

A relational background knowledge boosting based topic model for Chinese poems

Lei Peng^{1,2}, Paitoon Porntrakoon¹

¹Vincent Mary School of Science and Technology, Assumption University, Bangkok, Thailand

²Library and Information Science Center, Chongqing Three Gorges Medical College, Chongqing, China

Article Info

Article history:

Received Mar 25, 2024

Revised Apr 9, 2024

Accepted Apr 13, 2024

Keywords:

Gibbs sampling

Latent dirichlet allocation

Short text

TextRank

Topic model

ABSTRACT

Classical Chinese poetry has been increasingly popular in recent years, and modeling its topic is quite a promising area of research. Chinese poems have the characteristic of short in length, but traditional topic models perform poorly when faced with short texts due to the text sparsity. Therefore, topic model should be improved to satisfy the scenario of classical Chinese poems. In this paper, a relational background knowledge boosting based topic model (RBKBTM) was proposed to overcome the text sparsity of Chinese poems. We incorporated background information into the model, which expanded the text content from the semantic perspective. The background knowledge was combined using word embedding and TextRank and was then fed into the core computing process. Subsequently, a new sampling formula was derived. Our proposed model was tested on three different tasks using three different datasets. The results demonstrate that the incorporated background knowledge can effectively overcome text sparsity, improving the performance and effectiveness of the topic model.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Lei Peng

Vincent Mary School of Science and Technology, Assumption University

Bangkok, Thailand

Email: amon5728@163.com

1. INTRODUCTION

Classical Chinese poetry is renowned for its brevity, exquisite language, and profound meanings, making it a precious cultural heritage for future generations [1]. Despite the short length, these poems contain deep and profound artistic conceptions, reflecting the poets' unique insights into society, nature, and human emotions. For example, in Li Bai's "Quiet Night Thoughts": "Before my bed, the bright moonlight is like frost on the ground. I raise my head to gaze at the bright moon and lower it to think of my hometown". This poem expresses the poet's longing for his hometown in a simple four-line verse, with sincere and touching emotions that resonate deeply within the reader's heart. Poetry inspires people to contemplate the true meaning of life and represents the pinnacle of literary and artistic achievement [2].

For a considerable period, classical Chinese poetry, as a representative of Chinese traditional culture, has lacked the means of utilizing information technology for research. Currently, most Chinese natural language processing (NLP) primarily focuses on modern Chinese language, resulting in relatively little research on classical Chinese poetry. The research of classical Chinese poetry thus represents a newly emerging field urgently requiring exploration. With a vast number of Classical Chinese poems, most libraries still rely heavily on manual labor for literature categorization, which is time-consuming and labor-intensive. Utilizing information technology means for the categorization and organization of these poems would be of significant social value.

Topic model is an important generative algorithm in the field of machine learning and is typically categorized as a type of clustering algorithm [3]. Its main application is to induce and model topics from textual data, helping to reveal the hidden thematic structure behind the text [4]. Applying topic modeling to classical Chinese poems would facilitate rapid literature categorization, enhance library management of classical Chinese literatures, and fill the gap in text mining research on classical Chinese corpus. To our knowledge, there has been no research on topic modeling for classical Chinese poems. Here, we hope to take the first step in this attempt. However, topic modeling for classical Chinese poems faces challenges primarily because of the short length of the texts, typically within 20-30 characters, and the extremely low repetition of characters. Thus, it is essentially an issue of topic modeling for short texts, which is currently still under active research.

The attention to modeling topic for short texts was largely due to the rise of social network services (SNS) in the early of this century [5]. In most literature, Twitter, Facebook, Sina Weibo, and Bullet Screens were often used as subjects of study for short texts [6]-[18]. Researchers hoped to uncover the potential conversational topics in social media and grasp social trends. Because topic model was originally proposed and designed for long text scenario, it has no problem when dealing with long text like articles in the newspaper. However, when faced with short texts, it cannot effectively handle the data sparsity and always performs poorly [19]. Assuming we have 2 documents, and the contents are as: i) Doc_1: "The temperature is cool, and I like the temperature." and ii) Doc_2: "I like the temperature." The word "temperature" appears twice in Doc_1 and it appears only once as other words in Doc_2. We, as human beings, can tell that both documents focus on "temperature". But, for the computer algorithm, it may be difficult to provide the focus point of the Doc_2, because each word in Doc_2 is unique. The lower frequency of word co-occurrences would disrupt the statistical calculations of the model. This will make the intermediate results randomly sampled, which affects subsequent computing [19]. Based on the review of existing literature, there are three ways adopted by researchers in tackling the issue of short text topic modeling.

This first strategy is mainly to improve the models' framework to increase the term co-occurrence in text. Some researchers changed the distribution assumption that all words in a short text were drawn from one topic [6], [7], [20]. The bi-term topic model (BTM) and its extensions constructed word-pairs to increase the co-occurrence of the terms in the texts, and assumed that each term in a word-pair was drawn independently from the other term [8], [9]. To avoid the effect of the noisy word in the documents, researchers replaced the symmetric Dirichlet prior with the "Spike and Slab", which worked as a binary discriminator for judging whether a word was representative [10], [20]. In some studies, each document was assumed to have one topic distribution, which could help cluster similar words to obtain topic homogeneity [11]. However, other studies showed that the expanded information considered in these researches was not enough, and the sparsity problem still existed [21].

This second strategy mainly focuses on concatenating short texts into longer pseudo-documents. Some researchers aggregated tweets over authorships and hashtags [12], [13]. To overcome the semantic sparsity problem, location information is utilized to form the special region concept for the model processing phase [14]. Tweet texts that contained words with similar semantic meanings were grouped together [15]. Short texts were aggregated using the similarities of morphemes, of which the computing was mostly done by combining the co-occurrence frequency and the distance between vector representation of terms [16]. In this way, topic models which are not improved were then directly applied to get more important topics from the expanded contexts. Other researchers used the standard topic model for short text by getting together the process of aggregating short texts and generating topics [21], [22].

The third strategy tries to improve the topic inference by using external large datasets as the background knowledge. In order to model topics of short texts, researchers built topic models on long texts, which were designed to share the object domain [23], [24]. The inferring computing of short texts was performed by using background knowledge from topic-related long texts [17]. Some others used the pre-trained word embeddings as background knowledge to infer the number of topics and aggregated short texts into long pseudo-texts [18], [20].

Due to the topic model relying on word frequency counting in its core computations, the first strategy still encounters difficulties with short texts, despite yielding some enhancement in experimental results by improving the model framework. However, essentially, it still fails to address the issue of sparse text. As for the second strategy, researchers indeed obtain longer texts by concatenating short ones. Yet, because these short texts originate from different documents, which are likely to be unrelated, forcibly combining them for topic modeling results in disorderly topics for each document, making it unpredictable and incomprehensible. The third strategy involves utilizing external knowledge for assistance. However, researchers independently calculate external knowledge and the topic model. They either apply external knowledge to text preprocessing or to the results of the topic model, failing to deeply integrate external knowledge into the topic model, thereby underutilizing it.

Through analysis, we found that the first and third methods are more feasible approaches. Therefore, we plan to combine improved modeling with the integration of external knowledge for this research. The significant improvement lies in our deep integration of external knowledge into the core computations of the topic model. Unlike the third strategy mentioned above, where external knowledge is placed outside the topic model, we derive new topic sampling formulas, fundamentally reconstructing it. In essence, our goal is to address the challenge of short texts topic modeling for classical Chinese poems through the incorporation of external knowledge.

We have entered the era of deep learning, and one of the significant accomplishments in the field of NLP is the development of word embedding. This innovative technology has been widely applied to various NLP tasks and has consistently yielded impressive results in recent years [25], [26]. In this paper, our inspiration is to utilize word embedding as the source of external knowledge, which has been demonstrated as an effective way to enrich the semantic understanding and representation of textual data.

The paper is organized in the following way. The section 1 is the introduction, which primarily introduces the background and significance of the research, literature review, and the inspiration of our method. The section 2 presents the model we proposed, including some foundational knowledge, specific details of our proposed model, and formula derivation. The section 3 elaborates on the experimental setup, including the introduction of datasets, comparison methods, and evaluation criteria. The section 4 describes the experimental results, and the section 4 presents the conclusion.

2. METHOD

2.1. Latent dirichlet allocation

Latent dirichlet allocation (LDA) is one of the topic models which is capable of automatically generating topics [27]. It has evolved from the unigram model (UM), latent semantic analysis (LSA), probabilistic latent semantic analysis (PLSA), and finally to LDA [28]. Its fundamental concept is straightforward: the entire corpus emerges from the term generation process. In this framework, there exist two distributions: one for the document-topic relationship, and another for the topic-term relationship. The process entails selecting a topic from the document-topic distribution for each position in a document, followed by selecting a term from the topic-term distribution assigning on this position. Figures 1 illustrates the plate notation of LDA. The LDA can be computed in two primary ways for its core running process. One is Variational Inference, which employs the expectation maximization (EM) algorithm [29]. In the E-step, the coupling relationship between the latent variables is neutralized through variational assumption to obtain the variational distribution. In the M-step, the variational parameters are fixed, and the above expectations are maximized via a series of Newton steps, ultimately yielding the parameters through iterations. Another is Gibbs sampling [30]. In recent years, it has gained popularity due to its simplicity. The underlying idea is to achieve a Markov stationary state by continuously sampling from the distribution. It a special case of Markov chain Monte Carlo (MCMC), and is primarily used in situations involving multidimensional random variables. It continuously samples the topic by marginal probability distribution to finally derive the joint probability distribution. It should be noted that the explanation of the symbols in Figure 1 can be lookup in Table 1. Since the method proposed in the section 2.5 is based on LDA, the symbols in Figure 1 also appear in our model.

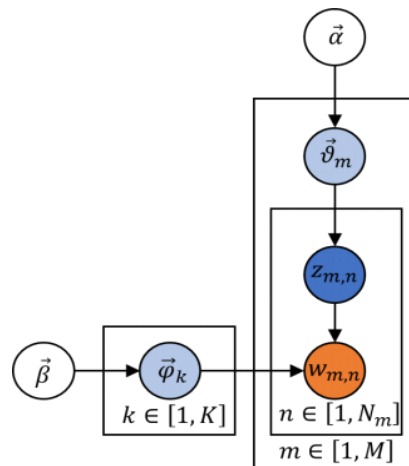


Figure 1. LDA plate notation

2.2. Word embedding

Word embedding is the general name of methods that map words to real number vectors, and it is the product of combining deep learning and NLP, among which Word2Vec is the representative model. Word2Vec was presented by Mikolov *et al.* [31] in 2013, and then implemented by Google. It simplifies the neural language model proposed by Bengio *et al.* [32] in 2003, removes the hidden layer and optimizes the softmax normalization process. It uses a shallow neural network to generate a semantic vector for the words given the unlabeled corpus. Large enough corpus should be used for training process. After running the Word2Vec, each word in the vocabulary would correspond to a vector with n dimensions, in which n should be set by the researcher in advance. This result is also called the distributed presentation for a word. Once obtaining the distributed representation by training the large enough corpus, the relationship (distance) between words can be conveniently measured. For instance, we can compute the word distance like, “Beijing”-“Tokyo” \approx “China”-“Japan”. In this research, we train word embedding model using large classical Chinese corpus. Then, we construct a character similarity matrix by it and feed the matrix into TextRank algorithm to obtain the background knowledge.

2.3. TextRank

TextRank is a graph-based sorting algorithm for keyword extraction and document summary. In TextRank algorithm, the calculation formula was provided by Mihalcea *et al.* [33] in 2004:

$$WS(V_i) = (1 - d) + d * \sum_{V_j \in In(V_i)} \frac{W_{ji}}{\sum_{V_k \in Out(V_j)} W_{jk}} WS(V_j) \quad (1)$$

in which, $WS(V_i)$ represents the weight of sentence i , and the sum on the right represents the contribution of each sentence to this sentence. In a single document, we can roughly think that all sentences are adjacent. W_{ji} represents the similarity of sentence j and i , and $WS(V_j)$ represents the weight of the last iterated sentence V_j . $In(V_i)$ means the precursor nodes of V_i , that the nodes point to V_i . $Out(V_j)$ means the follow-up nodes of V_j , that the nodes point out from V_j . d is the damping factor, generally 0.85 [33].

Original TextRank takes a sentence as the constituent unit of the text, but here we take Chinese character as the constituent unit of the text. Thus, similarity is computed between character. We construct the character similarity matrix and feed it into TextRank. In this way, the weight of each Chinese character can be obtained through a series of calculations. Our objective is to use TextRank to generate background knowledge.

2.4. Background knowledge computing

Background knowledge is an important concept. To give an analogy, background knowledge is akin to the common sense that children acquire as they grow up. Judgments based on this common sense tend to align more closely with human thought processes. For instance, as users of natural language, we understand that ‘pretty’ and ‘beautiful’ have very similar meanings. However, if we do not tell the computer of this, it will treat these two words as unrelated, which could adversely affect the computing results. Here, we use TextRank to generate a corresponding term importance file for each document, which is composed of the weight scores of Chinese characters in the document. This term importance file is regarded as background knowledge. Then we use the below Algorithm 1 to compute the background knowledge vector $\vec{\psi}_d$. It should be noted that this is the vector calculation for a specific document d , and we need to do this for all documents to get the whole Ψ .

Algorithm 1. Compute background knowledge vector $\vec{\psi}_d$

Input: The vocabulary which is a list comprising V terms, the current document with index d , the word embedding pretrained model wv .

Output: The background knowledge vector $\vec{\psi}_d$

```
# initialize the influential vector  $\vec{\psi}_d$  by length of  $V$ 
psi_d = zeros([V])
# count the appeared times of each term
term_list = Document[d].get_terms()
distinguished_term_count_dict = {}
for each term in term_list:
    if each term not in distinguished_term_count_dict:
        distinguished_term_count_dict[each_term] = 1
    else:
        distinguished_term_count_dict[each_term] += 1
# compute distinguished terms list
terms_list_distinguished = distinguished_term_count_dict.keys()
```

```

# compute TextRank value
compute relationship of each two terms in terms_list_distinguished[] using vw.similarity()
--> similarity[][]
compute TextRank value of each term by using matrix of similarity[][] -->
term_trval_dict{}
# get final influential value affected by weight
final_term_weight_dict = {}
for each_term in terms_list_distinguished[]:
    tr_probability = term_trval_dict[each_term]
    count = distinguished_term_count_dict[each_term]
    final_term_weight_dict[each_term] = tr_probability * count
# put this on vocabulary to get vector
for i in [1, V]:
    psi_d[i] = final_term_weight_dict[vocabulary[i]]
normalize psi_d
return psi_d
    
```

2.5. Proposed model

In this paper, we propose a relational background knowledge boosting based topic model (RBKBTM) to solve the text sparsity problem. The text generation process of the proposed model is described as follows: We can regard each document as composed of different positions, and for each position there is a word put on it. The number of positions is the length of the document. Each document corresponds to a topic distribution, and for each iteration we pick a topic from the topic distribution and assign it to a position. Each topic corresponds to a word distribution, and for each position with a topic, we sample a word from the word distribution. For each position with a word, it has a corresponding weight, which is looked up in the document preference. In this way, each position corresponds to a sampled word, and each word has a corresponding weight. The proposed RBKBTM is described as Figure 2.

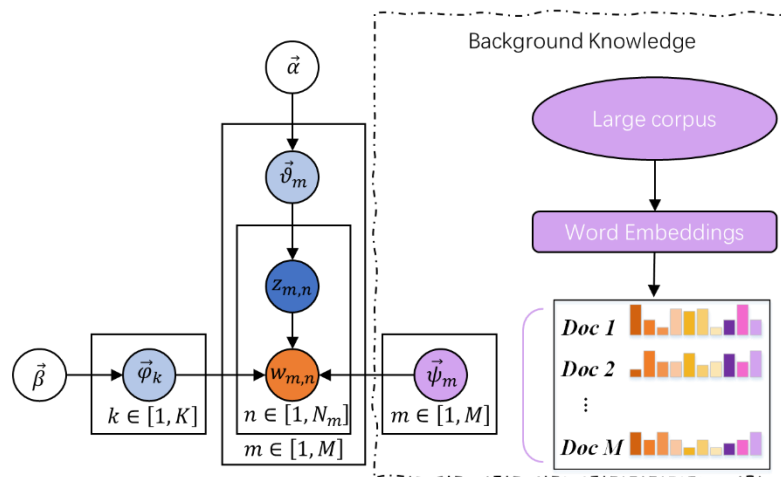


Figure 2. The proposed RBKBTM

As we can see from Figure 2, the right part within the dash line is the background knowledge proposed in this research, and the framework on the left side of the dash line inherits from LDA. We use $\vec{\alpha}$ to generate a $\vec{\theta}_m$ (topic distribution) which corresponds to a document m . In $\vec{\theta}_m$ we sample a topic and assign it to the n -th position of document m , and we denote it as $z_{m,n}$. Similar to the function of $\vec{\alpha}$, we also obtain K instances of $\vec{\varphi}$ (word distribution) which correspond to K topics. For the position assigned the topic $z_{m,n}$ above, we combine the $\vec{\varphi}$ s with $z_{m,n}$ and sample a word from the $\vec{\varphi}_{z_{m,n}}$. We denote this word as $w_{m,n}$. For this word, we look up in $\vec{\psi}_m$ (document preference) to obtain its final weight and assign it to $w_{m,n}$. All notations in Figure 1 are described in Table 1.

In Table 1, we can conclude that, K , M , and N are the numbers. $\vec{\alpha}$ and $\vec{\beta}$ are the vectors, and they would be fed into the Dirichlet distributions to generate the distributions of $\vec{\theta}_m$ and $\vec{\varphi}_k$. $z_{m,n}$ is the sampled topic and it collaborates with $\vec{\varphi}_k$ to generate a word $w_{m,n}$. $\vec{\psi}_m$ is the background knowledge of a specific document, and it would provide the weight of the word mentioned above. The background knowledge is computed through Algorithm 1. Here, we give a description of the text generation process of our model,

as shown in Algorithm 2, in which the topic plate and document plate are the same with LDA, and the only changes are the last two steps in the word plate.

Table 1. The RBKBTM notation explanation table

Notation	Explanation
K	K is the number of topics.
M	M is the number of documents.
N_m	N_m is the length of m -th document, i.e., the count of word positions in document.
$\vec{\alpha}$	$\vec{\alpha}$ is the parameter of document-topic Dirichlet distribution $Dir(\vec{p} \vec{\alpha})$, in which $\vec{\alpha} = (\alpha_1, \dots, \alpha_K)$. The dimension of $\vec{\alpha}$ is K , which corresponds to K topics.
$\vec{\theta}_m$	$\vec{\theta}_m$ is the probability distribution over topics of m -th document. M documents correspond to M $\vec{\theta}$ s. Each $\vec{\theta}$ is drawn from $Dir(\vec{p} \vec{\alpha})$ and $\vec{\theta}$ is a multinomial distribution, in which $\vec{\theta} = (\theta_1, \dots, \theta_K)$. Θ is the set of $\vec{\theta}_m$ s, and $\Theta = \{\vec{\theta}_m\}_{m=1}^M$ ($M \times K$ matrix).
$z_{m,n}$	$z_{m,n}$ is the topic assigned to the n -th position in m -th document, which is affected by θ_i , because θ_i is its probability value.
$\vec{\beta}$	$\vec{\beta}$ is the parameter of topic-word Dirichlet distribution $Dir(\vec{p} \vec{\beta})$, in which $\vec{\beta} = (\beta_1, \dots, \beta_V)$, and V is the vocabulary size.
$\vec{\varphi}_k$	$\vec{\varphi}_k$ is the probability distribution over vocabulary of k -th topic. K topics correspond to K $\vec{\varphi}$ s. Each $\vec{\varphi}$ is drawn from $Dir(\vec{p} \vec{\beta})$ and $\vec{\varphi}$ is a multinomial distribution, in which $\vec{\varphi} = (\varphi_1, \dots, \varphi_V)$. Φ is the set of $\vec{\varphi}_k$ s, and $\Phi = \{\vec{\varphi}_k\}_{k=1}^K$ ($K \times V$ matrix).
$w_{m,n}$	$w_{m,n}$ is the word at the n -th position in m -th document, which is sampled from $\vec{\varphi}_{z_{m,n}}$.
$\vec{\psi}_m$	$\vec{\psi}_m$ is the background knowledge of m -th document, which is computed in the pre-phase. Each $\vec{\psi}$ is a probability distribution over vocabulary, in which $\vec{\psi} = (\psi_1, \dots, \psi_V)$. Ψ is the set of $\vec{\psi}_m$ s, and $\Psi = \{\vec{\psi}_m\}_{m=1}^M$ ($M \times V$ matrix).

Algorithm 2. Text generation process of RBKBTM

```

//Topic plate
for topic k in {1, ..., K} do
    sample a distribution over words  $\varphi_k \sim \text{Dirichlet}(\beta_k)$ 
//Document plate
for document d in {1, ..., D} do
    sample mixture of topics  $\theta_d \sim \text{Dirichlet}(\alpha)$ 
    sample count of words  $W_d \sim \text{Poisson}(\xi_d)$ 
    //word plate
    for word w in {1, W_d} do
        sample topic  $z_{(d,w)} \sim \text{Multinomial}(\theta_d)$ 
        sample word  $w \sim \text{Multinomial}(\varphi_{z_{(d,w)}})$ 
        resizing word weight  $l \sim \psi_{(d, V_w)}$ 

```

2.6. RBKBTM topic inference

Since exact posterior inference is intractable in RBKBTM, we resort to a collapsed Gibbs sampling algorithm for an approximate posterior inference [31]. This method is straightforward to derive, performs comparably in speed to other estimators, and can provide an approximation of a global maximum. Firstly, the joint distribution of the model is obtained as in (2):

$$p(\vec{w}, \vec{z}, \vec{l} | \vec{\alpha}, \vec{\beta}) = L(\Psi) \cdot p(\vec{w} | \vec{z}, \vec{\beta}) \cdot p(\vec{z} | \vec{\alpha}), \quad (2)$$

in which, $L(\Psi)$ is the likelihood of “word re-sizing weights”, and it can be computed as in (3):

$$L(\Psi) = \prod_{m=1}^M \prod_{t=1}^V \psi_{m,t}^{n_m^{(t)}}, \quad (3)$$

in which $n_m^{(t)}$ denotes the number of times that term t has been observed in document m , M is the number of documents, and V is the vocabulary size. Indeed, for each row, only terms in the corresponding document have the positive value of ψ with a positive number of counts. For other terms, $n_m^{(t)}$ is 0. For the second term $p(\vec{w} | \vec{z}, \vec{\beta})$ and third term $p(\vec{z} | \vec{\alpha})$, they are the same inherited from LDA model, in which:

$$p(\vec{w} | \vec{z}, \vec{\beta}) = \prod_{z=1}^K \frac{\Delta(\vec{n}_z + \vec{\beta})}{\Delta(\vec{\beta})}, \vec{n}_z = \{n_z^{(t)}\}_{t=1}^V, \text{ and} \quad (4)$$

$$p(\vec{z} | \vec{\alpha}) = \prod_{m=1}^M \frac{\Delta(\vec{n}_m + \vec{\alpha})}{\Delta(\vec{\alpha})}, \vec{n}_m = \{n_m^{(k)}\}_{k=1}^K. \quad (5)$$

Thus, the joint distribution is finally obtained (6) and (7).

$$p(\vec{w}, \vec{z}, \vec{l} | \vec{\alpha}, \vec{\beta}) = L(\Psi) \cdot p(\vec{w} | \vec{z}, \vec{\beta}) \cdot p(\vec{z} | \vec{\alpha}) \tag{6}$$

$$= \prod_{m=1}^M \prod_{t=1}^V \psi_{m,t}^{n_{m,t}^{(t)}} \cdot \prod_{z=1}^K \frac{\Delta(\vec{n}_z + \vec{\beta})}{\Delta(\vec{\beta})} \cdot \prod_{m=1}^M \frac{\Delta(\vec{n}_m + \vec{\alpha})}{\Delta(\vec{\alpha})} \tag{7}$$

Then the sampling function is derived as (8) and (9).

$$p(z_i = k | \vec{z}_{-i}, \vec{w}, \vec{l}) = \frac{p(\vec{w}, \vec{z}, \vec{l})}{p(\vec{w}, \vec{z}_{-i}, \vec{l})} \propto \frac{p(\vec{w}, \vec{z}, \vec{l})}{p(\vec{w}_{-i}, \vec{z}_{-i}, \vec{l}_{-i})} \tag{8}$$

$$\propto \frac{\prod_{m=1}^M \prod_{t=1}^V \psi_{m,t}^{n_{m,t}^{(t)}} \cdot \prod_{z=1}^K \frac{\Delta(\vec{n}_z + \vec{\beta})}{\Delta(\vec{\beta})} \cdot \prod_{m=1}^M \frac{\Delta(\vec{n}_m + \vec{\alpha})}{\Delta(\vec{\alpha})}}{\prod_{m=1}^M \prod_{t=1}^V \psi_{m,t,-i}^{n_{m,t,-i}^{(t)}} \cdot \prod_{z=1}^K \frac{\Delta(\vec{n}_z + \vec{\beta})}{\Delta(\vec{\beta})} \cdot \prod_{m=1}^M \frac{\Delta(\vec{n}_m + \vec{\alpha})}{\Delta(\vec{\alpha})}} \tag{9}$$

The symbol $-i$ means the word i , y -th position in x -th document, is excluded when computing. After several steps of simplification, we can obtain (10).

$$\propto \psi_i^{n_i} \cdot \frac{\Delta(\vec{n}_z + \vec{\beta})}{\Delta(\vec{n}_z + \vec{\beta})} \cdot \frac{\Delta(\vec{n}_m + \vec{\alpha})}{\Delta(\vec{n}_m + \vec{\alpha})} \tag{10}$$

For the parameter estimation, we can get φ and θ by the definition:

$$\varphi_{k,t} = \frac{n_{k,-i}^{(t)} + \beta_t}{\sum_{t=1}^V (n_{k,-i}^{(t)} + \beta_t)} \tag{11}$$

$$\vartheta_{m,k} = \frac{n_{m,-i}^{(k)} + \alpha_k}{\sum_{k=1}^K (n_{m,-i}^{(k)} + \alpha_k)} \tag{12}$$

For the three strategies mentioned in the introduction, we discarded the second method of text concatenation, and made significant improvements to the third method, which utilizes background knowledge. We deconstructed it to serve as an external feature matrix for documents and integrated it into the core computation of the topic model, addressing the issue of underutilization of external knowledge. Furthermore, we improved the architecture of the topic model by incorporating a module for background knowledge, thereby enhancing the model's effective computation of word frequencies compared to the first approach, which solely focused on enhancing the model architecture by introducing co-occurrence methods for words.

3. EXPERIMENT SETUP

3.1. Experimental design and procedures

The experiment is designed in following steps: (i) Collect and clean the classical Chinese language materials including classical Chinese poems obtained from the Internet. After obtaining the raw corpus, we need to perform the text pre-processing, including text cleaning (HTML tags, mistaken characters, un-regular space, non-Chinese characters, and special Chinese punctuations), character transformation (Traditional Chinese character to Simplified Chinese character). In this process, the “regular expression” technique will be used in the program to filter the content. Classical Chinese poems write using classical Chinese language, which has the characteristic that each character indicates a meaning, so we just perform a simple tokenization, that is, we use space to separate each character in the text. (ii) Implement our proposed RBKBTM according to the content of section 2. We plan to use Python programming language to implement it, which would need the help of python libraries such as Numpy, Scipy, Pandas, NetworkX, and Gensim. (iii) Implement the compared models, such as LDA, BTM, LF-DMM, etc., which will be introduced in section 3.3, and these models might be implemented by other languages. (iv) Run all the models to obtain the outputs. Typically, in the output of each compared model, there are two important matrices used for subsequent evaluation: a topic-word matrix and a document-topic matrix. (v) Set the evaluation metric. We will utilize four metrics, namely *PMI*, *Accuracy*, *PURITY*, and *NMI*, for evaluation, as detailed in section 3.4.

The evaluation metrics will be implemented using Python language. (vi) Obtain the values of various indices and make comparisons to evaluate the results. For clarity and conciseness, we summarized the above process and provided a more visual representation in Figure 3.

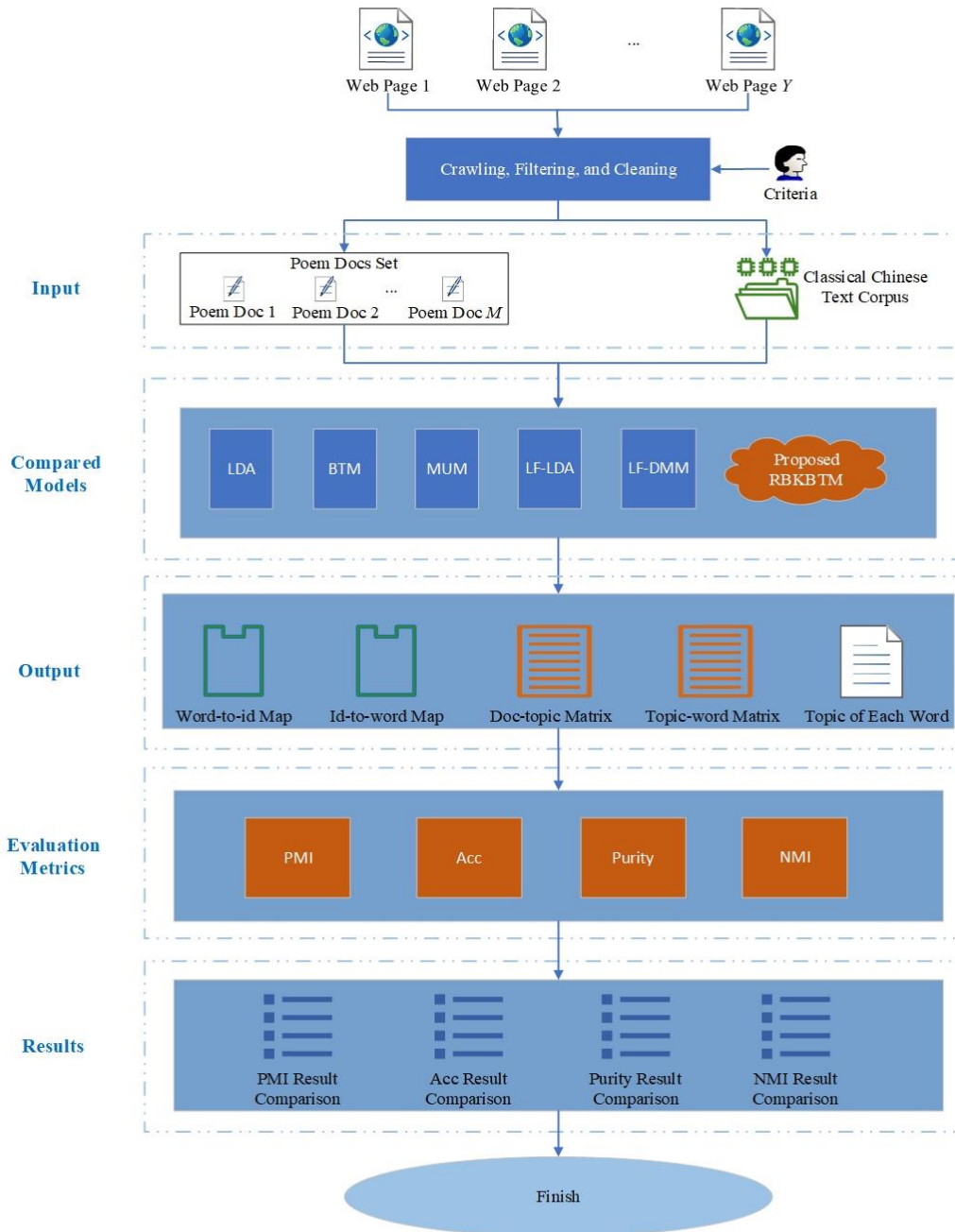


Figure 3. The framework of the whole experimental procedures

3.2. Dataset

For word embedding training, we collect a classical Chinese language corpus of 43 million Chinese characters with vocabulary length of 16,413 which is thought large enough to train the model. This is regarded as background knowledge. For the test data, we use three datasets that can be downloaded from the Internet, which are used to perform the evaluation experiment. The three datasets are introduced as follows.

3.2.1. Theme classification dataset for Chinese classical poetry

The first dataset is theme classification dataset for Chinese classical poetry (TCCP). This is a labeled classical Chinese poetry dataset that was publicly released for use in research of sememe knowledge

in 2022 [34]. The topics of the texts cover the following 9 categories: homesickness, chanting things, landscape, missing someone, meditating on the history, pastoral, frontier war, boudoir resentment, and farewell.

3.2.2. Fine-grained sentimental poetry corpus

The second dataset is fine-grained sentimental poetry corpus (FSCP). This is a dataset used in research of sentiment controllable poetry generation in 2019, and it is a fine-grained sentiment labeled poems dataset provided by THUAIPoet group from Tsinghua University. For the whole poems the labels are divided into 5 classes: negative, implicit negative, neutral, implicit positive, and positive [35].

3.2.3. Chinese classical poetry dataset

The third dataset is Chinese classical poetry dataset (CCPD). This is a publicly released dataset used in research of transformer on classical Chinese poetry in 2023 [36], and it has sentiment labels and topic labels. It contains more than 17,000 documents in this dataset. Table 2 provides detailed information of the datasets used in this paper. The download URLs of the datasets are also provided in the last column. We can see that the TCCP has only topic labels and FSPC has only sentiment labels. We can also know the CCPD dataset contains the highest number of documents, and has both sentiment and topic labels.

Table 2. The dataset statistics table

SN.	Name of datasets	Docs #	Sentiment labeled	Topic labeled	Download URL
(1)	TCCP	3,190	No	Yes	https://github.com/shuizhonghaitong/classification_GAT/tree/master/data
(2)	FSPC	5,000	Yes	No	https://github.com/THUNLP-AIPoet/Datasets/tree/master/FSPC
(3)	CCPD	17,026	Yes	Yes	https://github.com/Chinese-classical-poetry/Dateset

3.3. Compared models

3.3.1. Mixture of unigrams model

In mixture of unigrams model (MUM), one document corresponds to one topic, and the words are drawn from a topic-word multinomial distribution [37]. It assumes topic number is only one, but this is not always applicable for document analysis in real scenes, because it is very common that a document contains two or more topics. Furthermore, if we take the topic as feature for classification task, we would find this feature vector has only one dimension, which does not fit for the classification computing. Thus, for the document decomposition ability evaluation task, the MUM is not considered as a comparable model.

3.3.2. Latent dirichlet allocation

It has always been the baseline compared with the new proposed method. In LDA, the topics and words are both drawn from multinomial distributions, and topic distribution and word distribution are both drawn from dirichlet distribution. Each document is assigned a topic distribution, thus make the inference applicable to real case [27].

3.3.3. Bi-term topic model

BTM is always a popular base line in short text topic modelling evaluation. In this model, two independent words are drawn from a topic-word distribution at a time, and this expands the texts to some extent and solves the text sparsity problem. For the document-topic part, BTM does not change anything with LDA. The topic-word distribution and document-topic distribution are also multinomial distributions and are drawn from Dirichlet distributions. It is reported to achieve good performance [8], however, in some real short text scenes its performance is not as good as claimed [19], [20].

3.3.4. Latent feature latent dirichlet allocation

The idea of this model is to combine a latent feature model with LDA (LF-LDA), which uses a latent feature matrix formed by the topic vector and the pre-trained word vector. An indicator is set to balance the original topic word matrix and the latent feature matrix [38]. It achieved good results in the experiment.

3.3.5. Latent feature-dirichlet multinomial mixture

The idea behind latent feature-dirichlet multinomial mixture (LF-DMM) is the same as LF-LDA, just the main model is using DMM [38]. Both LF-LDA and LF-DMM are short text topic models proposed to

overcome the sparsity problem, and are always used as baseline in researches [3], [22]. It should be noted that these two models need the pre-trained word vector which is just the word embedding result.

3.4. Evaluation tasks and metrics

3.4.1. Topic coherence evaluation task

Topic coherence is the topic organizing ability among the words. Since perplexity is less correlated to human interpretability, better perplexity score does not necessarily indicate the better coherence of topics [39]. Therefore, since after Newman *et al.* [39] in 2010 introduced the topic coherence in documents, people gradually turned to using topic coherence to measure the performance of topic models. Pointwise mutual information (PMI) is an effective way to compute topic coherence, which is calculated as:

$$PMI - Score(t) = \frac{2}{N(N-1)} \cdot \sum_{i=1}^{N-1} \sum_{j=i+1}^N \log \frac{p(t_i t_j)}{p(t_i)p(t_j)} \quad (13)$$

where t_* represents a term, N denotes the number of terms that are on the top list of a specific topic, $p(t_i, t_j)$ is the co-occurrence probability of t_i and t_j , and is computed as “(the number of documents that t_i and t_j both appear)/the number of total documents in the corpus”. $p(t_i)$ is the occurrence probability of t_i , and was computed as “the number of documents that t_i appears/the number of total documents in the corpus”. For PMI score, the larger it is, the better the model is. Here, we set N to 10 in this task to perform the experiment.

3.4.2. Document decomposition ability evaluation task

We use classification accuracy to evaluate the decomposition ability. Once the topic model is performed, the document could be represented by different topics. One important thing of the results of the topic model is the “document-topic matrix”, in which each row corresponds to a document, and each column corresponds to a topic id. If we take each topic as a feature, each row can be regarded as a feature vector, which could be fed into machine learning algorithms, e.g., classification algorithms, for further computation. Each topic should differ largely from others; thus, it makes the classification performance better, and we take this as the “document diversity” index. The accuracy of document classification using topics as feature vector is calculated in (14).

$$Accuracy = \frac{Count(TP) + Count(TN)}{Count(TP) + Count(FP) + Count(TN) + Count(FN)} \quad (14)$$

TP indicates that the data is originally positive and classified into positive class, TN indicates that the data is originally negative and classified into negative class, FP indicates that the data is originally negative but now classified into positive class, and FN indicates that the data is originally positive but now classified into negative class. In this task, we use support vector machine classifier implemented in Python SKlearn library to perform the multilabel classification, and we do 5-fold cross validation.

3.4.3. Clustering quality evaluation task

Two clustering performance evaluation methods are used here, one is purity, and the other is normalized mutual information (NMI) [40], which are always the methods used in topic model evaluation papers [13], [19], [22], [41]. In conventional clustering evaluation, each cluster has its corresponding label, and the purity and NMI are both computed using cluster ID and label ID. The cluster corresponds to the topic here. The purity metric tries to test the ratio of the number of documents assigned “correct” topics to all the documents in the corpus. The purity value is in the range of 0 and 1, and the higher is the better. The purity is computed by (15):

$$Purity = \frac{1}{n} \sum_{c \in (1, \dots, C)} \max_{l \in (1, \dots, L)} (n_{l,c}) \quad (15)$$

where n denotes the number of total documents in the dataset, c is the cluster id assigned by the model of which there are totally C clusters, l is the label id which is the ground truth of which there are totally L labels, $\max(\cdot)$ is a function that returns the largest value of the parameters, and $n_{l,c}$ is the number of documents that belong to both group l and cluster c .

We also use NMI to evaluate overall documents cluster quality. NMI measures the correlations of the cluster assignments and the ground truth groups of the documents. NMI is formally defined as (16):

$$NMI = \frac{\sum_{l,c} n_{l,c} \log\left(\frac{n \cdot n_{l,c}}{n_l \cdot n_c}\right)}{\sqrt{\left(\sum_l n_l \log\frac{n_l}{n}\right)\left(\sum_c n_c \log\frac{n_c}{n}\right)}} \tag{16}$$

where n_l represents the number of documents in group l which is labeled in advance, n_c denotes the number of documents in cluster c which is generated by models, and $n_{l,c}$ is the number of documents that belong to both group l and cluster c , and n is the number of documents in the dataset. When the clustering results perfectly match the ground truth groups, the NMI value will be one. On the other hand, when the clustering results are randomly generated, the NMI value will be close to zero. For NMI, the higher the better.

4. RESULTS AND DISCUSSION

4.1. Model results

For three tasks mentioned above, we set the number of topics to 50, and the results are shown in the following figures. Different models are marked in different colors. In each model, the results are computed from topic number 5 to 50.

4.1.1. Topic coherence evaluation task

For the topic coherence evaluation task, we use PMI scores as the evaluation metric. Figure 4 illustrates the PMI scores, where Figure 4(a) presents the results on CCPD, Figure 4(b) on FSPC, and Figure 4(c) on TCCP. The specific results are shown in the figures.

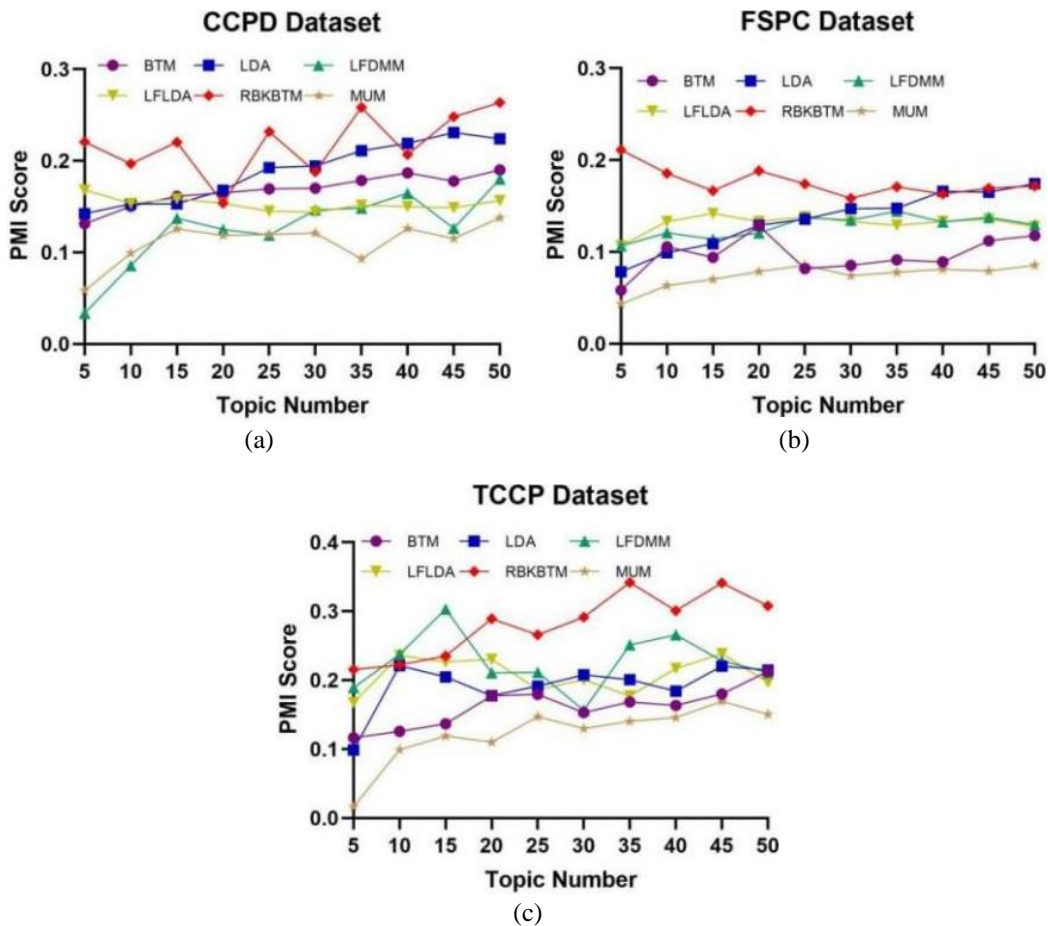


Figure 4. The PMI score result on; (a) CCPD, (b) FSPC, and (c) TCCP dataset

As shown in Figure 4, for most methods in most datasets, the PMI scores increase with topic number, reaching their maximum when topic number is in range from 35 to 50. But there are some exceptions, such as LFLDA in CCPD, RBKBTM and BTM in FSPC, LFDMM and LFLDA in TCCP, where

fluctuations and declines appear. While our method not completely outperforms other methods across the entire range, it still maintains a leading position overall.

4.1.2. Document decomposition ability evaluation task

For document decomposition ability evaluation task, we use the accuracy of classification for evaluation. Figure 5 shows the accuracy values, where Figure 5(a) presents the results on CCPD using emotion labels, and Figure 5(b) on FSPC using sentiment labels.

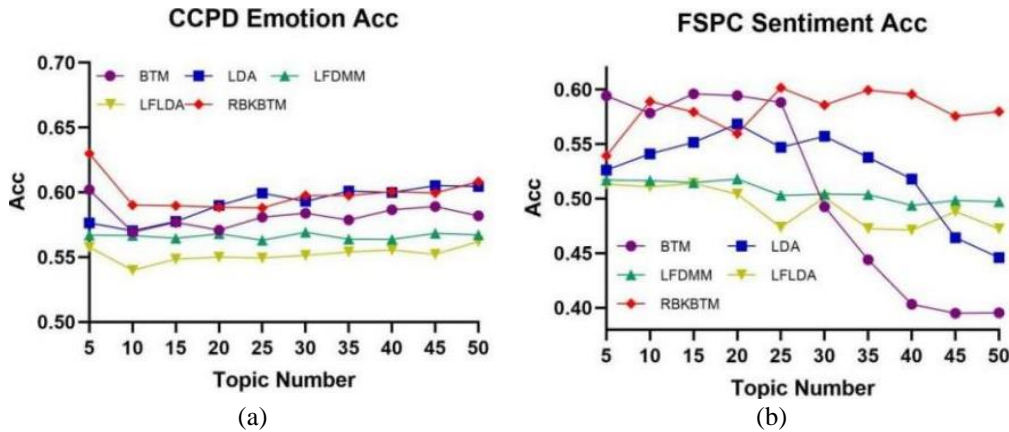


Figure 5. The accuracy calculation on (a) CCPD and (b) FSPC dataset

As shown in Figure 5, for CCPD the accuracy lines of all models are more stable, and for each model, it fluctuates within a relatively narrow range. For FSPC, the results of all models differ significantly, with only RBKBTM, LFDMM and LFLDA being in a relatively stable state. We can see that in both sub-figures, the number of topics has little impact on the accuracy of RBKBTM, and it varies around a constant mean level. The accuracy of our models mostly exceeds 55%, which is higher than that of others.

4.1.3. Clustering quality evaluation task

For assessing clustering quality, we use the purity and NMI as our metric. Illustrated in Figure 6 are the purity curves, with Figure 6(a) showing results from CCPD, and Figure 6(b) from TCCP. Figure 7 shows the NMI curves, with Figure 7(a) representing results from CCPD, and Figure 7(b) from TCCP.

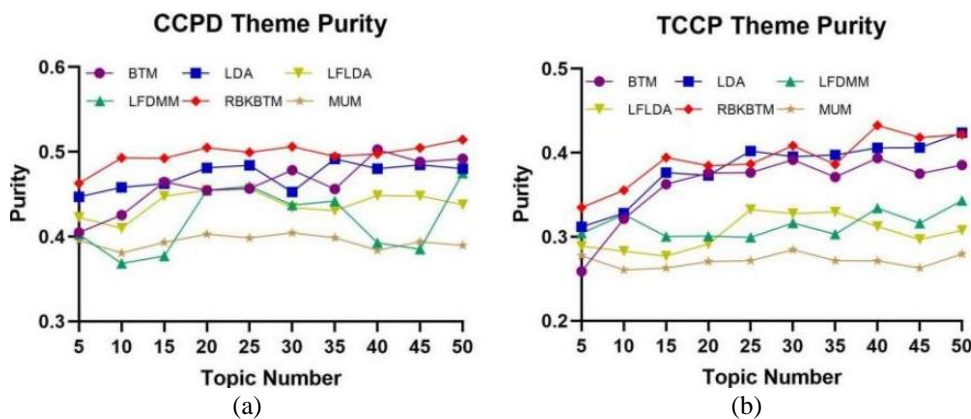


Figure 6. The purity calculation on (a) CCPD and (b) TCCP dataset

For purity, we can see that the RBKBTM result of CCPD remains relatively stable, surpassing nearly all the other models only except for BTM where topic number is 40. For the RBKBTM result of TCCP, it increases with fluctuations, overtaking that of others but LDA, and it was exceeded by LDA when

topic number is 25 and 35. As shown in Figure 6(b), the results of all models are distributed uniformly to some extent. In Figure 6(b), RBKBTM, LDA and BTM are in tier 1 level, while LFLDA, LFDMM and MUM are in tier 2 level. The values of all the results are in area of 0.2 to 0.6.

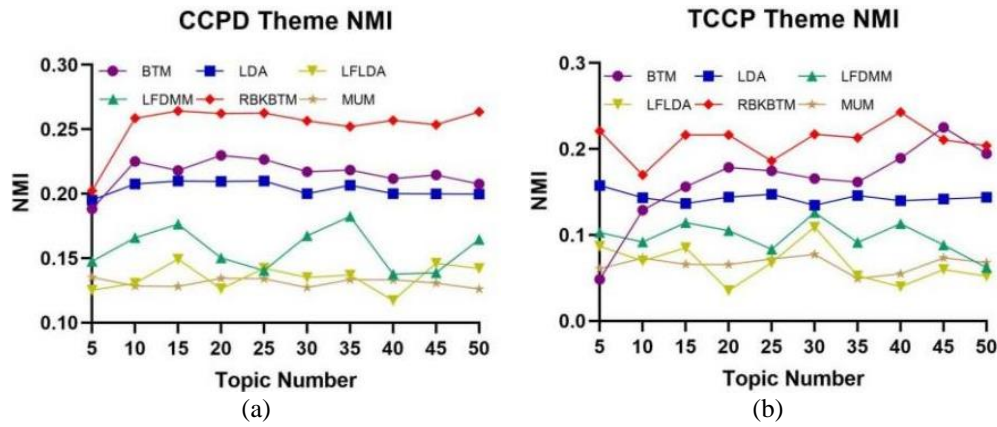


Figure 7. The NMI calculation on (a) CCPD and (b) TCCP dataset

For NMI results, we can see that all models perform relatively steadily on CCPD dataset except for LFDMM with strong fluctuations. While on TCCP, except for the stable performance of LDA, the performance of other models is relatively fluctuating. The RBKBTM result of CCPD is stable, surpassing the results of all the other models, while on TCCP it shows some fluctuation, being only exceeded by BTM when topic number is 45. As is shown in both sub-figures, RBKBTM, BTM, and LDA are in tier 1 level, while LFLDA, LFDMM and MUM are in tier 2 level, and the values of all the results are in area of 0.1 to 0.3. In Figure 7(a), the results of all models appear relatively uniform from the perspective of the mean computing. Overall, the RBKBTM results overtake that of BTM, LDA, LFDMM, LFLDA, and MUM apparently.

4.2. Statistical analysis

For the observed outcomes mentioned above, we took out one curve from a figure and treated each experimental topic number as an independent event. Then we obtained 10 results for each model in each dataset. This enabled us to conduct statistical analysis, e.g., mean, standard deviation (SD), and significance test. The statistical analysis can help evaluate the proposed model. For four evaluation metrics in above three tasks, the results are shown in Figures 8 to 11 respectively, where the error bars are represented using SD. For the significance test, we incorporate ANOVA analysis to evaluate the performance of our proposed model comparing to the baselines, making it a comprehensive assessment. After p-value is obtained, we use the statistical notation * to make a tag, in which * indicates $p \leq 0.05$, ** indicates $p \leq 0.01$, *** indicates $p \leq 0.001$, and **** indicates $p \leq 0.0001$. Usually, a p-value falls below 0.05 and it indicates statistical significance [42]. And we regard p-value of greater than 0.05 as not significant (ns).

4.2.1. Topic coherence evaluation task

As shown in Figure 8, the x-axis represents the 6 models, including the proposed model and the compared models, while the y-axis denotes the PMI score. We can see the PMI scores of RBKBTM outperform all the others in both three datasets. According to p-values, the results of our model show significant differences from most of the compared models. The p-values denoted with “ns” in Figure 8(a), 8(b), and 8(c) are 0.06, 0.06, and 0.07, respectively, which are close to the significance threshold 0.05. This also proves our model demonstrates a reasonably good performance from the perspective of significant analysis.

4.2.2. Document decomposition ability evaluation task

As shown in Figure 9(a), the results of all methods are close, which are around 0.58. All of the models demonstrate comparatively low SDs. The result of RBKBTM shows strong significant differences from models except LDA. In Figure 9(b), the results of BTM, LDA, LFDMM, and LFLDA are around 0.5, while that of RBKBTM is close to 0.6. We can see our model has a relatively small SD compared to BTM and LDA. The accuracy difference between RBKBTM and BTM does not seem significant, but it does not affect the excellence of RBKBTM in the results.

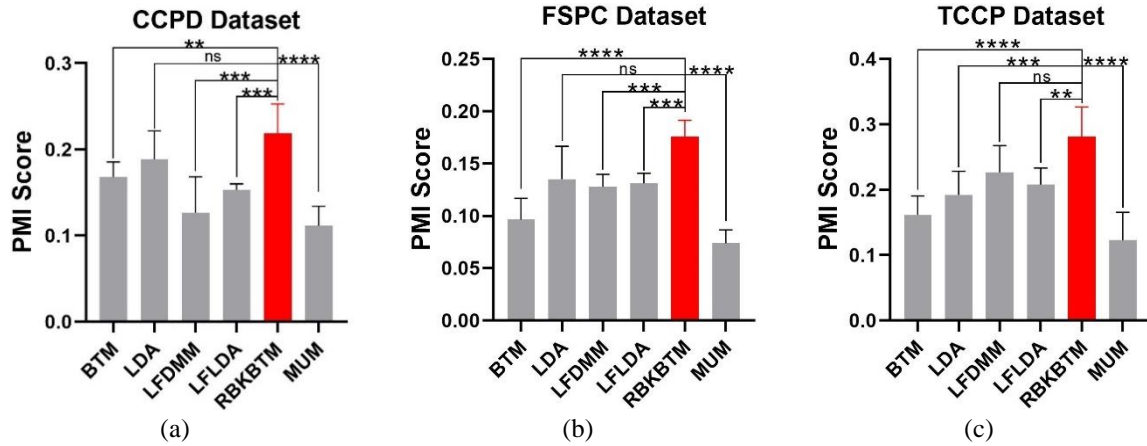


Figure 8. The PMI mean with SD on; (a) CCPD, (b) FSPC, and (c) TCCP dataset

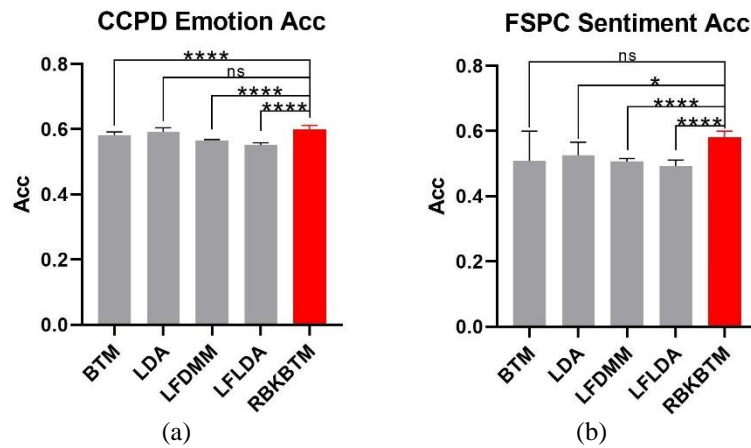


Figure 9. The accuracy mean with SD on (a) CCPD and (b) FSPC dataset

4.2.3. Clustering quality evaluation task

As shown in Figure 10, the result of RBKBTM exhibits significant differences from all other compared models in Figure 10(a), while it shows no significant difference compared to LDA in Figure 10(b). The SDs are generally higher in Figure 10(b) compared to 10(a). We can conclude that, in both Figure 10(a) and 10(b), the purity values of our model exceed those of the others, demonstrating the stability of our model.

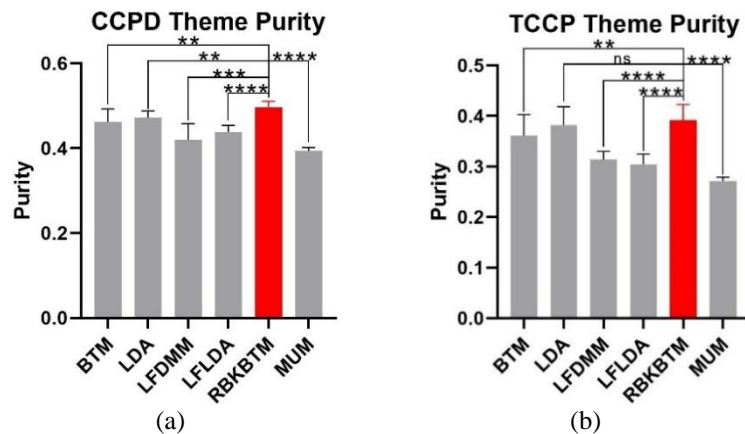


Figure 10. The purity mean with SD on (a) CCPD and (b) TCCP dataset

As shown in Figure 11, our model performs well in the significance tests in both sub-figures, especially in Figure 11(b), where our model shows strong significance over all other models. For the mean value, our method surpasses others by a large margin and demonstrates a good performance. Among both Figure 11(a) and Figure 11(b), the NMI scores of BTM and LDA are closest to our results, but their values still have some discrepancy compared to our method. We can also see that the SDs in Figure 11(a) are all greater than in Figure 11(b).

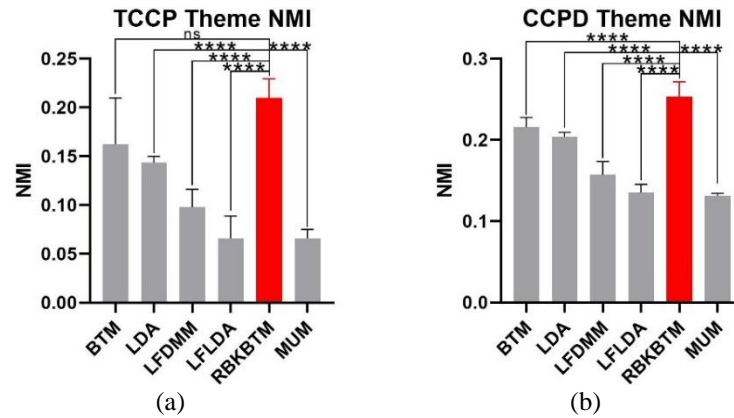


Figure 11. The NMI mean with SD on (a) CCPD and (b) TCCP dataset

4.3. Discussion

We evaluate our proposed model using various metrics in NLP tasks, with a focus on topic coherence, document decomposition ability, and clustering quality. Performance varies across tasks and datasets, with different trends observed in the results curve details. While some methods demonstrate improved performance with an increase in the number of topics, exceptions exist, showing fluctuations and declines in certain scenarios. Despite not consistently outperforming others, our method maintains a leading position overall, as evidenced by its competitive performance across multiple evaluation metrics and datasets. Overall, our method has shown promising results in terms of mean performance across all three tasks. Additionally, we observe that BTM and LDA demonstrate relatively strong baseline performance levels among the compared methods.

Comparing with others, our model not only surpasses them but also demonstrates stability, specifically, our SD remains at a lower level across all three tasks. Moreover, the results of significant differences also demonstrate the effectiveness of our method. One possible reason for this could be the incorporation of extensive classical Chinese corpus as background knowledge, as Chinese poems were written using classical Chinese language by ancient Chinese scholars. During the training of background knowledge, we acquired a vast amount of classical Chinese language data that ancient Chinese scholars studied in school. Integrating this background knowledge significantly enhances the model's performance. However, it also results in additional time consumption during runtime when loading the background knowledge, impacting computational efficiency to some extent.

Our research aims to construct a topic model suitable for classical Chinese poems. Due to the short length of the text, traditional topic models cannot be effectively applied. Therefore, by integrating background knowledge, we enhance the performance of the topic model. Topic model is an important research area in NLP, with its outcomes applicable to text clustering, sentiment analysis, document classification, information retrieval, and more. Our proposed model expands the application of topic models in the domain of Chinese short texts, particularly in poems. While we have improved the performance of the topic model, the inclusion of background knowledge introduces some computational overhead, which is an unavoidable issue. Thus, finding suitable methods to reduce the runtime of the topic model is a future research direction. Overall, although there is an increase in computational time, it does not diminish the robustness and effectiveness of our model.

5. CONCLUSION

Classical Chinese poems lack information technology for research. To address this gap, we proposed a Relational Background Knowledge Boosting based Topic Model in this paper to solve the problem of short

text. The text sparsity was addressed by incorporating background knowledge, which was created with the combination of TextRank and word embedding. In experiments, we compared our model with five others and found it to be the best in overall performance. Our research can aid in semantic understanding and information extraction from classical Chinese poems, as well as facilitate text classification and clustering. With the rising popularity of bullet screen culture, which often incorporates classical poems in comments, our proposed model can analyze such data to identify sentiment tendencies and emotional nuances under different topics, thereby laying the groundwork for personalized recommendations and user modeling. Additionally, this research contributes to enhancing library management efficiency, providing valuable insights for conducting traditional Chinese social analysis, and advancing our understanding of the development of Chinese history. In the future, we will explore how to use distributed computing to improve the model's running efficiency.

ACKNOWLEDGEMENTS

This work was supported by the Chongqing Three Gorges Medical College of China (No. 2019XZYB13) and by the Chongqing Association of Higher Education under Chongqing Municipal of China (No. CQGJ21B128).




REFERENCES

- [1] Y. Dong, H. Zheng, S. X. Wu, F. Huang, S. Peng, S. B. Sun, and H. Zeng, "The effect of Chinese pop background music on Chinese poetry reading comprehension," *Psychology of Music*, vol. 50, no.5, pp. 1544-1565, 2022, doi: 10.1177/03057356211062940.
- [2] J. D. Lee and T. Wong, "Glimpses of ancient China from classical Chinese poems," in *Proc. International Conference on Computational Linguistics*, 2012, pp. 621-632.
- [3] C. D. P. Laureate, W. Buntine, and H. Linger, "A systematic review of the use of topic models for short text social media analysis," *Artificial Intelligence Reviews*, vol. 56, pp. 14223-14255, doi: 10.1007/s10462-023-10471-x.
- [4] A. Rania, Y. Tet, and B. Morad, "Using topic modeling methods for short-text data: a comparative analysis," *Frontiers in Artificial Intelligence*, vol. 3, 2020, doi: 10.3389/frai.2020.00042.
- [5] B. A. H. Murshed, S. Mallappa, and J. Abawajy, "Short text topic modelling approaches in the context of big data: taxonomy, survey, and analysis," *Artificial Intelligence Reviews*, vol. 56, pp. 5133-5260, 2023, doi: 10.1007/s10462-022-10254-w.
- [6] J. Yin and J. Wang, "A model-based approach for text clustering with outlier detection," *IEEE Xplore*, 2016, doi: 10.1109/ICDE.2016.7498276.
- [7] J. Yin, D. Chao, Z. Liu, W. Zhang, X. Yu, and J. Wang, "Model-based clustering of short text streams," *The 24th ACM SIGKDD international conference on knowledge discovery & data mining*, 2018, doi: 10.1145/3219819.3220094.
- [8] X. Cheng, X. Yan, Y. Lan, and J. Guo, "BTM: topic modeling over short texts," *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 17, pp. 2933-2946, 2014, doi: 10.1109/TKDE.2014.2313872.
- [9] J. Chen, Z. Gong, and W. Liu, "A dirichlet process biterm-based mixture model for short text stream clustering," *Applied Intelligence*, vol. 55, no. 10, pp. 1614-1624, 2020, doi: 10.1007/s10489-019-01606-1.
- [10] T. Lin, W. Tian, Q. Mei, and H. Cheng, "The dual-sparse topic model," *The 23rd international conference on World wide web*, 2014, doi: 10.1145/2566486.2567980.
- [11] J. He, L. Li, Y. Wang, and X. Wu, "Targeted aspects oriented topic modeling for short texts," *Applied Intelligence*, vol. 55, no. 13, pp. 2394-2409, 2020, doi: 10.1007/s10489-020-01672-w.
- [12] J. Weng, E. Lim, J. Jiang, and Q. He, "TwitterRank: finding topic-sensitive influential twitterers," *Web Search and Data Mining*, 2010, doi: 10.1145/1718487.1718520.
- [13] R. Mehrotra, S. Sanner, W. Buntine, and L. Xie, "Improving LDA topic models for microblogs via tweet pooling and automatic labeling," in *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval - SIGIR '13*, 2013, doi: 10.1145/2484028.2484166.
- [14] F. Kou, J. Du, C. Yang, Y. Shi, M. Liang, Z. Xue, and H. Li, "A multi-feature probabilistic graphical model for social network semantic search," *Neurocomputing*, vol. 341, pp. 72-83, doi: 10.1016/j.neucom.2018.03.086.
- [15] L. Hong and B. D. Davison, "Empirical study of topic modeling in Twitter," *Knowledge Discovery and Data Mining*, 2010, doi: 10.1145/1964858.1964870.
- [16] P. V. Bicalho, M. Pita, G. F. S. Pedrosa, A. Lacerda, and G. L. Pappa, "A general framework to expand short text for topic modeling," *Information Sciences*, vol. 398, pp. 71-86, 2017, doi: 10.1016/j.ins.2017.02.007.
- [17] O. Jin, N. N. Liu, K. Zhao, Y. Yu, and Q. Yang, "Transferring topical knowledge from auxiliary long texts for short text clustering," in *Proceedings of the 20th ACM international conference on Information and knowledge management - CIKM '11*, 2011, doi: 10.1145/2063576.2063689.
- [18] J. Qiang, P. Chen, T. Wang, and X. Wu, "Topic modeling over short texts by incorporating word embeddings," in *Advances in Knowledge Discovery and Data Mining: 21st Pacific-Asia Conference*, PAKDD 2017, Jeju, South Korea, May 23-26, 2017, doi: 10.48550/arxiv.1609.08496.
- [19] J. Qiang, Z. Qian, Y. Li, Y. Yuan, and X. Wu, "Short text topic modeling techniques, applications, and performance: a survey," *IEEE Transactions on Knowledge and Data Engineering*, vol. 39, no. 8, pp. 1432-1450, 2020, doi: 10.1109/tkde.2020.2992485.
- [20] J. Chen, Z. Gong, and W. Liu, "A nonparametric model for online topic discovery with word embeddings," *Information Sciences*, vol. 509, pp. 37-52, 2019, doi: 10.1016/j.ins.2019.07.048.
- [21] Y. Zuo, C. Li, H. Lin, and J. Wu, "Topic modeling of short texts: a pseudo-document view with word embedding enhancement," *IEEE Transactions on Knowledge and Data Engineering*, vol. 6, 2023, doi: 10.1109/tkde.2021.3073195.
- [22] X. Quan, C. Kit, Y. Ge, and S. J. Pan, "Short and sparse text topic modeling via self-aggregation," in *Proceedings of the 24th International Conference on Artificial Intelligence (IJCAI'15)*, 2015, doi: 10.5555/2832415.2832564.




- [23] X. H. Phan, C. T. Nguyen, D. T. Le, L. M. Nguyen, S. Horiguchi, and Q. T. Ha, "A hidden topic-based framework toward building applications with short web documents," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 12, pp. 966-981, 2010, doi: 10.1109/tkde.2010.27.
- [24] X. H. Phan, L. M. Nguyen, and S. Horiguchi, "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*, 2008, doi: 10.1145/1367497.1367510.
- [25] Y. Li and T. Yang, "Word embedding for understanding natural language: a survey," in *Studies in Big Data*, vol. 26. Springer, Cham, 2018, doi: 10.1007/978-3-319-53817-4_4.
- [26] B. Wang, A. Y. Wang, F. Chen, Y. Wang, and C. J. Kuo, "Evaluating word embedding models: methods and experimental results," *APSIPA Transactions on Signal and Information Processing*, vol. 13, no. 1, 2019, doi: 10.1017/atsip.2019.12.
- [27] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of Machine Learning Research*, vol. 8, pp. 998-1027, 2003, doi: 10.5555/944919.944937.
- [28] S. H. Mohammed and S. Al-augby, "LSA & LDA topic modeling classification: comparison study on e-books," *Indonesian Journal of Electrical Engineering and Computer Science (IJECS)*, vol. 19, no. 1, pp. 353-362, 2020, doi: 10.11591/ijeecs.v19.i1.pp353-362.
- [29] Z. Li and M. Li, "Impact of missing data on EM algorithm under rayleigh distribution," *Indonesian Journal of Electrical Engineering and Computer Science (IJECS)*, vol. 12, no. 6, pp. 4717-4723, 2014, doi: 10.11591/telkomnika.v12i6.5491.
- [30] B. Subeno, R. Kusumaningrum, and F. Farikhin, "Optimisation towards latent dirichlet allocation: its topic number and collapsed gibbs sampling inference process," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 8, no. 5, pp. 3204-3213, 2018, doi: 10.11591/ijece.v8i5.pp3204-3213.
- [31] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," *Neural Information Processing Systems*, vol. 31, pp. 3116-3124, 2013, doi: 10.48550/arXiv.1310.4546.
- [32] Y. Bengio, R. Ducharme, and P. Vincent, "A neural probabilistic language model," *Journal of Machine Learning Research*, 2003, doi: 10.5555/944919.944966.
- [33] R. Mihalcea and P. Tarau, "TextRank: bringing order into text," *ACL Anthology*, 2004.
- [34] J. Zhao, T. Bai, Y. Wei, and B. Wu, "PoetryBERT: pre-training with sememe knowledge for classical Chinese poetry," *International Conference on Data Mining and Big Data*, 2022, pp. 374-389, doi: 10.1007/978-981-19-8991-9_26.
- [35] H. Chen, X. Yi, M. Sun, W. Li, C. Yang, and Z. Guo, "Sentiment-controllable Chinese poetry generation," in *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, 2019, pp. 4925-4931, doi: 10.24963/ijcai.2019/684.
- [36] B. Wu, Y. Wei, Y. Zhu, and L. Hu, "Knowledge-guided transformer for joint topic and sentiment analysis of Chinese classical poetry," *Research Square*, 2023. doi: 10.21203/rs.3.rs-2725320/v1.
- [37] K. Nigam, A. K. McCallum, S. Thrun, and T. Mitchell, "Text classification from labeled and unlabeled documents using EM," *Machine Learning*, vol. 44, no. 2/3, pp. 108-139, 2000, doi: 10.1023/a:1007692713085.
- [38] D. Q. Nguyen, R. Billingsley, L. Du, and M. Johnson, "Improving topic models with latent feature word representations," *Transactions of the Association for Computational Linguistics*, vol. 8, pp. 304-318, 2015, doi: 10.1162/tac1_a_00140.
- [39] D. Newman, J. H. Lau, K. Grieser, and T. Baldwin, "Automatic evaluation of topic coherence," in *Proceedings of the 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, 2010, pp. 105-113.
- [40] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to information retrieval*. Cambridge: Cambridge University Press, 2008.
- [41] X. Li, Y. Wang, A. Zhang, C. Li, J. Chi, and J. Ouyang, "Filtering out the noise in short text topic modeling," *Information Sciences*, vol. 461, pp. 88-101, 2018, doi: 10.1016/j.ins.2018.04.071.
- [42] V. Bewick, L. Cheek, and J. Ball, "Statistics review 12: survival analysis," *Critical Care*, vol. 8, pp. 389-394, 2004, doi: 10.1186/cc2955.

BIOGRAPHIES OF AUTHORS



Lei Peng    is a Ph.D. candidate at the Vincent Mary School of Science and Technology at Assumption University, Thailand. He completed his B.Eng. in the Computer Science and Technology department at Shangqiu University of China and his M.Eng. in the Computer Science and Technology department at Guizhou University of China. His research focuses on information retrieval, text mining, network security, and application theory of information technology. He can be contacted at email: amon5728@163.com.



Paitoon Porntrakoon    received his Ph.D. in Information Technology from Assumption University, Thailand in the year 2018. He is currently the Graduate Program Director in Information Technology of the Vincent Mary School of Science and Technology at Assumption University, Thailand. His research interests are similarity searching, location detection, trust and distrust, social commerce, and Thai sentiment analysis. He can be contacted at email: paitoon@scitech.au.edu.