can be matched by Wikipedia concepts and the concept types are constant. E.g., "algorithmic" is a concept type for that it can be matched by the Wikipedia concept "Algorithm". Assuming that some words that are not matched by Wikipedia are also partly meaningful, we divide the non-concept words into two statuses: meaningful type and general type. Actually, we view the non-concept word as a dynamic status between meaningful type and general type. Whether the non-concept type in a document is a meaningful type or a general type is determined by an indicator variable y drawn from a multinomial distribution π. The decision is made during the sampling process for the type of each document in each iteration, which means that a non-concept type in a document can be viewed as a meaningful type in this sampling process and then treated as a general type in the next sampling process. The distribution π is affected by the proportion of the concept types' number to the size of vocabulary and takes the proportion as its priors. Then we assign a higher sampling rate strategy for the concept type, a common sampling rate strategy for the meaningful type and a lower sampling rate strategy for the general type.

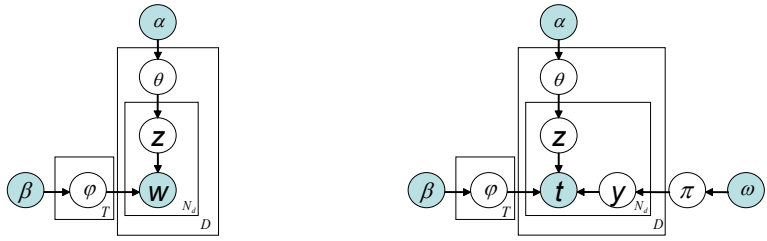
In order to evaluate the performance of our approach, we employ three sampling strategies as follows. 1). *Standard CGS strategy*: this sampling strategy does not reduce the sampling rate for the type in documents. 2). *Dynamic ECGS strategy*: this strategy has been illustrated in Section 2. We follow Dynamic-ECGS with different γ to be sampling strategies for our Wikipedia-based CGS strategies. A larger γ leads to relatively larger sampling rate in Dynamic-ECGS. 3). *Shortcut sampling strategy*: this strategy samples each type in a document only once in each iteration. The order of these strategies according to the decreasing order of sampling rate is shown as follows:

$$\text{Standard CGS} > \text{ECGS-}\gamma \sim 10 > \text{ECGS-}\gamma \sim 1 > \text{Shortcut CGS}$$

The reason we choose these strategies is that these strategies own different sampling rate which we need to assign a higher one for concept type and lower one for general type. We take experiments of choosing three different sampling strategies from the fours for the three statuses of types we defined to evaluate the performance of Wikipedia-based CGS approach.

3.1. Wikipedia-based CGS Framework

Wikipedia-based Collapsed Gibbs sampling algorithm assigns different sampling rate to different statuses of types as what we defined in Subsection 3.1. Concept types are constant during the sampling procedure, while meaningful types and general types are dynamic. We introduce a random variable y for each non-concept type to decide whether the type is meaningful or general. For those non-concept types in each document in each iteration during sampling process, if y =0, the type will be viewed as a meaningful one in this iteration; and if y=1, the type will be considered as a general one. Figure1 (b) shows the graphical model of Wikipedia-based CGS framework for LDA. The generative process is formulated as follows.



(a) LDA **(b) Wikipedia-based CGS framework**

Figure 1: the graphical model for LDA and our proposed method

1. Draw a multinomial distribution $\pi \sim Dir(\omega)$
2. For each topic, draw a multinomial distribution over words, $\varphi \sim Dir(\beta)$
3. For each document, draw a multinomial distribution over topics, $\theta \sim Dir(\alpha)$
4. For each type *t* in each document
   a) Draw a topic $z \sim Multi(\theta)$
   b) For each type *t′* matched by Wikipedia concepts
      i. Draw $t' \sim Multi(\varphi)$ with a higher sampling rate
   c) For each type *t″* non-matched by Wikipedia concepts
      i. Draw $y \sim Multi(\pi)$
      ii. Draw $t'' \sim Multi(\varphi)$ with a common sampling rate if $y = 0$
      iii. Draw $t'' \sim Multi(\varphi)$ with a lower sampling rate if $y = 1$

Due to the random variable *y*, the decision of meaningful type and general type is dynamic during the sampling process, and is affected by the number of concept words in vocabulary. Thus, as we employ a higher sampling rate for concept types and a lower sampling rate for general types, the total sampling times of Wikipedia-based CGS algorithm are reduced but focus more on both the concept types and meaningful types, which demonstrates a significant improvement for both the efficiency and the generalization performance.

4. Experiments

As is shown in Table 2, the experiments are conducted on three real world data sets: KOS blog entries (from dailykos.com), NIPS full papers (from books.nips.cc), and Enron emails (from www.cs.cmu.edu/~Enron).

	D	W	V
KOS	3,430	0.4×10^6	6,906
NIPS	1,500	1.9×10^6	12,419
Enron	39,861	6.4×10^6	28,102

Table 2: Details of three datasets used in experiments, D is the number of documents, W is the total number of words in the collection, and V is the size of vocabulary.

The experiments aim to demonstrate the speedup of Wikipedia-based CGS approach against the standard CGS and Dynamic ECGS algorithms and to show the better leverage of both improving efficiency and optimizing the topic extraction. We present the results of the experiments from three perspectives. We use perplexity curve to validate the convergence of Wikipedia-based CGS. Then, we measure the execution time of Wikipedia-based CGS algorithms by setting different number of topics under fixed iterations. Finally, we detect the proportion of sampling times for concept types to total sampling times over iterations to validate that Wikipedia-based CGS algorithm focuses more on those meaningful words. All the experiments are compared with standard CGS algorithm and Dynamic-ECGS algorithms. Wikipedia-based CGS algorithm requires us to know which words in vocabulary of the dataset can be matched by Wikipedia concepts. Here, we first briefly introduce the web service we use for matching Wikipedia concepts -- Wikipedia Miner.

4.1. Wikipedia Miner

Wikipedia Miner (Milne, 2009) is a toolkit for tapping the rich semantics encoded within Wikipedia. Here we use the search service to detect the concept types in vocabulary. We call the search service to find if there are any concepts that correspond to the word in vocabulary. E.g. given a query word "ai" for this service, the XML file returning from search service demonstrates "Artificial intelligence" as most related concept for this word. In KOS dataset, the proportion of the number of concept words to the size of vocabulary is 52%, which means that 52% words in KOS vocabulary can be matched by Wikipedia concepts. In NIPS dataset, the proportion is 54% and in Enron dataset the proportion is 24%. The proportions will affect the decision of a non-concept type whether to be set as a meaningful type or a general type during the sampling process.

4.2. Experimental setup

We implement the standard CGS, Dynamic-ECGS and Wikipedia-based CGS algorithms in JAVA. All the experiments are run 500 iterations. And we set the Dirichlet parameter α =50/K, β=0.02 proposed by Griffiths and Steyvers; Dirichlet parameter ω is set by the proportion of the number of concept words to the size of vocabulary. All models are training on 500 iterations. Due to the definition that Wikipedia-based CGS algorithm requires different sampling strategies for concept types, meaningful types and general types respectively, we experiment with different strategy combinations for Wikipedia-based CGS algorithm as is shown in Table 3.

	Concept type	Meaningful type	General type
Strategy 1	Standard CGS	ECGS γ~10	ECGS γ~1
Strategy 2	Standard CGS	ECGS γ~10	Shortcut CGS
Strategy 3	ECGS γ~10	ECGS γ~10	Shortcut CGS
Strategy 4	ECGS γ~10	ECGS γ~1	Shortcut CGS

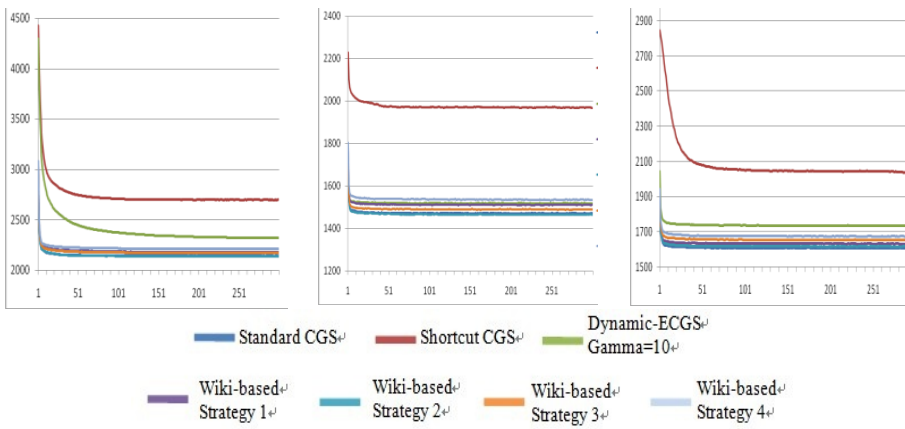
Table 3: Different strategy combinations for Wikipedia-based CGS algorithm.

4.3. Convergence Analysis

We use perplexity value to measure the convergence of Wikipedia-based CGS algorithm. Given test dataset D, the perplexity can be calculated as follows.

$$Perplexity(D) = \exp\{-\frac{\sum_{d=1}^{M} \log(p(w_d \mid D_{train}))}{\sum_{d=1}^{M} N_d}\}$$

We present the perplexity of the experiments for Wikipedia-based CGS, standard CGS and ECGS algorithms on KOS dataset, NIPS dataset and Enron email dataset. For each dataset in the experiments, 3/4 data is used for training, 1/4 data is used for testing. For NIPS dataset and KOS dataset, the number of topics is set as 40, and for Enron dataset, the experiments run with the number of topics as 100. Perplexity over iterations for Wikipedia-based CGS, standard CGS and ECGS algorithms is depicted in Figure 2. Due to the fact that a lower perplexity value indicates better generalization performance, we can observe that shortcut CGS algorithm converges to a suboptimal high perplexity value on all datasets, which makes it difficult to infer on new dataset. The reason that can be attributed to the assumption of the sampling strategy is that all repetitive tokens in a document represent the same topics. We can also observe that all the results of Wikipedia-based CGS approach show a better perplexity performance than that of Dynamic-ECGS with γ =10 on all these three datasets. The reason that we choose Dynamic-ECGS with γ =10 as a baseline is that we find out that a larger γ contributes to a better perplexity value but shows unsatisfied performance on efficiency. Even that Dynamic-ECGS with a larger γ can not show the efficiency improvement compared to standard CGS. From the experiments, we can see that our Wikipedia-based CGS approach converges as rapidly as standard CGS and has a significant improvement for efficiency. Moreover, Wikipedia-based CGS approach distinguishes the statuses of types in a document. Therefore, although the total sampling times of Wikipedia-based CGS strategies reduce, all the different settings of Wikipedia-based CGS strategies show optimized effects on perplexity performance.

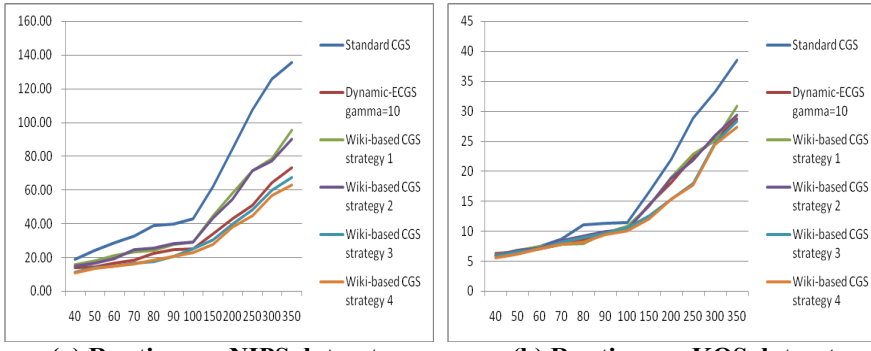


(a) Enron dataset **(b) KOS dataset** **(c) NIPS dataset**

Figure 2: perplexity value versus number of iterations on three datasets. Y-axis represents the perplexity value and X-axis represents the number of iterations

4.4. Speedup Results

The speedup results are evaluated by runtime of the different Wikipedia-based CGS strategies, Dynamic-ECGS and standard CGS algorithms in 500 iterations on both KOS and NIPS datasets. We experiment with different number of topics for these sampling strategies and depict the results in Figure 3. As for the poor performance of shortcut CGS in perplexity, we assume that shortcut CGS algorithm is just designed for speedup but not suitable for using on the real world datasets. Therefore, we do not make comparison with shortcut CGS algorithm. Runtime of standard CGS increases linearly with K, so we use standard CGS strategy as baseline to investigate the efficiency improvement of Wikipedia-based CGS approach. In Figure 3, we can see that with the increasing number of topics, Wikipedia-based CGS strategies show a remarkable efficiency improvement. The standard CGS strategy samples every token in a documents and Dynamic-ECGS strategy samples all types in a document with the same strategy. In contrast, Wikipedia-based CGS strategies separate the types in a document into three statuses. These strategies highlight the concept types and save sampling times from general types. This biased view for types makes the sampling process more efficient than other sampling strategies. From the result on KOS and NIPS datasets, we can also see that the improvement of efficiency on NIPS dataset is more outstanding than that on KOS dataset. The reason we analyze is that KOS dataset is collected from blog entries which have less redundancy repetitive tokens, while NIPS dataset is a collection from science papers including more repetitive words in document for clarifying the main idea of the paper and more official and scientific words that can be matched by Wikipedia concepts. Wikipedia-based CGS approach can use an equivalent sampling strategy as the standard CGS or Dynamic-ECGS strategy for concept types and employ other sampling strategies with lower sampling rate for those meaningful types and general types to improve the efficiency. From the experimental analysis, we can see that Wikipedia-based CGS strategies provide a subtle way to leverage the efficiency improvement and the goal of extracting meaningful and focused topics.



(a) Runtime on NIPS dataset **(b) Runtime on KOS dataset**

Figure 3: runtime results on KOS and NIPS datasets. Y-axis represents the runtime (minutes) of training model and X-axis represents the number of topics

4.5. Sampling Proportion of Concept Types

The main difference among Wikipedia-based CGS strategies and other sampling strategies is that the former ones emphasize the concept types and focuses more sampling times on them. In this subsection, we will calculate the sampling times for the concept types to validate our main idea of Wikipedia-based CGS strategies over iterations. We record the proportion of sampling times for concept types to total sampling times over 500 iterations with different algorithms by setting the number of topics K=40 on NIPS and KOS datasets and K=100 on Enron dataset. We depict the curves in Figure 4.

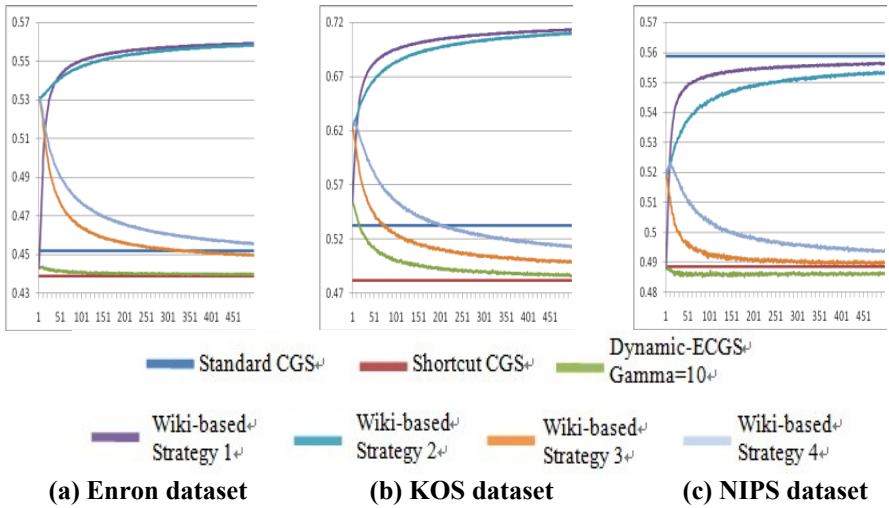


Figure 4: Sampling proportion of concept types over iterations on three datasets. Y-axis represents the proportion and X-axis represents the number of iterations

From the experiment results, we can observe that standard CGS algorithm has a fixed lower sampling proportion for concept types for that it does not reduce any sampling rate for all tokens in documents. Dynamic-ECGS algorithm has an even decreasing trend to a lower sampling proportion over iterations for concept types due to its non-biased sampling strategy. For Wikipedia-based strategy 3 and 4, we can see that due to the chosen strategies for the three statuses of types, which remarkably reduce the sampling rate of types, the sampling proportion of concept types decreases over iterations. However, the sampling proportion of concept types in Wikipedia-based strategy 3 and 4 are still higher than Dynamic-ECGS and Shortcut CGS algorithm, which means Wikipedia-based CGS algorithm focuses more on concept types than other algorithms when sampling. For Wikipedia-based strategy 1 and 2, we assign high sampling rate strategy for concept types, common sampling rate strategy for meaningful types and lower sampling rate strategy for general types, so we can see that the proportion of sampling times for concept types to total sampling times is increasing over iterations, which shows that Wikipedia-based CGS strategies focus more on those meaningful concept words than others, making it achieve the improvement for both the efficiency and accuracy.

5. Discussion and Future Works

Although we distinguish the words in documents into three statuses, we ignore the relatedness between the concept types and the document in this paper. Indeed, the relatedness between concept types in a document should also be considered to further improve the efficiency of sampling approach. We mark all these ideas to our future works.

6. Conclusions

In this paper, we present the Wikipedia-based Collapsed Gibbs sampling approach for improving the efficiency of LDA. This novel sampling strategy bias the types in a document to three statuses including concept type, meaningful type and general type according to their meaningful status by using Wikipedia concept as background knowledge. Instead of taking an equivalent sampling rate for all types in a document, Wikipedia-based CGS strategy incorporates three sampling algorithms with different sampling rate for the three statuses we define in order to sample more times on concept types and save sampling times from general types. We evaluate the experimental results from three aspects with different settings and validate that our approach obtains promising speedup and shows a better generalization performance.

7. Acknowledgements

This research work was supported by TSINGHUA National Laboratory for Information Science and Technology.

8. References

Blei, D.M., Ng, A.Y. and Jordan, M.I. (2003). "Latent dirichlet allocation", in *J. Mach. Learn. Res.*, 3:993–1022.

Griffiths, T. and Steyvers, M. (2004). "Finding scientific topics", in *Proceedings of the National Academy of Sciences,* 101(90001):5228–5235.

Lin, C. and He, Y. (2009). "Joint Sentiment/Topic Model for Sentiment Analysis", in *CIKM*.

Milne, D. (2009). "An open-source toolkit for mining Wikipedia", in *Proc. New Zealand Computer Science Research Student Conf*., *NZCSRSC*'09, Auckland, New Zealand.

Newman, D., Asuncion, A., Smyth, P. and Welling, M. (2007). "Distributed inference for latent dirichlet allocation", Volume 20, pages 1081–1088.

Porteous, I., Newman, D., Ihler, A., Asuncion, A., Smyth, P. and Welling, M. (2008). "Fast collapsed gibbs sampling for latent dirichlet allocation", in *SIGKDD*, pages 569–577. ACM.

Steyvers, M., Smyth, P., Rosen-Zvi, M. and Griffiths, T.L. (2004). "Probabilistic author-topic models for information discovery", in *SIGKDD*, pages 306–315.

Wang, Y., Bai, H., Stanton, M., Chen, W.Y. and Chang, E. (2009). "Plda: Parallel latent dirichlet allocation for large-scale applications", in *AAIM*, pages 301–314.

Xiao, H. and Stibor, T. (2010). "Efficient Collapsed Gibbs Sampling For Latent Dirichlet Allocation", in *JMLR*.

Yao, L., Mimno, D. and McCallum, A. (2009). "Efficient methods for topic model inference on streaming document collections", in *SIGKDD*, pages 937-946. ACM.