Sequential Embedding Induced Text Clustering, a Non-parametric Bayesian Approach

Tiehang Duan[1(✉)], Qi Lou[2], Sargur N. Srihari[1], and Xiaohui Xie[2]

[1] Department of Computer Science and Engineering, State University of New York at Buffalo, Buffalo, NY 14260, USA
tiehangd@buffalo.edu, srihari@cedar.buffalo.edu

[2] Department of Computer Science, University of California, Irvine, Irvine, CA 92617, USA
{qlou,xhx}@ics.uci.edu

Abstract. Current state-of-the-art nonparametric Bayesian text clustering methods model documents through multinomial distribution on bags of words. Although these methods can effectively utilize the word burstiness representation of documents and achieve decent performance, they do not explore the sequential information of text and relationships among synonyms. In this paper, the documents are modeled as the joint of bags of words, sequential features and word embeddings. We proposed **S**equential Embedding **i**nduced **D**irichlet **P**rocess **M**ixture **M**odel (SiDPMM) to effectively exploit this joint document representation in text clustering. The sequential features are extracted by the encoder-decoder component. Word embeddings produced by the continuous-bag-of-words (CBOW) model are introduced to handle synonyms. Experimental results demonstrate the benefits of our model in two major aspects: (1) improved performance across multiple diverse text datasets in terms of the normalized mutual information (NMI); (2) more accurate inference of ground truth cluster numbers with regularization effect on tiny outlier clusters.

1 Introduction

The goal of text clustering is to group documents based on the content and topics. It has wide applications in news classification and summarization, document organization, trend analysis and content recommendation on social websites [13, 17]. While text clustering shares the challenges of general clustering problems including high dimensionality of data, scalability to large datasets and prior estimation of cluster number [1], it also bears its own uniqueness: (1) text data is inherently sequential and the order of words matters in the interpretation of document meaning. For example, the sentence “people eating vegetables” has a totally different meaning from the sentence “vegetables eating people”, although two sentences share the same bag-of-words representation. (2) Many English

Q. Yang et al. (Eds.): PAKDD 2019, LNAI 11441, pp. 68–80, 2019.
https://doi.org/10.1007/978-3-030-16142-2_6

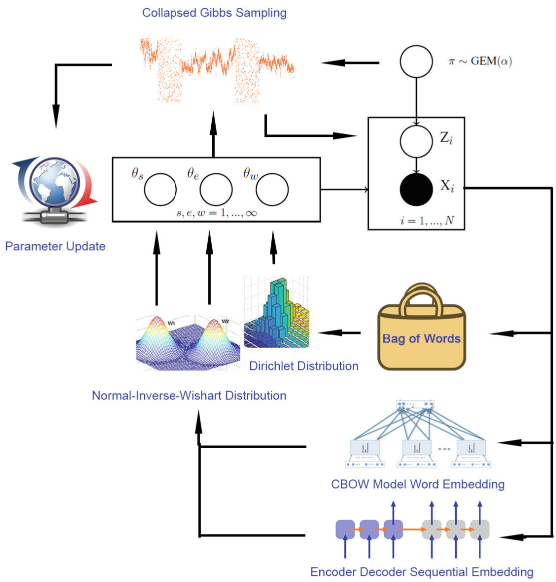


Fig. 1. Illustration of the proposed sequential embedding induced Dirichlet process mixture model (SiDPMM).

words have synonyms. Clustering methods taking synonyms into account will possibly be more effective to identify documents with similar meanings.

Pioneering works in text clustering have been done to address the general challenges of clustering. Among them nonparametric Bayesian text clustering utilizes Dirichlet process to model the mixture distribution of text clusters and eliminate the need of pre-specifying the number of clusters. Current methods use bag of words for document modeling. In this work, as shown in Fig. 1, the Bayesian nonparametric model is extended to utilize knowledge extracted from an encoder-decoder model and word2vec embedding, and documents are jointly modeled by bag of words, sequential features and word embeddings. We derive an efficient collapsed Gibbs sampling algorithm for performing inference under the new model.

Our Contributions. (1) The proposed SiDPMM is able to incorporate rich feature representations. To the best of our knowledge, this is the first work that utilizes sequential features in nonparametric Bayesian text clustering. The features are extracted through an encoder-decoder model. It also takes synonyms into account by including CBOW word embeddings as text features, considering that documents formed with synonym words are more likely to be clustered together. (2) We derive a collapsed Gibbs sampling algorithm for the proposed model, which enables efficient inference. (3) Experimental results show that our model outperforms current state-of-the-art methods across multiple datasets, and have a more accurate inference on the number of clusters due to its desirable regularization effect on tiny outlier clusters.

2 Related Work

Traditional clustering algorithms such as K-means, Hierarchical Clustering, Singular Value Decomposition, Affinity Propagation have been successfully applied in the field of text clustering (see [23] for a comparison of these methods on short text clustering). Algorithms utilizing spectral graph analysis [4], sparse matrix factorization [25], probabilistic models [24] were proposed for performance improvement. As text is usually represented as a huge sparse vector, previous works have shown that feature selection [7,14] and dimension reduction [9] are also crucial.

Most classic methods require access to prior knowledge about the number of clusters, which is not always available in many real-world scenarios. Dirichlet Process Mixture Model (DPMM) has achieved state-of-the-art performance in text clustering with its capability to model arbitrary number of clusters [27,29]; number of clusters is automatically selected in the process of posterior inference. Variational inference [2] and Gibbs sampling [6,21] can be applied to infer cluster assignments in these models.

A closely related field of text clustering is topic modeling. Instead of clustering the documents, topic modeling aims to discover latent topics in document collections [3]. Recent works showed performance of topic modeling can be significantly improved by integrating word embeddings in the model [16,26,30].

The encoder-decoder model was recently introduced in natural language processing and computer vision to model sequential data such as phrases [10,11] and videos [12]. It has shown great performance on a number of tasks including machine translation [5], question answering [22] and video description [12]. Its strength of extracting sequential features is revealed in these applications.

3 Description of SiDPMM

Our text clustering model is based on the Dirichlet process mixture model (DPMM), the limit form of the Dirichlet mixture model (DMM). When DPMM is applied to clustering, the size of clusters are characterized by the stick-breaking process, and prior of cluster assignment for each sample is characterized by the Chinese restaurant process. The Dirichlet process can model arbitrary number of clusters which is typically inferred via collapsed Gibbs sampling or variational inference. We refer readers to [2,21] for more details about DPMM.

We tailor DPMM to our task by learning clusters with multiple distinct information sources for documents, i.e., bag-of-words representations, word embeddings and sequential embeddings, which requires specifically designed likelihood, priors, and inference mechanism.

Table 1. Notations

Notation	Meaning
d_i	the i-th document
$\boldsymbol{d}_{k,\neg i}$	documents belonging to cluster k excluding d_i
K	total number of clusters
c_i	cluster assignment of d_i
$\boldsymbol{c}_{k,\neg i}$	cluster assignments of cluster k excluding document i
θ_k	parameters of cluster k
r_k	number of documents in cluster k
u_i	number of words in document i
u_i^t	occurrence of word t in document i
$u_{k,\neg i}$	number of words in cluster k excluding d_i
$u_{k,\neg i}^t$	occurrence of word t in cluster k excluding d_i
w_i	the set of bag of words in d_i
s_i	sequential information embedding of d_i
e_i	word embedding of d_i
V	vocabulary size
Θ_s	set of hyper-parameters $\{\mu_s, \lambda_s, \nu_s, \Sigma_s\}$
Θ_e	set of hyper-parameters $\{\mu_e, \lambda_e, \nu_e, \Sigma_e\}$
α	parameter of Chinese restaurant process
β	hyper-parameter for multinomial modeling of bag of words
ϵ	dimensionality of sequential embedding vector
δ_k	parameter of multinomial distribution for the k-th cluster

To start with, we first introduce the likelihood function $F(d_i|\theta_k)$ over documents:

$$F(d_i|\theta_k) = \text{Mult}(w_i|\delta_k)\mathcal{N}(e_i|\mu_e^k, \Sigma_e^k)\mathcal{N}(s_i|\mu_s^k, \Sigma_s^k) \tag{1}$$

where $\theta_k = (\mu_e^k, \Sigma_e^k, \mu_s^k, \Sigma_s^k, \delta_k)$, with $\delta_k = (\delta_k^1, \ldots, \delta_k^V)$ and $\sum_{j=1}^V \delta_k^j = 1$. e_i is the word embedding and s_i is the encoded sequential vector. The multinomial component $\text{Mult}(w_i|\delta_k)$ captures the distribution of bag of words; the Normal components $\mathcal{N}(e_i|\mu_e^k, \Sigma_e^k)$, $\mathcal{N}(s_i|\mu_s^k, \Sigma_s^k)$ measure similarities of word and sequential embeddings. This model is general enough to model the characteristic of any text and also specific enough to capture the key information of each document including word embeddings and sequential embeddings.

The prior is set to be conjugate with the likelihood for integrating out the cluster parameters during the inference phase (Table 1). As Dirichlet distribution is the conjugate prior of multinomial distribution and Normal-inverse-Wishart (NiW) is the conjugate prior of normal distribution, we used the composition of Dirichlet distribution and NiW distribution to serve as the conjugate prior $\mathbb{G}_0$, which is defined as:

$$\mathbb{G}_0(\theta_k) = \text{Diri}(\delta_k|\beta)\text{NiW}(\mu_s^k, \Sigma_s^k|\Theta_s)\text{NiW}(\mu_e^k, \Sigma_e^k|\Theta_e) \tag{2}$$

where Diri denotes the Dirichlet distribution and NiW denotes the Normal-inverse-Wishart distribution. Θ_s denotes hyper-parameters $\{\mu_{s0}, \lambda_{s0}, \nu_{s0}, \Sigma_{s0}\}$ for the encoder-decoder component and Θ_e denotes hyper-parameters $\{\mu_{e0}, \lambda_{e0}, \nu_{e0}, \Sigma_{e0}\}$ for CBOW word embedding component.

4 Inference via Collapsed Gibbs Sampling

We adopt collapsed Gibbs sampling for inference due to its efficiency. It reduces the dimensionality of the sampling space by integrating out cluster parameters, which leads to faster convergence.

The cluster assignment k for document i is decided based on the posterior distribution $p(c_i = k|\boldsymbol{c}_{\neg i}, \boldsymbol{d}, \theta)$. It can be represented as product of cluster prior and document likelihood.

$$p(c_i|\boldsymbol{c}_{\neg i}, \boldsymbol{d}, \theta) = \frac{p(c_i, \boldsymbol{c}_{\neg i}, \boldsymbol{d}|\theta)}{p(\boldsymbol{c}_{\neg i}, \boldsymbol{d}|\theta)} \propto \frac{p(\boldsymbol{c}, \boldsymbol{d}|\theta)}{p(\boldsymbol{c}_{\neg i}, \boldsymbol{d}_{\neg i}|\theta)} = \frac{p(\boldsymbol{c}|\theta)}{p(\boldsymbol{c}_{\neg i}|\theta)} \frac{p(\boldsymbol{d}|\boldsymbol{c}, \theta)}{p(\boldsymbol{d}_{\neg i}|\boldsymbol{c}, \theta)}$$
$$= p(c_i|\boldsymbol{c}_{\neg i}, \theta) p(d_i|\boldsymbol{d}_{\neg i}, \boldsymbol{c}, \theta) \quad (3)$$

Based on the Chinese restaurant process depiction of DPMM, we have

$$p(c_i|\boldsymbol{c}_{\neg i}, \theta) = p(c_i|\boldsymbol{c}_{\neg i}, \alpha)$$
$$= \begin{cases} \frac{r_{k,\neg i}}{D-1+\alpha} & \text{choose an existing cluster } k \\ \frac{\alpha}{D-1+\alpha} & \text{create a new cluster} \end{cases} \quad (4)$$

$(D - 1)$ is the total number of documents in the corpus excluding current document i.

Given the number of variables introduced in the model, direct sampling from the joint distribution is not practical. Thus, we assume conditional independence on the variables by allowing the factorization of the second term in (3) as:

$$p(d_i|\boldsymbol{d}_{\neg i}, \boldsymbol{c}, \theta) \propto p(w_i|\boldsymbol{d}_{\neg i}, \boldsymbol{c}, \theta) p(e_i|\boldsymbol{d}_{\neg i}, \boldsymbol{c}, \theta) p(s_i|\boldsymbol{d}_{\neg i}, \boldsymbol{c}, \theta) \quad (5)$$

The calculation for each component $p(w_i|\boldsymbol{d}_{\neg i}, \boldsymbol{c}, \theta)$, $p(e_i|\boldsymbol{d}_{\neg i}, \boldsymbol{c}, \theta)$ and $p(s_i|\boldsymbol{d}_{\neg i}, \boldsymbol{c}, \theta)$ is derived below:

$$p(w_i|\boldsymbol{d}_{\neg i}, \boldsymbol{c}, \theta) = p(w_i|c_i = k, \boldsymbol{d}_{k,\neg i}, \beta) = \int p(w_i|\delta_k) p(\delta_k|\boldsymbol{d}_{k,\neg i}, \beta) d\delta_k \quad (6)$$

where the first term in the above integral is

$$p(w_i|\delta_k) = \prod_{t \in w_i} \text{Mult}(t|\delta_k) = \prod_{t=1}^{V} \delta_{k,t}^{u_i^t} \quad (7)$$

$\delta_{k,t}$ is the probability of term t bursting in cluster k and u_i^t is the count of term t in document i. The second term in (6) is

$$p(\delta_k|\boldsymbol{d}_{k,\neg i}, \beta) = \frac{p(\delta_k|\beta) p(\boldsymbol{d}_{k,\neg i}|\delta_k)}{\int_k p(\delta_k|\beta) p(\boldsymbol{d}_{k,\neg i}|\delta_k) d\delta_k} \quad (8)$$

By defining $\Delta(\boldsymbol{\beta}) = \frac{\prod_{k=1}^{K} \Gamma(\beta)}{\Gamma(\sum_{k=1}^{K} \beta)}$ similar to [28], we have

$$p(\delta_k|\boldsymbol{d}_{k,\neg i}, \beta) = \frac{\frac{1}{\Delta(\beta)} \prod_{t=1}^{V} \delta_{k,t}^{\beta-1} \prod_{t=1}^{V} \delta_{k,t}^{u_{k,\neg i}^t}}{\int_k \frac{1}{\Delta(\beta)} \prod_{t=1}^{V} \delta_{k,t}^{\beta-1} \prod_{t=1}^{V} \delta_{k,t}^{u_{k,\neg i}^t} d\delta_k}$$
$$= \frac{1}{\Delta(\boldsymbol{u}_{k,\neg i} + \beta)} \prod_{t=1}^{V} \delta_{k,t}^{u_{k,\neg i}^t + \beta - 1} \quad (9)$$

Based on (7) and (9), (6) becomes

$$\begin{aligned} p(w_i|\boldsymbol{d}_{\neg i},\boldsymbol{c},\theta) &= \int_k \frac{1}{\Delta(\boldsymbol{u}_{k,\neg i}+\beta)} \prod_{t=1}^{V} \delta_{k,t}^{u_{k,\neg i}^t+\beta-1} \prod_{t=1}^{V} \delta_{k,t}^{u_i^t} d\delta_k \\ &= \frac{\prod_{t=1}^{V}\prod_{j=1}^{u_i^t}(u_{k,\neg i}^t+\beta+j-1)}{\prod_{j=1}^{u_i}(u_{k,\neg i}+V\beta+j-1)} \end{aligned} \tag{10}$$

As we see from (10), the high dimensionality challenge of text clustering is naturally circumvented by multiplying one dimension of the vector space at a time. $p(e_i|\boldsymbol{d}_{\neg i},\boldsymbol{c},\theta)$ and $p(s_i|\boldsymbol{d}_{\neg i},\boldsymbol{c},\theta)$ in (5) are derived based on properties of NiW distribution:

$$\begin{aligned} & p(s_i|\boldsymbol{d}_{\neg i},\boldsymbol{c},\theta) = p(s_i|c_i=k,\boldsymbol{d}_{k,\neg i},\theta) \\ &= \int_{\mu_k}\int_{\Sigma_k} p(s_i|\mu_k,\Sigma_k)p(\mu_k,\Sigma_k|c_i=k,\boldsymbol{d}_{k,\neg i},\theta)d\mu_k d\Sigma_k \\ &= \int_{\mu_k}\int_{\Sigma_k} \mathcal{N}(s_i|\mu_k,\Sigma_k)\mathrm{NiW}(\mu_k,\Sigma_k|\Theta_s^{k,\neg i})d\mu_k d\Sigma_k \end{aligned} \tag{11}$$

where μ and Σ are the mean and variance of the sequential embedding, $\Theta_s^{k,\neg i}$ includes $\{\mu_s^{k,\neg i},\lambda_s^{k,\neg i},\nu_s^{k,\neg i},\Sigma_s^{k,\neg i}\}$ which is the hyper-parameter in the NiW distribution of cluster k.

We define the normalization constant $Z(\epsilon,\lambda,\nu,\Sigma)$ of NiW distribution as

$$Z(\epsilon,\lambda,\nu,\Sigma) = 2^{\frac{(\nu+1)\epsilon}{2}}\pi^{\frac{\epsilon(\epsilon+1)}{4}}\lambda^{\frac{-\epsilon}{2}}|\Sigma|^{\frac{-\nu}{2}}\prod_{i=1}^{\epsilon}\Gamma(\frac{\nu+1-i}{2}) \tag{12}$$

where ϵ is the dimensionality of sequential embedding vector. Therefore

$$\begin{aligned} & p(s_i|\boldsymbol{d}_{\neg i},\boldsymbol{c},\theta) \\ &= \int_{\mu_k}\int_{\Sigma_k} \mathcal{N}(s_i|\mu_k,\Sigma_k)\mathrm{NiW}(\mu_k,\Sigma_k|\Theta_s^{k,\neg i})d\mu_k d\Sigma_k \\ &= (\pi)^{\frac{-\epsilon}{2}}(\frac{\lambda_s^k}{\lambda_s^{k,\neg i}})^{\frac{-\epsilon}{2}}\frac{|\Sigma_s^k|^{\frac{-\nu_s^k}{2}}}{|\Sigma_s^{k,\neg i}|^{\frac{-\nu_s^{k,\neg i}}{2}}}\prod_{j=1}^{\epsilon}\frac{\Gamma(\frac{\nu_s^k+1-j}{2})}{\Gamma(\frac{\nu_s^{k,\neg i}+1-j}{2})} \end{aligned} \tag{13}$$

As $\nu_s^k = \nu_s^{k,\neg i}+1$, we have

$$p(s_i|\boldsymbol{d}_{\neg i},\boldsymbol{c},\theta) = (\pi)^{\frac{-\epsilon}{2}}(\frac{\lambda_s^k}{\lambda_s^{k,\neg i}})^{\frac{-\epsilon}{2}}\frac{|\Sigma_s^k|^{\frac{-\nu_s^k}{2}}}{|\Sigma_s^{k,\neg i}|^{\frac{-\nu_s^{k,\neg i}}{2}}}\frac{\Gamma(\frac{\nu_s^k}{2})}{\Gamma(\frac{\nu_s^k-\epsilon}{2})} \tag{14}$$

The derivation of $p(e_i|\boldsymbol{d}_{\neg i},\boldsymbol{c},\theta)$ is analogous to that of $p(s_i|\boldsymbol{d}_{\neg i},\boldsymbol{c},\theta)$ as they are following the same form of distribution, thus,

$$p(e_i|\boldsymbol{d}_{\neg i},\boldsymbol{c},\theta) = (\pi)^{\frac{-\epsilon}{2}}(\frac{\lambda_e^k}{\lambda_e^{k,\neg i}})^{\frac{-\epsilon}{2}}\frac{|\Sigma_e^k|^{\frac{-\nu_e^k}{2}}}{|\Sigma_e^{k,\neg i}|^{\frac{-\nu_e^{k,\neg i}}{2}}}\frac{\Gamma(\frac{\nu_e^k}{2})}{\Gamma(\frac{\nu_e^k-\epsilon}{2})} \tag{15}$$

Algorithm 1 presents the complete inference procedure.

Algorithm 1. Inference of SiDPMM Model

Data : For each document i, the bag of words $\boldsymbol{w_i}$, word embedding $\boldsymbol{e_i}$, sequential embedding $\boldsymbol{s_i}$
Result : Number of clusters K, cluster assignments for each document $\boldsymbol{c}$

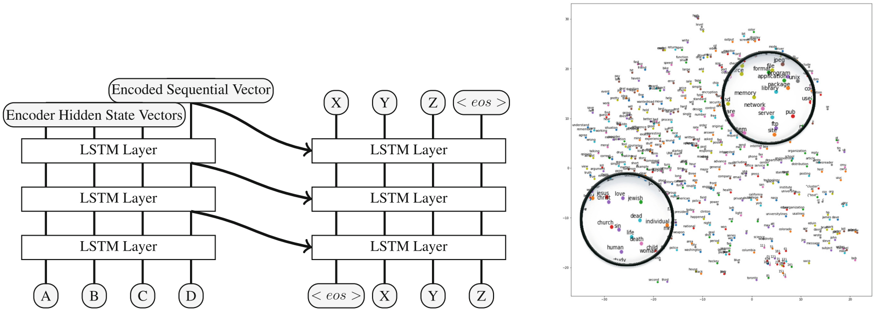
```
/* Initialization                                                    */
1  K=0
2  for each document i do
3  |   compute cluster prior p(c_i|c_{¬i}, α) ▷ (4)
4  |   calculate p(w_i|d_{k,¬i}, c_i = k, θ) ▷ (10)
5  |   calculate p(s_i|d_{k,¬i}, c_i = k, θ) ▷ (14)
6  |   calculate p(e_i|d_{k,¬i}, c_i = k, θ) ▷ (15)
7  |   calculate p(d_i|d_{k,¬i}, c_i = k, θ) ▷ (5)
8  |   sample cluster c_i ∼ p(c_i = k|c_{¬i}, d, θ) ▷ (3)
9  |   if c_i = K + 1 then
10 |   |   K=K+1
11 |   end
12 |   update parameters of cluster c_i
13 end
/* Collapsed Gibbs Sampling, N iterations                            */
14 for Iter= 1 to N do
15 |   for each document i do
16 |   |   delete document i from cluster c_i, update parameters of cluster c_i
17 |   |   if cluster c_i is empty then
18 |   |   |   K=K-1
19 |   |   end
20 |   |   repeat line 3 to line 7
21 |   |   sample a new cluster c_i for document i ▷ (3)
22 |   |   if c_i = K + 1 then
23 |   |   |   K=K+1
24 |   |   end
25 |   |   update parameters of cluster c_i
26 |   end
27 end
```

5 Extraction of Sequential Feature and Synonyms Embedding

In this section, we describe how to extract sequential embeddings with an encoder-decoder component and synonyms embeddings with the CBOW model.

The encoder-decoder component is formed with two LSTM stacks [8], one is for mapping the sequential input data to a fixed length vector, the other is for decoding the vector to a sequential output. To learn embeddings, we set the input sequence and output sequence to be the same. An illustration of the encoder-decoder mechanism is shown in Fig. 2a. The last output of the encoder LSTM stack contains information of the whole phrase. In machine translation, researchers have found the information is rich enough for the original phrase to be decoded into translations of another language [18].

Current state-of-the-art text clustering methods adopt one-hot encoding for word representation. It neglects semantic relationship between similar words.



(a) (b)

Fig. 2. (a) The Encoder-Decoder Component. It is formed by two LSTM stacks, one is for mapping a sequential input data to a fixed-length vector, the other is for decoding the vector to a sequential output. (b) Word embedding of Google News Title Set. Words describing the same topic have similar embeddings and are clustered together

Recently, researchers have shown multiple degrees of similarity can be revealed among words with word embedding techniques [20]. Utilizing such embeddings means we can cluster the documents based on *meaning of words* instead of the word itself. As shown in Fig. 2b, words describing the same topic have similar embeddings and are clustered together. The CBOW model is used to learn word embeddings by predicting each word based on word context (weighted nearby surrounding words). The embedding vector e_i is the average of word embeddings in d_i. Readers are referred to [19] for details about the CBOW model.

6 Experiments

In this section, we will demonstrate the effectiveness of our approach through a series of experiments. The detailed experimental settings are as follows:

Datasets. We run experiments on four diverse datasets including 20 News Group (20NG)[1], Tweet Set[2], and two datasets from [27]: Google News Title Set (T-Set) and Google News Snippet Set (S-Set). The 20NG dataset contains long documents with an average length of 138 while the documents in T-Set and Tweet Set are short with average length less than 10. Phrase structures are sparse in T-Set, while rich in 20NG and S-Set. The Tweet Set contains moderate phrase structures.

Baselines. We compare SiDPMM against two classic clustering methods, K-means and latent Dirichlet allocation (LDA), and two recent methods GSDMM [28] and GSDPMM [27] that are state-of-the-art in nonparametric Bayesian text clustering.

[1] http://qwone.com/~jason/20Newsgroups/.
[2] http://trec.nist.gov/data/microblog.html.

Table 2. NMI scores on various dataset-parameter settings. K is the prior number of clusters for K-means, LDA and GSDMM, set to be four different values including the ground truth for each dataset. K is not used for SiDPMM and GSDPMM. 20 independent runs for each setting.

	K	SiDPMM	SiDPMM-sf[a]	SiDPMM-we[b]	K-means	LDA	GSDMM	GSDPMM
20NG	10	.689 ± .006	.686 ± .005	.680 ± .006	.235 ± .008	.585 ± .013	.613 ± .007	.667 ± .004
	20	.689 ± .006	.686 ± .005	.680 ± .006	.321 ± .006	.602 ± .012	.642 ± .004	.667 ± .004
	30	.689 ± .006	.686 ± .005	.680 ± .006	.336 ± .005	.611 ± .012	.649 ± .005	.667 ± .004
	50	.689 ± .006	.686 ± .005	.680 ± .006	.348 ± .006	.617 ± .013	.656 ± .002	.667 ± .004
T-Set	100	.878 ± .003	.872 ± .003	.877 ± .005	.687 ± .005	.769 ± .012	.830 ± .004	.873 ± .002
	150	.878 ± .003	.872 ± .003	.877 ± .005	.721 ± .009	.784 ± .015	.852 ± .009	.873 ± .002
	152	.878 ± .003	.872 ± .003	.877 ± .005	.720 ± .007	.786 ± .014	.853 ± .009	.873 ± .002
	200	.878 ± .003	.872 ± .003	.877 ± .005	.730 ± .008	.806 ± .013	.868 ± .006	.873 ± .002
S-Set	100	.916 ± .004	.910 ± .005	.902 ± .003	.739 ± .006	.848 ± .005	.854 ± .004	.891 ± .004
	150	.916 ± .004	.910 ± .005	.902 ± .003	.756 ± .006	.850 ± .006	.867 ± .008	.891 ± .004
	152	.916 ± .004	.910 ± .005	.902 ± .003	.757 ± .007	.852 ± .005	.867 ± .009	.891 ± .004
	200	.916 ± .004	.910 ± .005	.902 ± .003	.768 ± .007	.862 ± .004	.885 ± .005	.891 ± .004
Tweet	50	.894 ± .007	.887 ± .006	.884 ± .005	.696 ± .008	.775 ± .012	.844 ± .006	.875 ± .005
	90	.894 ± .007	.887 ± .006	.884 ± .005	.725 ± .007	.797 ± .011	.862 ± .008	.875 ± .005
	110	.894 ± .007	.887 ± .006	.884 ± .005	.732 ± .006	.806 ± .010	.867 ± .006	.875 ± .005
	150	.894 ± .007	.887 ± .006	.884 ± .005	.742 ± .006	.811 ± .012	.871 ± .004	.875 ± .005

[a] SiDPMM model only integrating sequential features.
[b] SiDPMM model only integrating word embeddings.

Metrics. We take the normalized mutual information (NMI) as the major evaluation metric in our experiments since NMI is widely used in this field. NMI scores range from 0 to 1. Perfect labeling is scored to 1 while random assignments tend to achieve scores close to 0.

Encoder-Decoder Component. We truncate the sequence length to be 48 for Tweet Set and Google News dataset and 240 for 20NG dataset. The document with characters length shorter than this sequence length is padded with zeros. The encoder-decoder model is trained for 10 iterations. The length of hidden vectors is set to be 40, and length of input vector is 67 (number of different characters). Weights in the LSTM stack are uniformly initialized to be 0.01. Adam [15] optimizer is used to optimize the network with its learning rate set to 0.01.

Word Embedding Component. The vocabulary size is set to 100,000 which is enough to accommodate most of the words present in the dataset. We set the embedding vector length to be 40. To facilitate training with small datasets such as the Tweet Set, we augment each dataset with a well-known large-scaled text dataset[3] during training. Window size is set to be 1, meaning we only consider the words that are neighbors of the target word as its word context. We apply stochastic gradient descent for optimization with a total of 100,000 descent steps.

[3] http://mattmahoney.net/dc/text8.zip.

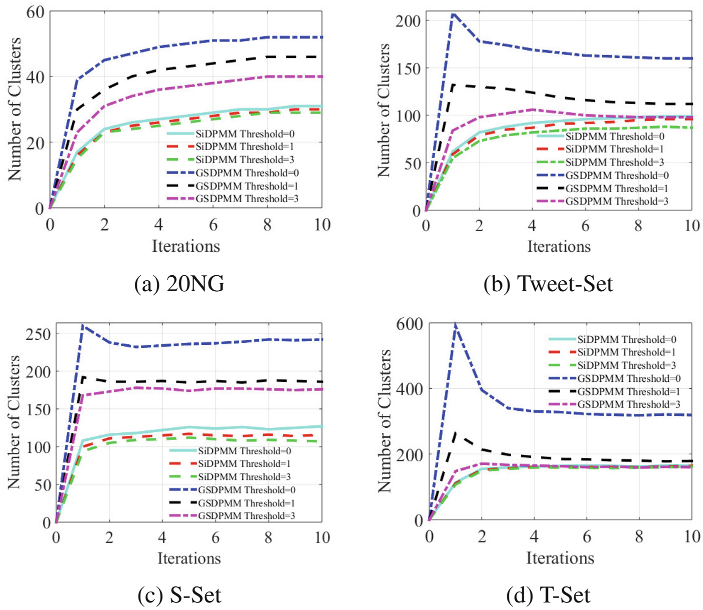


Fig. 3. Number of clusters with size above a given threshold found in each iteration by SiDPMM and GSDPMM. A cluster with size smaller than the given threshold does not count. Plots (a)–(d) are for the datasets 20NG, Tweet-Set, S-Set and T-Set respectively.

Priors. Hyper-parameter α of the Dirichlet process is set to be $0.1 \times |\boldsymbol{d}|$, where $|\boldsymbol{d}|$ is number of documents in the dataset. Hyper-parameter β for the Multinomial modeling of bag of words is $0.002 \times V$, and parameters for the prior NiW distribution of word embedding and sequential embedding are $\{\mu_0 = \mathbf{0}, \lambda_0 = 1, \nu_0 = \epsilon, \Sigma_0 = I\}$.

6.1 Empirical Results

Table 2 reports the mean and standard deviation of the NMI scores across various settings. From Table 2, we observe that SiDPMM outperforms K-means, LDA and GSDMM across all the settings by significant margins. GSDPMM has comparative performance with SiDPMM on T-Set, while SiDPMM performs better in other three datasets. We noted the average length of T-Set is short and phrase structures are scarce in its documents. To unveil the influence of each of the component on the model performance, we included implementation of SiDPMM model only integrating sequential features (denoted as SiDPMM-sf) and SiDPMM model only integrating word embeddings (denoted as SiDPMM-we) into the comparison. We noted the contribution from sequential embedding is significant in 20NG, S-Set and moderate in Tweet-Set.

SiDPMM and GSDPMM can automatically determine the number of clusters. Table 3 shows that number of clusters inferred by SiDPMM are much more accu-

Table 3. Inferred number of clusters by SiDPMM and GSDPMM. Other baseline methods are not included because they require pre-specified number of clusters.

	Number of clusters			Diff. ratio	
	Ground truth	GSDPMM	SiDPMM	GSDPMM	SiDPMM
20NG	20	52	31	160%	**55%**
T-Set	152	323	171	113%	**13%**
S-Set	152	246	126	62%	**17%**
Tweet	110	161	99	46%	**10%**

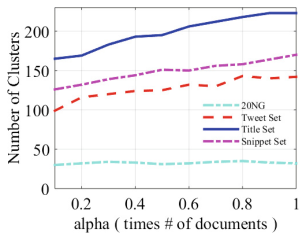


Fig. 4. Number of clusters found by SiDPMM with different α values, revealing the relative strength of prior (compared to likelihood) in determining posterior distribution

rate compared to those from GSDPMM across all the datasets. We can observe that GSDPMM tends to create more clusters than SiDPMM. As illustrated in Fig. 3, many of those clusters created by GSDPMM are quite small; while in constrast, SiDPMM tends to suppress tiny clusters and thus are more robust to outliers. The sequential and word embedding components in SiDPMM are responsible for this regularization effect on number of clusters.

The hyper-parameter α in the Dirichlet process determines the prior probability of creating a new cluster (see Eq. (4)). We explore the influence of different α values on our model. Fig. 4 shows that the number of clusters typically grows with α; as observed for Tweet Set, T-Set and S-Set, but not the case for the 20NG dataset. This reveals the relative strength of prior (compared to likelihood) in determining posterior cluster distribution. The documents in 20NG have large average length (137.5 words per document). In the sampling process, the likelihood dominates the posterior distribution and the small difference caused by different α in the prior distribution is negligible, while for documents with small average length, the difference in likelihood is not significant and thus prior affects more of the posterior distribution.

7 Conclusion

In this paper, we propose a nonparametric Bayesian text clustering method (SiDPMM) which models documents as the joint of bag of words, word embeddings and sequential features. The approach is based on the observation that sequential information plays a key role in the interpretation of phrases and word embedding is very effective for measuring similarity between synonyms. The sequential features are extracted with an encoder-decoder component and word embeddings are extracted with the CBOW model. A detailed collapsed Gibbs sampling algorithm is derived for the posterior inference. Experimental results show our approach outperforms current state-of-the-art methods, and is more accurate in inferring the number of clusters with the desirable regularization effect on tiny scattered clusters.

References

1. Berkhin, P.: A survey of clustering data mining techniques. In: Kogan, J., Nicholas, C., Teboulle, M. (eds.) Grouping Multidimensional Data, pp. 25–71. Springer, Heidelberg (2006). https://doi.org/10.1007/3-540-28349-8_2
2. Blei, D.M., Jordan, M.I.: Variational inference for Dirichlet process mixtures. Bayesian Anal. **1**(1), 121–143 (2006). https://doi.org/10.1214/06-BA104
3. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent Dirichlet allocation. J. Mach. Learn. Res. **3**, 993–1022 (2003)
4. Cai, D., He, X., Han, J.: SRDA: an efficient algorithm for large-scale discriminant analysis. IEEE Trans. Knowl. Data Eng. **20**(1), 1–12 (2008)
5. Cho, K., et al.: Learning phrase representations using RNN encoder-decoder for statistical machine translation. In: EMNLP 2014, pp. 1724–1734. Association for Computational Linguistics, Doha, Qatar, October 2014. http://www.aclweb.org/anthology/D14-1179
6. Duan, T., Pinto, J.P., Xie, X.: Parallel clustering of single cell transcriptomic data with split-merge sampling on Dirichlet process mixtures. Bioinformatics p. bty702 (2018). https://doi.org/10.1093/bioinformatics/bty702
7. Duan, T., Srihari, S.N.: Pseudo boosted deep belief network. In: Villa, A.E.P., Masulli, P., Pons Rivero, A.J. (eds.) ICANN 2016. LNCS, vol. 9887, pp. 105–112. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-44781-0_13
8. Duan, T., Srihari, S.N.: Layerwise interweaving convolutional LSTM. In: Mouhoub, M., Langlais, P. (eds.) AI 2017. LNCS, vol. 10233, pp. 272–277. Springer, Heidelberg (2017). https://doi.org/10.1007/978-3-319-57351-9_31
9. Gomez, J.C., Moens, M.F.: PCA document reconstruction for email classification. Comput. Stat. Data Anal. **56**(3), 741–751 (2012)
10. Gu, Y., Chen, S., Marsic, I.: Deep multimodal learning for emotion recognition in spoken language. CoRR abs/1802.08332 (2018)
11. Gu, Y., Li, X., Chen, S., Zhang, J., Marsic, I.: Speech intention classification with multimodal deep learning. In: Mouhoub, M., Langlais, P. (eds.) AI 2017. LNCS (LNAI), vol. 10233, pp. 260–271. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-57351-9_30
12. Hori, C., Hori, T., Lee, T., Sumi, K., Hershey, J.R., Marks, T.K.: Attention-based multimodal fusion for video description. CoRR abs/1701.03126 (2017)
13. Hotho, A., Staab, S., Maedche, A.: Ontology-based text clustering. In: Proceedings of the IJCAI 2001 Workshop Text Learning: Beyond Supervision (2001)
14. Huang, R., Yu, G., Wang, Z.: Dirichlet process mixture model for document clustering with feature partition. IEEE Trans. Knowl. Data Eng. **25**(8), 1748–1759 (2013)
15. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. CoRR abs/1412.6980 (2014). http://arxiv.org/abs/1412.6980
16. Li, Y., et al.: Towards differentially private truth discovery for crowd sensing systems. CoRR abs/1810.04760 (2018)
17. Liu, M., Chen, L., Liu, B., Wang, X.: VRCA: a clustering algorithm for massive amount of texts. In: IJCAI 2015, pp. 2355–2361. AAAI Press (2015). http://dl.acm.org/citation.cfm?id=2832415.2832576
18. Luong, M., Pham, H., Manning, C.D.: Effective approaches to attention-based neural machine translation. CoRR abs/1508.04025 (2015)
19. Mikolov, T., et al.: Distributed representations of words and phrases and their compositionality. In: Burges, C.J.C., Bottou, L., Welling, M., Ghahramani, Z., Weinberger, K.Q. (eds.) NIPS, pp. 3111–3119. Curran Associates, Inc. (2013)

20. Mikolov, T., Yih, W., Zweig, G.: Linguistic regularities in continuous space word representations. In: HLT-NAACL, pp. 746–751 (2013)
21. Neal, R.M.: Markov chain sampling methods for Dirichlet process mixture models. J. Comput. Graph. Stat. **9**(2), 249–265 (2000)
22. Nie, Y., Han, Y., Huang, J., Jiao, B., Li, A.: Attention-based encoder-decoder model for answer selection in question answering. Front. Inf. Technol. Electron. Eng. **18**(4), 535–544 (2017)
23. Rangrej, A., Kulkarni, S., Tendulkar, A.V.: Comparative study of clustering techniques for short text documents. In: Proceedings of the 20th International Conference Companion on World Wide Web, WWW 2011, pp. 111–112. ACM, New York (2011)
24. Shafiei, M.M., Milios, E.E.: Latent Dirichlet co-clustering. In: Sixth International Conference on Data Mining (ICDM 2006), pp. 542–551, December 2006
25. Wang, F., Zhang, C., Li, T.: Regularized clustering for documents. In: SIGIR 2007, pp. 95–102. ACM, New York (2007)
26. Xun, G., Li, Y., Zhao, W.X., Gao, J., Zhang, A.: A correlated topic model using word embeddings. In: IJCAI 2017, pp. 4207–4213 (2017)
27. Yin, J., Wang, J.: A model-based approach for text clustering with outlier detection. In: 2016 IEEE 32nd International Conference on Data Engineering (ICDE), pp. 625–636, May 2016
28. Yin, J., Wang, J.: A Dirichlet multinomial mixture model-based approach for short text clustering. In: KDD 2014, pp. 233–242. ACM, New York (2014)
29. Yu, G., Huang, R., Wang, Z.: Document clustering via Dirichlet process mixture model with feature selection. In: KDD 2010, pp. 763–772. ACM, New York (2010)
30. Zhang, H., Li, Y., Ma, F., Gao, J., Su, L.: Texttruth: an unsupervised approach to discover trustworthy information from multi-sourced text data. In: KDD 2018, pp. 2729–2737. ACM, New York (2018). https://doi.org/10.1145/3219819.3219977