Improving Document Clustering for Short Texts by Long Documents via a Dirichlet Multinomial Allocation Model

Yingying Yan[1,2], Ruizhang Huang[1,2] (✉), Can Ma[1,2], Liyang Xu[1,2], Zhiyuan Ding[1,2], Rui Wang[1,2], Ting Huang[1,2], and Bowei Liu[1,2]

[1] College of Computer Science and Technology, Guizhou University, Guiyang, Guizhou, China
[2] Guizhou Provincial Key Laboratory of Public Big Data, Guizhou University, Guiyang, Guizhou, China
yyingy0921@gmail.com, rzhuang@gzu.edu.cn
{canma.love, lyxu90, subsontding}@gmail.com
{ruiwang1239, durant.huang, bwei.liu}@gmail.com

Abstract. Document clustering for short texts has received considerable interest. Traditional document clustering approaches are designed for long documents and perform poorly for short texts due to the their sparseness representation. To better understand short texts, we observe that words that appear in long documents can enrich short text context and improve the clustering performance for short texts. In this paper, we propose a novel model, namely DDMA*fs*, which 1) improves the clustering performance of short texts by sharing structural knowledge of long documents to short texts; 2) automatically identifies the number of clusters; 3) separates discriminative words from irrelevant words for long documents to obtain high quality structural knowledge. Our experiments indicate that the DDMA*fs* model performs well on the synthetic dataset and real datasets. Comparisons between the DDMA*fs* model and state-of-the-art short text clustering approaches show that the DDMA*fs* model is effective.

Keywords: Short text clustering; Dirichlet Multinomial Allocation; Gibbs sampling algorithm

1 Introduction

With the rapid development of the Internet, huge amount of short texts are generated. Short text clustering is of great interest for many applications. For example, document clustering for twitter messages is of substantial usage for analyzing the public opinions and interests. However, directly applying traditional document clustering models to short texts is with poor performance. The main reason is that the representation of short texts is highly sparse. Short texts are with a strict limit on the text length. For instance, twitter restricts the number of words in 140 characters for each message. As a result, discriminative terms

are in short and the number of common terms shared by related short texts is small.

Compared with short texts, long documents are with rich content and a large amount of discriminative terms. There are a number of document clustering approaches that achieve promising performance for discovering latent structure for long documents. Besides, it is practical to collect long documents related to short texts in real usage. For example, related long documents of twitter messages can be found from various document sources, such as news websites and blogs sites of news analysis. Therefore, to deal with the sparse representation problem of short texts, it would be useful if the high quality structural knowledge discovered from long documents can be shared to short texts to improve the understanding of short texts.

In practice, not every word in long documents is useful. Long document is normally represented by a number of discriminative words and a large amount of non-discriminative words. Only discriminative words are useful for grouping documents. The involvement of irrelevant noise words confuses the clustering process and leads to poor clustering solution for long documents which limit the effect of sharing the structural knowledge of long documents to improve the document clustering performance for short texts. This situation aggravates when the number of clusters are unknown.

The second challenge for short text clustering is to determine the number of clusters. Traditional short text clustering approaches consider the number of cluster as a predefined parameter. However, given large-scale short texts, users have to scan the whole document collection with the purpose of estimating the number of clusters. Apparently, it is time-consuming. In addition, inappropriate estimations of the number of clusters misdirect the short text clustering process and lead to bad clustering results.

In this paper, we propose a novel model, namely Dual Dirichlet Multinomial Allocation with feature selection (DDMA*fs*) to 1) improve the discovery of document structure for short texts by sharing structural knowledge of long documents; 2) relieve the effect of poor quality of long document representation by separating discriminative words from non-discriminative words automatically; 3) automatically identify the number of clusters of both long documents and short texts simultaneously. DDMA*fs* model is developed based on the Dirichlet Multinomial Allocation model(DMA) [3] which shows promising performance on document clustering for both long documents [6, 18] and short texts [17]. Long documents and short texts share the same set of latent clusters so that the structural knowledge can be transferred from long documents to short texts. Discriminative words are automatically separated from non-discriminative words for long documents. On the other hand, all terms in short texts are regarded as discriminative due to the sparse representation problem of short texts. Latent structure of short texts is further improved by only using the structural knowledge discovered from high quality discriminative words of long documents.

To determine the number of clusters, a Gibbs sampling algorithm is developed for the DDMA*fs* model. When a new data point arrives, it either rises

from existing clusters or starts a new cluster. The number of clusters for short texts K_S and long documents K_L are discovered automatically along the Gibbs sampling algorithm. Noted that K_S and K_L are not necessarily the same in our development. It is more practical for users to collect a large amount of long documents without the needed to guarantee that every long document should be directly related to short texts.

We have conducted extensive experiments on our proposed model by using both synthetic and realistic datasets. We compared our approach with state-of-the-art document clustering approaches. Experimental results show that our proposed approach is effective.

2 Related Work

Existing works mainly focused on utilizing external resources to enrich the contexts of short texts. [14, 19] aggregated the short texts into lengthy pseudo-documents for training topic model. Hong and Davison [4] presented several schemes to train a standard topic model with aggregated messages from Twitter. Hotho et al. [5] integrated Wordnet into the clustering process. In [10,11,16], Wikipedia is considered as a background base to enrich the knowledge of short texts.

There has been little work on sharing structural knowledge of long documents to short texts. In [9], Jin et al. proposed the Dual Latent Dirichlet Allocation (DLDA) model which enhances short text clustering by incorporating auxiliary long texts. However, DLDA is a probabilistic finite mixture model and the number of clusters is pre-defined. In reality, the number of clusters should be determined after the clustering process rather than got in advance.

Some methods have been introduced to find an estimation of the number of clusters K. The direct solution is to train the model with different value of K and pick the one with the highest likelihood on held-out dataset [12]. Another way is to assign a prior to K and then compute the posterior distribution of K to determine the most probable number of clusters [2]. In [17], Yin et.al inferred the number of clusters by the GSDMM model. DPMFS model [18] and DPMFP model [6] was proposed to estimate the document collection structure by utilizing the Dirichlet Process model. However, none of these approaches considers to automatically infer number of clusters for long documents and short texts simultaneously.

3 DDMA*fs* Model

Formally, we define the following notations:

- A word w is the basic unit of discrete data, defined to be an item from a vocabulary indexed by $\{1, \cdots, V\}$.
- A document can be represented as V-dimensional vector $x_d = \{x_{d1}, \cdots, x_{dV}\}$, where x_{dj} is the number of appearance of the j-th word in the document x_d.

– A dataset is a collection of D documents which are composed of two parts. The first part is long document set which is a collection of L long documents denoted by $D_L = \{x_1, x_2, \cdots, x_L\}$. The other is short text set which is a collection of S short texts denoted by $D_S = \{x_1, x_2, \cdots, x_S\}$.

The DDMA*fs* model is a generative probability model for long documents and short texts. Following the feature partition model mentioned in [6], a latent binary vector γ is used to partition words of long documents to two groups, in particular, the discriminative words and non-discriminative words. A mixture of components is used to generate short texts and discriminative words of long documents, where each component corresponds to a latent cluster characterized by a distribution over words. Non-discriminative words for long documents are generated from a background cluster. The generative process of the DDMA*fs* model is as follows:

1. Choose $\gamma_j \mid \omega \sim B(1, \omega)$, where $j = 1, 2, \cdots, V$.
2. Choose $\phi_k \mid \beta \sim \text{Dirichlet}(\beta_1, \beta_2, \cdots, \beta_V.)$, where $k = 1, 2, \cdots, K$.
3. Choose $\phi_0 \mid \lambda \sim \text{Dirichlet}(\lambda_1, \lambda_2, \cdots, \lambda_V)$.
4. Choose $\theta_S \mid \alpha \sim \text{Dirichlet}(\frac{\alpha}{K}, \frac{\alpha}{K}, \cdots, \frac{\alpha}{K})$;
   Choose $\theta_L \mid \alpha \sim \text{Dirichlet}(\frac{\alpha}{K}, \frac{\alpha}{K}, \cdots, \frac{\alpha}{K})$.
5. Choose $z_s \mid \theta_S \sim \text{Discrete}(\theta_{S1}, \theta_{S2}, \cdots, \theta_{SK})$, where $s = 1, 2, \cdots, S$;
   Choose $z_l \mid \theta_L \sim \text{Discrete}(\theta_{L1}, \theta_{L2}, \cdots, \theta_{LK})$, where $l = 1, 2, \cdots, L$.
6. Choose $x_s \mid \phi_{z_s} \sim \text{Multinomial}\ (\mid x_s \mid; \phi_{z_s})$, where $s = 1, 2, \cdots, S$;
   Choose $x_l \cdot \gamma \mid \phi_{z_l}, \gamma \sim \text{Multinomial}\ (\mid x_l \mid_{\gamma}; \phi_{z_l})$;
   Choose $x_l \cdot (1-\gamma) \mid \phi_0, \gamma \sim \text{Multinomial}\ (\mid x_l \mid_{1-\gamma}; \phi_0)$, where $l = 1, 2, \cdots, L$.

where ω is the parameter of the Bernoulli distribution, which represents the probability of each word expected to be discriminative. $|x_s|$ and $|x_l|$ are the total appearance of words in a short text x_s or a long document x_l, respectively. ϕ_k is the multinomial parameter representing the cluster k. K is the overall total number clusters for both long documents and short texts. L and S are the number of long documents and short texts respectively. The K-dimensional parameter θ_S and θ_L are the mixture weights of clusters for short texts and long documents, respectively; z_s and z_l indicate the latent cluster assigned to short text x_s and long document x_l, respectively. The graphical representation of DDMA*fs* model is shown in Fig.1.

The approximation of the probability density function of the dataset D_S and D_L given $\{z_1, z_2, \cdots, z_S\}$, $\{z_1, z_2, \cdots, z_L\}$, and γ can be represented as follows:

$$p(D_s|z_1, \cdots, z_S) \approx \prod_{s=1}^{S} \frac{|x_s|!}{\prod_{v=1}^{V} x_{sv}!} \cdot Q_{\beta} \tag{1}$$

$$p(D_L|z_1, \cdots, z_L, \gamma) \approx \prod_{l=1}^{L} \frac{|x_l|!}{\prod_{v=1}^{V} x_{lv}!} \cdot Q_{\beta,\lambda} \cdot Q_{\beta} \cdot Q_{\lambda} \tag{2}$$

$$Q_{\beta,\lambda} = \left(\frac{\Gamma(\sum_{v=1}^{V} \beta_v)}{\prod_{v=1}^{V} \Gamma(\beta_v)} \right)^{K} \cdot \frac{\Gamma(\sum_{v=1}^{V} \lambda_v)}{\prod_{v=1}^{V} \Gamma(\lambda_v)} \tag{3}$$

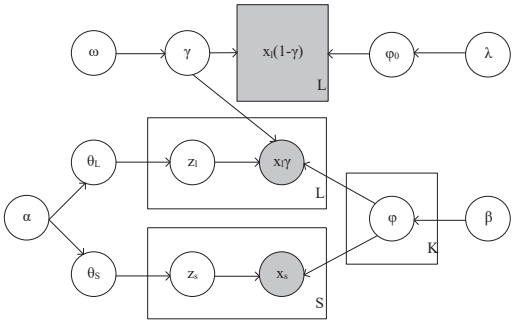

Fig. 1. Graphical representation of DDMA*fs* Model.

$$Q_\beta = \prod_{k=1}^{K} \frac{\prod_{v=1}^{V} \Gamma(\beta_v + \sum_{\{s:z_s=k\}} x_{sv} + \sum_{\{l:z_l=k\}} x_{lv}\gamma_v)}{\Gamma(\sum_{v=1}^{V} \beta_v + \sum_{v=1}^{V}(\sum_{\{s:z_s=k\}} x_{sv} + \sum_{\{l:z_l=k\}} x_{lv}\gamma_v))} \tag{4}$$

$$Q_\lambda = \frac{\prod_{v=1}^{V} \Gamma(\lambda_v + \sum_{l=1}^{L} x_{lv}(1-\gamma_v))}{\Gamma(\sum_{v=1}^{V} \lambda_v + \sum_{v=1}^{V} \sum_{l=1}^{L} x_{lv}(1-\gamma_v))} \tag{5}$$

4 Algorithm

In this section, a blocked Gibbs sampling algorithm is designed to infer the latent clusters and select discriminative words for long documents simultaneously.

For the DDMA*fs* model, the state of Markov chain is $\bar{U} = \{\gamma, \theta_L, \theta_S, \phi, zs, zl\}$, where $\gamma = \{\gamma_1, \cdots, \gamma_V\}$, $zl = \{z_1, \cdots, z_L\}$, $zs = \{z_1, \cdots, z_S\}$, $\phi = \{\phi_0, \cdots, \phi_K\}$. After initializing latent variables $\{\gamma, zl, zs\}$ and parameters $\{\alpha, \beta, \lambda, \omega\}$, the blocked Gibbs sampling inference procedure is as follows:

(1) Update the latent discriminative words indicator γ by repeating the following Metropolis step R times: a new candidate γ_{new} which adds or deletes a discriminative word is generated by randomly picking one of the V indices in γ_{old} and changing its value. The new candidate is accepted with the probability:

$$min\{1, \frac{p(\gamma_{new} \mid D_L, zl)}{p(\gamma_{old} \mid D_L, zl)}\} \tag{6}$$

where $p(\gamma \mid D_L, zl) \propto p(D_L \mid \gamma, zl) \cdot p(\gamma)$ and $p(D_L \mid \gamma, zl)$ is provided by Equation(4).

(2) Conditioned on the other latent variables, for $k = \{1, 2, \cdots, K\}$, if k is not in $\{z_1^*, z_2^*, \cdots, z_{K^*}^*\}$, draw ϕ_k from a dirichlet distribution with parameter β. Otherwise, update ϕ_k by sampling a value from a dirichlet distribution with parameter:

$$\left\{\beta_1 + \sum_{x_l:z_l=k} x_{l1}\gamma_1 + \sum_{x_s:z_s=k} x_{s1}, \cdots, \beta_V + \sum_{x_l:z_l=k} x_{lV}\gamma_V + \sum_{x_s:z_s=k} x_{sV}\right\} \tag{7}$$

(3) Update ϕ_0 by sampling a value from a dirichlet distribution with parameter:

$$\left\{\lambda_1 + \sum_{l=1}^{L} x_{l1}(1-\gamma_1), \cdots, \lambda_V + \sum_{l=1}^{L} x_{lV}(1-\gamma_V)\right\} \tag{8}$$

(4) Update θ_L by sampling a value from a dirichlet distribution with parameter:

$$\left\{\frac{\alpha}{K} + \sum_{l=1}^{L} I(z_l = 1), \cdots, \frac{\alpha}{K} + \sum_{l=1}^{L} I(z_l = K)\right\} \tag{9}$$

where $I(z_l = k)$ is an indicator function which equals to 1 if $z_l = k$.

(5) Update θ_S by sampling a value from a dirichlet distribution with parameter:

$$\left\{\frac{\alpha}{K} + \sum_{s=1}^{S} I(z_s = 1), \cdots, \frac{\alpha}{K} + \sum_{s=1}^{S} I(z_s = K)\right\} \tag{10}$$

where $I(z_s = k)$ is an indicator function which equals to 1 if $z_s = k$.

(6) Conditioned on the other latent variables, for $l = \{1, \cdots, L\}$, update z_l by sampling a value from a discrete distribution with parameter $\{p_{l1}, \cdots, p_{lK}\}$ where

$$\sum_{k=1}^{K} p_{lk} = 1 \text{ and } p_{lk} \propto \theta_{Lk} p(x_l \mid \phi_k, \phi_0, \gamma) \tag{11}$$

(7) Conditioned on the other latent variables, for $s = \{1, 2, \cdots, S\}$, update z_s by sampling a value from a discrete distribution with parameter $\{q_{s1}, \cdots, q_{sK}\}$ where

$$\sum_{k=1}^{K} q_{sk} = 1 \text{ and } q_{sk} \propto \theta_{Sk} p(x_s \mid \phi_k) \tag{12}$$

Note that the inference procedure focus on three parameters, in particular z, ϕ and θ which are closely related to the allocation of documents to clusters, the cluster representative parameters, and the cluster weight partitions. The parameter z, K, and γ are needed to be initialized. Other parameters are sampled during the inference process without the necessity of initialization. In our inference process, z is simply initialized by selecting a random cluster from $\{1, 2, \cdots, K\}$. The number of cluster K is initialized with a reasonably large value and can be automatically learned during the inference process. γ is initialized by randomly choosing one discriminative word in the dataset. All other words are initialized as non-discriminative with the value of γ equal to 0. The number of cluster estimated, denoted by K^*, is then determined by the size of $\{z_1^*, z_2^*, \cdots, z_{K^*}^*\}$ which is the set of non-empty clusters. K^* is much less than K with all the empty clusters eliminated. We can further divide K^* to K_S and K_L which are the number of clusters estimated for short texts and long texts. Note

that K_S and K_L are not necessarily the same. Through experimental study, we found that K_S and K_L are close to the true value of the number of clusters.

After the Markov chain has reached its stationary distribution, we collect H samples of $\{\gamma_1, \gamma_2, \cdots, \gamma_V\}$. we regard a word w_j is discriminative when the average value of γ_j of the last H samples is bigger than a threshold σ. (We set σ as 0.7 in our experiments).

5 Experiment

We study the performance of our proposed DDMA*fs* model by two sets of experiments. For the first set of experiments, synthetic datasets are used. For the second set of experiments, the DDMA*fs* model is evaluated on real document datasets. For both set of experiments, we used a standard document evaluation metric, in particular, Normalized Mutual Information (NMI) to evaluate the clustering performance [20].

5.1 Synthetic Dataset Experiments

Experimental Datasets We derived a synthetic dataset to evaluate the performance of our proposed DDMA*fs* model. The synthetic dataset consists of 3000 data points, in which 600 data points are used to represent short texts and 2400 data points are regarded as long documents. All data points were generated from 6000 features, in which 2000 features are regarded as discriminative features. The remaining 4000 features are regarded as non-discriminative features. For all data points, we derived one multinomial distribution to generate non-discriminative features for long documents. The parameter of the multinomial distribution, for non-discriminative features denoted as π_0, was generated randomly. Six multinomial distributions were used to represent latent clusters of discriminative features. Parameters of the six multinomial distributions, denoted as $\{\pi_1, \cdots, \pi_6\}$, were generated following the stick breaking approach of Dirichlet distribution [1]. In particular, one specific multinomial parameter $\pi_k = (u_1, \cdots, u_{2000})$ was generated as follows:

(1) For the first feature f_1, draw ι_1 from $Beta(\in_1, \sum_{j=2}^{V} \in_j)$ and then assign the probability of f_1 with ι_1, denoted as u_1.
(2) For feature f_i, where $2 \leq i < 2000$, draw ι_i from $Beta(\in_i, \sum_{j=i+1}^{V} \in_j)$ and then assign the probability of f_i, denoted as u_i, as follows:

$$u_i = \iota_i \pi_{j=0}^{i-1}(1 - \iota_j) \tag{13}$$

(3) u_{2000} is set with the remaining probability to ensure that $\sum_1^{2000} u_i = 1$.

In our experiment, we set $\in_i = 0.5$, where $i = 1, 2, \cdots, V$.

Each short text data point consists of 15 features. All features are regarded as discriminative generated from a multinomial mixture model with five components $\{\pi_1^S, \cdots, \pi_5^S\}$, where $\{\pi_1^S, \cdots, \pi_5^S\}$ is a subset of $\{\pi_1, \cdots, \pi_6\}$ and was selected randomly. In particular, the generation process of a short text data point x_i is as follows:

(1) Randomly select a cluster $\pi_k^S \in \{\pi_1^S, \cdots, \pi_5^S\}$
(2) Draw $x_i \sim Multinomial(\pi_k^S, 15)$

Each long document data point consists of 2000 features. The probability of a feature in long document data points to be discriminative is set to 0.6. Discriminative features were derived from a multinomial mixture model with 6 components $\{\pi_1^L, \cdots, \pi_6^L\}$. Non-discriminative features were generated from a multinomial distribution with parameter π_0. In particular, the generation process of a long document data point $x = (f_1, \cdots, f_i, \cdots, f_{2000})$ is as follows: For each feature f_i of x:

(1) Randomly select a probability $p_i \sim U[0, 1]$
(2) If $p_i > 0.6$, then
    a) randomly select a cluster $\pi_k^L \in \{\pi_1^L, \cdots, \pi_6^L\}$
    b) draw $f_i \sim Multinomial(\pi_k^L, 1)$
(3) Otherwise draw $f_i \sim Multinomial(\pi_0, 1)$

For our proposed DDMA*fs* model, we set $K = 30$, $\alpha = 0.1$, $\beta = 0.1$, $\omega = 0.01$, and $\lambda = 0.1$. The Metropolis step R was set to be 200. We ran our proposed DDMA*fs* model 10 times. The performance is computed by taking the average of these 10 experiments. Each experiment was conducted with 2000 iterations in which the first 500 as burn-in.

Experimental Performances We investigated the clustering performance for our proposed DDMA*fs* model by varying the number of long documents. Experimental results are depicted in Fig.2. From experimental results, it shows that our proposed DDMA*fs* model is effective for improving the clustering performance for short texts by transferring high quality structural knowledge discovered from long documents. When the number of long documents is equal to 0, the DDMA*fs* model is reduced to the ordinary DMA model. Clustering performances can be obviously improved when the number of long documents is increased. The improvement of the clustering performances is significant with a relatively small number of long documents. When the number of long documents is reasonably large, the DDMA*fs* model identifies almost perfect cluster structure.

We also investigated the performances of the DDMA*fs* model in one typical run by varying the number of iterations. The number of long documents involved is set to 2400. From Fig.3, it shows that the DDMA*fs* model reaches to a stable result within a few hundred iterations. Fig.4 and Fig.5 demonstrate the number of discriminative features, the number of clusters for short texts estimated, and the value of log likelihood with each iteration. The result shows that the number of discriminative features estimated is 2398 which is slightly larger than the real number. The feature selection process is faster to stabilize than the number of clusters estimated. The value of log likelihood increases obviously with decreasing number of clusters. DDMA*fs* model estimated 6 number of clusters for short texts and 14 clusters for long documents. The DDMA*fs* model is able to identify different numbers of clusters for short texts and long documents. Note that the

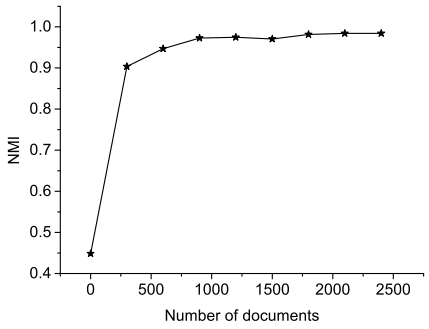

Fig. 2. Clustering performance of the DDMA*fs* model on the synthetic dataset with different number of long documents.

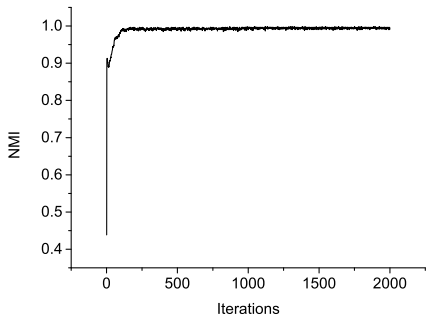

Fig. 3. Trace plot for NMI performance of the DDMA*fs* model for the synthetic dataset.

number of clusters estimated are slightly larger than the real ones. However, after removing those extremely small clusters, the number of clusters for both short texts and long documents are exactly the same with the real ones. The cluster assignments for data points are depicted in Fig.6 and Fig.7. Apparently, short data points are partitioned into 5 clusters and long data points are partitioned into 6 clusters which are the exact numbers of real clusters.

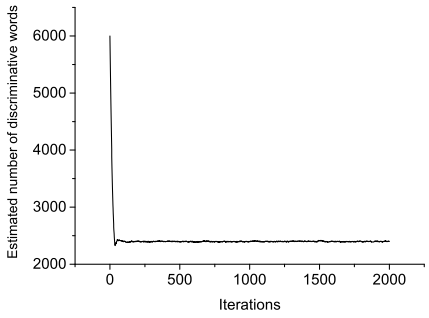

Fig. 4. Trace plot for the number of discriminative features for the synthetic datasets.

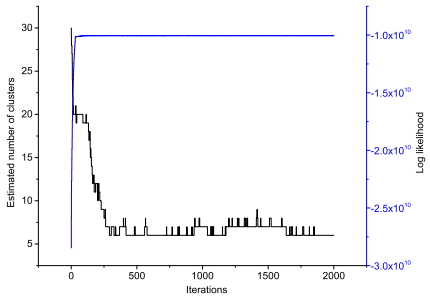

Fig. 5. Trace plot for the number of clusters for short text data points and log likelihood of the synthetic dataset.

To evaluate the generality on the performance of the DDMA*fs* model, we derived another synthetic dataset with 450 short texts and 1200 long documents by using similar strategy discussed before. Short texts were randomly selected from 4 classes. Long documents are organized in 3 classes. We obtained similar experimental performances except two observations: 1) the converging process is slower; 2) the NMI value, which is 0.90, is slightly worse than the previous synthetic experiments. The main reason is that only part of short texts is improved by long documents because the number of classes of short texts is smaller than long documents.

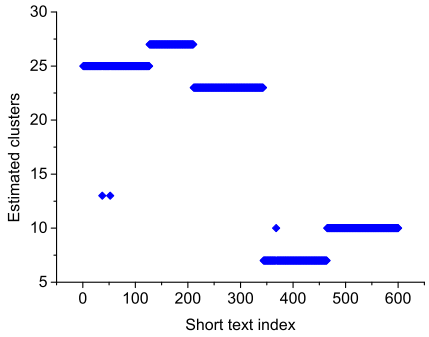

Fig. 6. Estimated cluster labels of the short text data points.

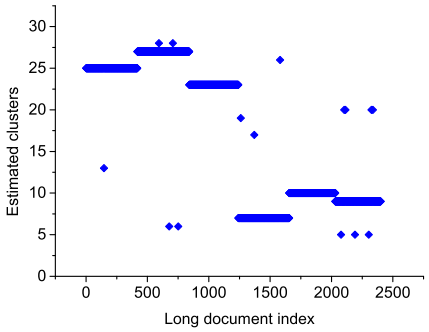

Fig. 7. Estimated cluster labels of the long document data points.

Discussions In this section, we investigated the sensitivity of the choices of hyper parameters α, β, λ, and ω for the DDMA*fs* model. Experiments were conducted on various values of these parameters. There are some other parameters, in particular, the initial number of clusters K and the Metropolis step parameter R. We discuss the setting of these parameters in detail in the following part of the section.

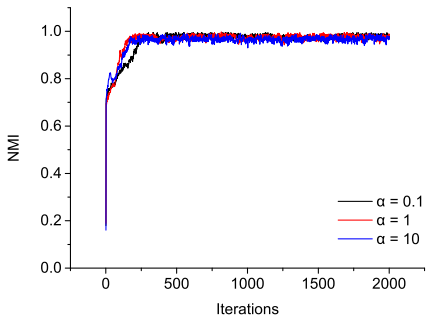

Fig. 8. The NMI result for short text data points when α gets different values.

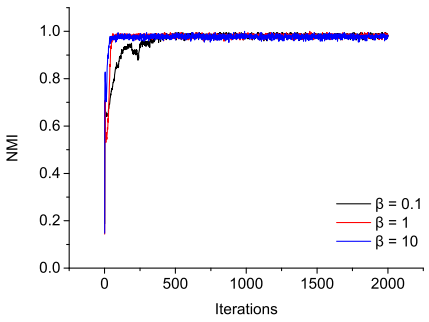

Fig. 9. The NMI result for short text data points when β gets different values.

Choice of α, β, λ and ω: We investigated the sensitivity of the choice of hyper parameters α, β, λ, and ω with the performance of the DDMA*fs* model. α, β, and λ were set to 0.1, 1, and 10 which corresponds to a small, moderate, and large prior values. We also experimented with different values of ω where ω was set to be a small value 0.01, a moderate value 0.1, and a large value 1. For the different values of α, β, λ, and ω, we set other values of parameters with the same setting discussed before. Fig.8 to Fig.11 show the clustering performance for short texts by varying the number of iterations. The DDMA*fs* model achieved almost perfect clustering structure in all these experiments. This indicates that the DDMA*fs* model is robust to the choice of hyper parameters.

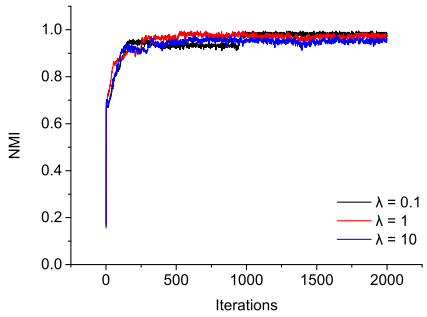

Fig. 10. The NMI result for short text data points when λ gets different values.

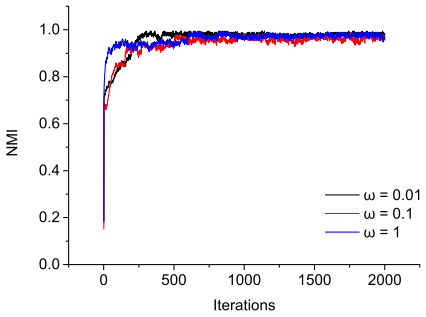

Fig. 11. The NMI result for short text data points when ω gets different values.

Choice of K and R: Theoretically, we should choose K to be the number of data points. In the process of experiment, we discovered that it is time-consuming. So we chose a relatively small K follow the advice of [7]. The number of Metropolis step R was set to be 200 in our algorithm because we found that lager value on R had little improvement on the clustering quality.

5.2 Real Dataset Experiments

Experimental Datasets Two real-word text corpora were used to generate experimental datasets for conducting experiments. The first corpus is the AMiner-Paper collection [13]. Three different research areas were chosen, in particular, the "graphical image", the "computer network", and the "database", to form a subset of the AMiner-Paper collection. The first experimental dataset, namely, AMpaperSet, was then derived from the subset by extracting 600 paper titles for short texts and another 450 paper abstracts for long documents. The second dataset is called the TweetSet dataset. We crawled 79413 tweets from 3 hot topics on twitter from the hashtag "JeSuisParis", "RefugeesWelcome" and "PlutoFlyby". In these tweets, there are 6399 tweets containing URLs and accessible URLs are 5577, we therefore crawled the content of the accessible URLs to form the set of long documents. The TweetSet dataset was then derived by randomly selecting 2400 documents from the crawled long documents and 600 documents from tweets.

We pre-processed all the datasets by stop-word removal. The summary of these two text document datasets is shown in Table 1.

Table 1. Summary Description of Datasets.

Datasets	L	S	V	K
AMpaperSet	450	600	3586	3
TweetSet	2400	600	33462	3

(L: Long Text Sets, S: Short Text Sets, V: Vocabulary size, K: Number of clusters.)

Experimental Setup and Experimental Performances For all set of experiments, we use the same parameter settings of α, β, ω, λ, and R of the synthetic dataset to real dataset experiment. For the number of initial clusters K, we set K to 30 for both AMpaperSet and TweetSet dataset.

For comparative study, we compare our models with four other approaches. We investigated the standard K-Means document clustering model taking the bag-of-word assumption as the first approach, labeled as KMEANS [8]. The K-MEANS approach is used as the benchmark. The number of cluster is required as the input parameter in this model. The second and third approaches are state-of-art document clustering approaches for short texts, in particular, GSDMM [17] and STCC [15]. The GSDMM approach is designed based on the dirichlet multinomial mixture model. A collapsed gibbs sampling algorithm is employed to infer the number of clusters automatically. The STCC approach is designed with the help of convolutional neural networks, and takes the number of clusters as a pre-defined parameter. The fourth approach, labeled as DLDA model, is the most recent short text clustering model which transfers structural knowledge learned from auxiliary long documents to short texts [9]. Although DLDA model utilizes the long documents to cope with sparse problem, it can't infer the number of clusters automatically. The KMEANS, GSDMM and STCC approaches are not specially designed for studying how long documents improve the clustering performance of short texts. Therefore, the type of data points can't be identified in these two models. We evaluated the experimental performances with and without long document data points for the KMEANS and GSDMM approaches respectively. For experiments with long documents, we merged short text and long document data points to form a single dataset and evaluated the performance on short texts only. We studied the performance of KMEANS, DLDA, and STCC when right or wrong number of clusters are given. Each comparative experiment was run 10 times. The performance is computed by taking the average of the NMI results on short text data points of these 10 experiments. The clustering performance of long documents is not the focus of our paper.

Table 2 depicts document clustering performances acquired by the DDMA*fs*, KMEANS, GSDMM, STCC, and DLDA models on the AMpaperSet and TweetSet dataset. From the experimental results, our proposed DDMA*fs* model apparently performs better compared with all other models. Therefore, the DDMA*fs* model is effective for discovering the latent document structure of short texts. Note that the DDMA*fs* model reduces to the ordinary DMA model, which shares the same document generation process with the GSDMM model, when no long documents are available. Compared the experimental performances of the DDMA*fs* and the GSDMM(s), it is obvious that long documents are able to help the clustering performance of short texts. The clustering performance can be greatly improved with the help of long documents. In all experiments with long documents, the DDMA*fs* model outperforms all other models. There are two main reasons. Firstly, long document data points are with a great number of non-discriminative features which deduce the quality of structural knowledge shared to short texts for all other models. Secondly, the KMEANS and GSD-

Table 2. Cluster performance on short texts on the AMpaperSet and TweetSet datasets.

	AMpaperSet	TweetSet
DDMA*fs*	0.465	0.557
KMEANS(s)	0.136	0.126
KMEANS(K=2)	0.341	0.091
KMEANS(K=3)	0.428	0.075
KMEANS(K=10)	0.260	0.293
GSDMM(s)	0.376	0.432
GSDMM	0.430	0.539
STCC(s)	0.16	0.18
DLDA(K=2)	0.287	0.277
DLDA(K=3)	0.342	0.311
DLDA(K=10)	0.187	0.232

(KMEANS(s)(GSDMM(s), or STCC(s)) indicates the clustering performance of KMEANS(GSDMM, or STCC) approach without the aid of long documents.)

MM models are not able to identify long document and short text data points which results in losing the information on cluster partition of short texts and long documents.

Table 3. Number of clusters on short texts estimated on the AMpaperSet and TweetSet datasets.

	DDMA*fs*	GSDMM(s)	GSDMM
AMpaperSet	20	21	27
TweetSet	17	25	23

Table 3 shows the estimated number of clusters on two real dataset. Among the five methods, KMEANS, STCC and DLDA are given the true value of K. All estimation on the number of clusters are larger than the true one due to the reason of outlier documents. The DDMA*fs* model obtains a relatively accurate estimation compared with GSDMM on two real datasets.

Table 4. Top 8 words of three typical larger clusters discovered by DDMA*fs* model on the AMpaperSet dataset.

Cluster	Top words
1	transact process optimization cost distribute system file memory
2	sensor wireless traffic node rout protocol rate channel
3	query relation model language object operation update semantic

Table 4 shows top 8 words of three typical larger clusters on the AMpaperSet dataset discovered by our proposed DDMA*fs* model. DDMA*fs* model

captures meaningful words associated with three clusters, in particular, "graphical image", "computer network", and "database". For some general clusters, the DDMA*fs* model subdivides the cluster to more specific sub-clusters as the number of clusters is unknown. As a result, more clusters are discovered than the real ones. As shown in Table 4, cluster 1 and 3 are two related clusters under the general cluster of "database".

We explores the impact of long documents to short texts on the TweetSet dataset. Similar performance was obtained with synthetic dataset as shown in Fig.2. We can find that the clustering performance is improved with increase of the number of long documents. It shows that the clustering effect of short texts will level up with the help of long documents in the real application.

6 Conclusion and Future Work

In this paper, we propose a DDMA*fs* model for the problem of short text clustering. Structural knowledge of long documents are shared to short texts so that the clustering performance of short texts can be greatly improved. A blocked Gibbs sampling technique is proposed to infer the cluster structure of short text set as well as the latent discriminative word subset of long document set. Our experiment shows that our approach achieves good performance with a reasonable set of long documents. The comparisons between DDMA*fs* and existing state-of-the-art models indicate that our approach is effective.

An interesting direction for future research is to study how to enhance the clustering of short texts via utilizing multi-source dataset rather than only long documents. Besides, we also concern to involve a small number of supervised information on long documents for improving the performance of short text clustering.

Acknowledgments. This work is supported by Nation Science Foundation of China (NO.61462011, NO.61202089), Introduce Talents Science Projects of Guizhou University (NO.2016050), Major Applied Basic Research Program of Guizhou Province (Grant No.JZ20142001) and Graduate Innovated Foundation of Guizhou University (NO.2016051).

References

1. Bela, A., Frigyik, A., Gupta, M.: Introduction to the dirichlet distribution and related processes. Department of Electrical Engineering, University of Washington (2010)
2. Cheeseman, P., Kelly, J., Self, M., Stutz, J., Taylor, W., Freeman, D.: Autoclass: A bayesian classification system. In: Readings in knowledge acquisition and learning. pp. 431–441. Morgan Kaufmann Publishers Inc. (1993)
3. Green, P.J., Richardson, S.: Modelling heterogeneity with and without the dirichlet process. Scandinavian journal of statistics 28(2), 355–375 (2001)

4. Hong, L., Davison, B.D.: Empirical study of topic modeling in twitter. In: Proceedings of the first workshop on social media analytics. pp. 80–88. ACM (2010)
5. Hotho, A., Staab, S., Stumme, G.: Wordnet improves text document clustering. In: Proceedings of the SIGIR 2003 Semantic Web Workshop. pp. 541–544 (2003)
6. Huang, R., Yu, G., Wang, Z., Zhang, J., Shi, L.: Dirichlet process mixture model for document clustering with feature partition. IEEE Transactions on Knowledge and Data Engineering 25(8), 1748–1759 (2013)
7. Ishwaran, H., James, L.F.: Gibbs sampling methods for stick-breaking priors. Journal of the American Statistical Association 96(453), 161–173 (2001)
8. Jain, A.K.: Data clustering: 50 years beyond k-means. Pattern recognition letters 31(8), 651–666 (2010)
9. Jin, O., Liu, N.N., Zhao, K., Yu, Y., Yang, Q.: Transferring topical knowledge from auxiliary long texts for short text clustering. In: Proceedings of the 20th ACM international conference on Information and knowledge management. pp. 775–784. ACM (2011)
10. Phan, X.H., Nguyen, C.T., Le, D.T., Nguyen, L.M., Horiguchi, S., Ha, Q.T.: A hidden topic-based framework toward building applications with short web documents. IEEE Transactions on Knowledge and Data Engineering 23(7), 961–976 (2011)
11. Phan, X.H., Nguyen, L.M., Horiguchi, S.: Learning to classify short and sparse text & web with hidden topics from large-scale data collections. In: Proceedings of the 17th international conference on World Wide Web. pp. 91–100. ACM (2008)
12. Smyth, P.: Model selection for probabilistic clustering using cross-validated likelihood. Statistics and computing 10(1), 63–72 (2000)
13. Tang, J., Zhang, J., Yao, L., Li, J., Zhang, L., Su, Z.: Arnetminer: extraction and mining of academic social networks. In: Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 990–998. ACM (2008)
14. Weng, J., Lim, E.P., Jiang, J., He, Q.: Twitterrank: finding topic-sensitive influential twitterers. In: Proceedings of the third ACM international conference on Web search and data mining. pp. 261–270. ACM (2010)
15. Xu, J., Wang, P., Tian, G., Xu, B., Zhao, J., Wang, F., Hao, H.: Short text clustering via convolutional neural networks. In: Proceedings of NAACL-HLT. pp. 62–69 (2015)
16. Yang, C.L., Benjamasutin, N., Chen-Burger, Y.H.: Mining hidden concepts: Using short text clustering and wikipedia knowledge. In: Advanced Information Networking and Applications Workshops (WAINA), 2014 28th International Conference on. pp. 675–680. IEEE (2014)
17. Yin, J., Wang, J.: A dirichlet multinomial mixture model-based approach for short text clustering. In: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 233–242. ACM (2014)
18. Yu, G., Huang, R., Wang, Z.: Document clustering via dirichlet process mixture model with feature selection. In: Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 763–772. ACM (2010)
19. Zhao, W.X., Jiang, J., Weng, J., He, J., Lim, E.P., Yan, H., Li, X.: Comparing twitter and traditional media using topic models. In: European Conference on Information Retrieval. pp. 338–349. Springer (2011)
20. Zhong, S.: Semi-supervised model-based document clustering: A comparative study. Machine learning 65(1), 3–29 (2006)