TextTiling Text Segmentation Based on Hierarchical Dirichlet Process Model

Wei Ye¹, Xiaogang Gong¹, Quan Zhang², Xiaoming Ju²
¹State Grid Zhejiang Electric Power Company Information and Communication Branch, HangZhou,310007
²East China Normal University, School Computer Science and Software Engineering, shanghai, 200062
Author: Wei Ye; email: ye_wei@zj.sgcc.com.cn; phone: 13605812779
Author: Xiaogang Gong; email: gong_xiaogang@zj.sgcc.com.cn; phone: 15858103414
Author: Quan Zhang; email: 51174500157@stu.ecnu.edu.cn; phone: 18852951520
Author: Xiaoming Ju; email: xmju@sei.ecnu.edu.cn; phone: 13774272718

Abstract: Text segmentation plays an extremely important role in many areas such as abstract extraction and information retrieval. Topic model is one of important methods in text segmentation. However, the current methods based on topic model generally rely on the manual setting of the number of topics. In order to improve the efficiency of text segmentation, this paper proposes a TextTiling text segmentation method based on Hierarchical Dirichlet process (HDP) model. Firstly, the method uses the HDP model to obtain a vector representation of the text in the topic space, which can also automatically generate the number of topics. Applying the theme vector to the TextTiling segmentation algorithm can implement text segmentation. The results show that the method can effectively improve the performance of text segmentation, and be free from manually set topics.

1. Introduction
With the rapid development of the network, human beings have gradually entered into a new era of network. The growth of information resources has grown at an explosive rate. All kinds of massive text information bring convenience to human beings. Meanwhile, they pose a huge challenge to text processing and analysis. Text segmentation is a critical step in solving this problem. It refers to dividing text according to the principle of subject-related, so that each semantic paragraph has the maximum subject-related inside and the minimum relevance between paragraphs [1]. According to this principle, it is possible to look for boundaries of different subjects.

Text segmentation has been widely used in the fields of information extraction and abstract generation [2]. For example, in information extraction, text segmentation can help users narrow down the scope of document retrieval and accurately locate in a short time, effectively improving the efficiency of information extraction. In abstract generation, text segmentation can divide text into several subtopics based on dividing text, using the words of the representative sentences under each word topic to form a theme idea. Then it can cover the whole text information more comprehensively.

Most of the current text segmentation methods use the topic model method, which can improve the accuracy of text segmentation. One of the first unsupervised linear topic segmentation algorithms was
introduced by Hearst (1997): TextTiling [3] segments texts in linear time by calculating the similarity between two blocks of words based on the cosine similarity. The calculation is accomplished by two vectors containing the number of occurring terms of each block. In C99 (Choi, 2000) [4] an algorithm was introduced that uses a matrix-based ranking and a clustering approach in order to relate the most similar textual units and to cluster groups of consecutive units into segments. The TopicTiling text segmentation algorithm [5,6,7] is based on the TextTiling algorithm, which uses the Latent Dirichlet Allocation [8] to segment the document and perform segmentation in linear time, which greatly simplifies other LDA-based segmentation methods. The shortcoming is that the traditional methods based on the topic model segmentation generally rely on the manual setting of the number of topics, and the number of topics set too high will cause over-fitting, and the setting too low will cause problems such as insufficient text representation [6]. Therefore, this paper uses HDP [9,10] model for text segmentation, and proposes a text segmentation method based on HDP model. This method gets rid of the dependence on the number of artificially set topics. The main idea is to obtain the vector representation of the text in the topic space through the HDP model, and then use the theme vector for the TextTiling segmentation algorithm to realize the text segmentation. The experimental results demonstrate the effectiveness of the proposed method.

The rest of the paper is organized as follows: Section 2 briefly introduces the theoretical basis of hierarchical Dirichlet and demonstrates the feasibility of this theory based on segmentation. In section 3, the text segmentation method based on the HDP model is described in detail. Section 4 presents the experiments and discusses the experimental results. In Section 5 conclusions are drawn.

2. Hierarchical Dirichlet Process Model

2.1. The basic principle of HDP model

The hierarchical Dirichlet process (HDP) is a nonparametric model extension of the LDA model. It is a stochastic process applied to the nonparametric Bayesian model and a multi-layered form of the Dirichlet process hybrid model, which can be applied to construct a myriad of basic component mixing models. The model can not only solve the problem of sharing multiple clusters between multiple documents, but also can be applied to generate the number of topics automatically and assign a topic ID to each word. Its application in the field of probabilistic topic models is also very extensive. The directed graph of HDP is shown in Figure 1.

![Fig.1 Directed graphical representations of hierarchical Dirichlet process](image)

For a collection containing \( J \) texts, the subject of each document is derived from the base distribution \( H \), which ensures that an unlimited number of topics are shared in each document. As can be seen from the figure 1. firstly, the Dirichlet process is composed of the aggregation parameter \( \gamma \) and the base distribution \( H: G_0 \sim DP(\gamma, H) \). Then, based on \( G_0 \), with \( \alpha_0 \) as the aggregation parameter, construct a Dirichlet process mixture model for each text: \( G_j \sim DP(\alpha_0, G_0), j = 1, 2, \ldots, M \). Finally, the Dirichlet process is used as the prior distribution to construct the Dirichlet process mixture model. The constructed Dirichlet process hybrid model given as:

\[
\theta_{ij} | G_j \sim G_j, X_{ij} | \theta_{ij} \sim F(\theta_{ij})
\]  

(1)
where the $F(\theta_{ji})$ function represents the distribution of the observed variable $\theta_{ji}$ in the case of a given parameter $X_{ji}$; The parameter $\theta_{ji}$ condition independently obeys the $G_0$ distribution, while the observed variable $X_{ji}$ condition independently obeys the $F(\theta_{ji})$ distribution, and $X_{ji}$ represents the word $i$ of the text $j$.

2.2. The Chinese restaurant franchise

There are many construction methods for the HDP model [9]: The Stick-breaking construct and the Chinese restaurant franchise construct. In this section we describe an analog of the Chinese restaurant process for hierarchical Dirichlet processes that we refer to as the “Chinese restaurant franchise.” In the Chinese restaurant franchise, the metaphor of the Chinese restaurant process is extended to allow multiple restaurants which share a set of dishes. In the model, there are $M$ restaurants, each of which can accommodate an unlimited number of tables. Each table shares the same menu, and the menus of each restaurant are the same. When a customer enters a restaurant, the customer can pick a table at will (sampled from the $G_0$ topic that appears in $\theta_a$), or choose whether to order dishes that are also popular in other restaurants (sampled from the $G_m$ topic that appears in several $\theta_a$).

In this setup, the restaurants correspond to groups and the customers correspond to the factors $X_{ji}$. The parameter sequence $(\phi_k)_{k=1}^{\infty}$ sampled from the base distribution $H$ is a different dish in the menu, and the variable $\psi_{jt}$ represents the dish served at table $t$ in restaurant $j$. Note that each $X_{ji}$ is associated with one $\psi_{jt}$, whereas each $\psi_{jt}$ is associated with one $\phi_k$. The introduction of indicator factors has been studied, so that $t_{ji}$ represents the relationship between $X_{ji}$ and $\psi_{jt}$, whereas $k_{jt}$ represents the relationship between $\psi_{jt}$ and $\phi_k$. Therefore, in the Chinese restaurant franchise metaphor, customer $i$ in restaurant $j$ sits at table $t_j$ whereas table $t$ in restaurant $j$ serves dish $k_{jt}$.

At the same time, several variables need to be defined to indicate the number of tables and the number of customers. Marginal counts are represented with dots. We use the notation $n_{nj}$ to represent the number of customers at table $t$ in the restaurant $j$, and use $n_{njk}$ to represent the number of customers who enjoy the dish $k$ at table $t$ in the restaurant $j$, so $n_{njk}$ represents the number of customers who enjoy the dish $k$ at the table $t$ in restaurant $j$. The notation $m_{jk}$ denotes the number of tables in restaurant $j$ serving dish $k$. Thus, $m_j$ represents the number of tables in restaurant $j$, $m_k$ represents the number of tables serving dish $k$, and $m_-$ indicates the number of tables in which all existing customers are occupied in all restaurants.

According to the CRP model, after the A and B integrals are eliminated, they are respectively obtained [10]:

$$
X_{ji} \mid X_{j1}, \ldots, X_{ji-1}, \beta_0, G_0 \sim \sum_{i=1}^{m_j} \frac{n_{nj}}{1 + \beta_0} \delta_{\psi_{jt}} + \frac{\beta_0}{1 + \beta_0} G_0
$$

(2)

$$
\psi_{jt} \mid \psi_{j1}, \psi_{j2}, \ldots, \psi_{j1-1}, \gamma, H \sim \sum_{k=1}^{K} \frac{m_k}{m_+ \gamma} \delta_{\phi_k} + \frac{\gamma}{m_+ \gamma} H
$$

(3)

According to the above two formulas, if the customer occupies a table of an existing customer, then equations $X_{ji} = \psi_{jt}$ and $t_{ji} = t$ are established. If customer occupies a new table, then we increment $m_j$ by one, draw $\psi_{j_{mj}} \sim G_0$, and set $X_{ji} = \psi_{j_{mj}}$ and $t_{ji} = m_j$.

If the table selects a dish that has been ordered by other customer, then the equations $\psi_{jt} = \phi_k$ and $k_{jt} = k$ are established. If we choose a new dish, then we increase $K$ by one, draw $\phi_k \sim H$, and set $\psi_{jt} = \phi_k$ and $k_{jt} = k$.

HDP’s CRF construction is the process of assigning tables and dishes to customers. Suppose a restaurant represents a text. Every customer in the restaurant represents every word in the text. A dish represents a potential topic, and each table in the restaurant represents a group of customers who share the same dish. The customer $X_{ji}$ enters the restaurant, chooses the table $i$ to sit down, and enjoys the dish $k$ with other customers at the same table, which represents the process in which a word in the text to be split is assigned to the topic $ID_k$ [9].
3. Text Segmentation Method Based on HDP Model

3.1. Method to represent words with topic IDs

The method for using information gained by topic models is conceptually simple: Instead of using words directly as features to characterize textual units, we use their topic IDs as assigned by Bayesian inference. HDP inference assigns a topic ID to each word in the test document in each inference iteration step, based on a TM trained on a training corpus. The first series of experiments use the topic IDs assigned to each word in the last inference iteration. Figure 2 depicts the general setup.

The general flow of text segmentation based on the topic model is shown in Figure 2. First, preprocessing steps like tokenizing, sentence segmentation, part-of-speech tagging or filtering are applied to the training and test documents. The training data used to estimate the topic models should ideally be from the same domain as the test documents. The topic model is pre-estimated once and can then be used for inference on the test documents: HDP inference assigns a topic ID to each word in the test document and generates a document topic distribution.

![Figure 2: Basic concept of text segmentation using Topic Models](image)

An example of a text annotated with topic IDs is presented as follow:

He:2 needs:24 to:45 decide:24 if:24 he:24 wants:13 to:32 play:24 for:13 Manchester:24 United:24 often:32 ;:2 or:24 if:24 he:25 can:54 play:24 the:13 entire:24 team:24 in:18 another:36 team:2: ..2 Seeing:37 young:24 players:24 like:37 Sancho:24 can:13 shine:34 in:18 Dortmund:24 ..2

The:12 Chinese:53 people:23 persist:53 in:2 emancipating:53 the:4 mind:5 :19 seeking:53 truth:53 from:39 facts:53, realizing:53 the:2 emancipation:53 of:2 the:2 mind:53 and:17 the:10 mutual:53 agitation:53 of:2 reform:53 and:35 opening:53 up:6, the:2 promotion:26 of:2 ideas:53 and:35 exploration:53 of:2 practice:53 ;:2 demonstrating:53 the:21 powerful:53 forces:53 led:26 by:2 thought:53 ..2

One can clearly see the boundary by looking at the most probable topic IDs. The first text is about World cup, having mostly topic ID 24 assigned to words. The second segment is about Reform and Opening. Most words of this segment are annotated with topic ID 53. Instead of statically assigning each word, the topic ID repeats the word and resamples the topic ID based on each document topic distribution and each topic word probability in the previous inference step. Through the above analysis, it is feasible to use the topic ID instead of the word to represent the text and perform text segmentation on this basis.

The traditional text segmentation algorithm uses vocabulary as a feature, and the document is regarded as a word vector composed of feature weights, which can be expressed as (4):

\[ s = (w_1, w_2, ..., w_n) \]

The \( w_i \) represents the weight of the vocabulary \( i \) in the text, generally taking the word frequency (TF) or Term frequency-inverse document frequency (TF-IDF), and \( n \) is the number of words. The text similarity calculation method based on word vector cannot measure the latent semantic relationship between features, but this correlation cannot be reflected in the vector space model, but can be reflected by the same topic ID under the theme model. To solve the problem, this paper uses topic vectors instead of word vectors to represent text in text segmentation. The form of the theme vector can be expressed as (5):

\[ s' = (t_1, t_2, ..., t_v) \]
The $t_j$ indicates the frequency at which the topic $ID_j$ appears in the text; $\nu$ represents the number of topics automatically generated by the HDP model.

3.2. Algorithm realization principle

The TextTiling algorithm does not use the original sentence in the document as the basic unit, but constructs a basic unit by constructing a sentence every 20 [3] words. Segmentation of text is achieved by calculating the similarity between two word blocks by cosine similarity. It can be seen from 2.1 that this method cannot measure the potential semantic relationship between features. Therefore, this paper proposes the topic vector obtained by using the HDP model as the representation of the sentence, and records the method as TTHDP. TTHDP splits the document into topic-sequences rather than sentences, each of which consists of w topic IDs. To calculate the similarity between two topic-sequences called sequence gap, TTHDP uses k topic-sequences called blocks on the left and right sides of the sequence gap. This parameter $k$ defines the so-called blocksize. The cosine similarity is applied to calculate a similarity score based on the topic frequency vectors of the adjacent blocks at each sequence gap. A value close to 1 indicates a high similarity among two blocks, and a value close to zero denotes a low similarity. Then for each sequence gap, a depth score $d_p$ is calculated for describing the sharpness of the gap given by (6):

$$d_p = \frac{1}{2(hl(p) - C_p + hr(p) - C_p)}$$  \hspace{1cm} (6)

The function $hl(p)$ returns the highest similarity score to the left of the sequence gap index $i$, which does not increase and $hr(p)$ returns the highest score on the right side. Then all local maximum locations are then searched based on the depth score.

In the next step, the obtained maximum scores are sorted. If the number of segments $n$ is given as an input parameter, the $n$ highest depth scores are used, otherwise only the cutoff function of the segment is applied when the depth score is greater than $\mu - \sigma/2$, where the mean $\mu$ and the standard deviation $\sigma$ are based on the overall calculation of the depth score. Since TTHDP calculates the depth on every topic-sequence using the highest gap, the could result in segmentation in the middle of the sentence. To avoid this, the final step is to ensure that the segmentation is at the nearest sentence boundary.

4. Test Results And Discussions

4.1. Data sets

In order to further verify the feasibility of the proposed method, this paper uses relevant evaluation indicators and uses Sogou classified news as experimental test corpus for experimental analysis.

In view of the lack of public data sets for Chinese text segmentation, reference to the experimental design of Shi Jing [8] and Zou Jian [13], using two data sets for experiments. The Sogou classified news was analyzed and used as an experimental test corpus. It mainly comes from Sohu News from June to July 2012, domestic, international, sports, social, entertainment and other channels, a total of 17,820 news texts.

(1) SogouC1 data set. A text segmentation data set was constructed by selecting 3000 papers from the data set and then constructing the text according to the Choi [4] data set format. For document generation, ten segments of 3-11 sentences each, taken from different documents, are combined forming one document. 400 documents consist of segments with a sentence length of 3-11 sentences and there are 100 documents each with sentence lengths of 3-5, 6-8 and 9-11. Specific information is shown in Table 1.

| Test corpus | T(3-11) | T(3-5) | T(6-8) | T(9-11) |
|-------------|---------|--------|--------|---------|
| Paragraph length | 3-11 | 3-5 | 6-8 | 9-11 |
| Number of texts | 400 | 100 | 100 | 100 |
(2) SogouC2 data set. In the remaining data sets, reference to the experimental design of A, etc., the data set is constructed by manually marking the topic boundaries. The data set contains 3 test sets, as shown in Table 2.

| ID | Number of texts | Number of topics | Number of sentences | Average length of split unit | Average number of topics in the text |
|----|-----------------|-----------------|---------------------|-----------------------------|-------------------------------------|
| 1  | 8               | 70              | 2000                | 30                          | 10                                  |
| 2  | 12              | 100             | 3000                | 30                          | 8                                   |
| 3  | 10              | 90              | 2600                | 30                          | 9                                   |

4.2. Evaluation

The performance of algorithm is evaluated using two measures, commonly used in the TS task: The $P_k$ [11] measure and the $W_{i\text{nd}o\text{wDiff}}$ (WD) [12] measure.

$P_k$ can be defined as:

$$P_k = P(\text{seg}) * P(\text{miss}) + (1 - P(\text{seg})) * P(\text{false alarm})$$

Where $P(\text{seg})$ denotes the probability that two sentences of distance $k$ belong to different subject fragments; $1 - P(\text{seg})$ indicates the probability that two sentences with distance $k$ belong to the same subject. In this experiment, $P(\text{seg}) = 0.5$ is set according to the setting in [13]. $P(\text{miss})$ is the probability that the result of the algorithm segmentation lacks a paragraph; $P(\text{false alarm})$ is the probability of adding a paragraph to the result of the algorithm segmentation.

$W_{i\text{nd}o\text{wDiff}}$ can be defined as:

$$W_{i\text{nd}o\text{wDiff}}(\text{ref}, \text{hyp}) = \frac{1}{N - k} \sum_{i=1}^{N-k} (|b(\text{ref}_i, \text{ref}_{i+k}) - b(\text{hyp}_i, \text{hyp}_{i+k})| > 0)$$

Where $\text{ref}$ represents the true segmentation of the document; $\text{hyp}$ represents algorithm segmentation; $b(i, j)$ represents the number of boundaries between the entire sentence $\text{sentence}_i$ and the entire sentence $\text{sentence}_j$; $N$ represents the number of whole sentences in the text; $k$ takes half of the average length of the segments in the real segmentation.

4.3. Experimental result

The data is pre-processed before the experiment, using regular expressions to remove noise such as the scripting language retained by these news data. At the same time, removing those that occur frequently, but does not help to segment the text, and then segmentation of the data set to obtain a word segmentation.

In order to facilitate comparison with relevant scholars' research, this paper selects three commonly used methods for comparative experiments, which are the TextTiling method proposed by Hearst [3], the C99 method proposed by Choi [4] and the TopicTiling method proposed by Ridel [5, 6, 7]. The above methods are respectively recorded as TT, C99, TopicTiling. The C99 method on the two data sets uses the parameter settings of the experiment in [7], and the TT method uses the parameter settings of the experiment in [4]. In order to verify the dependence of TopicTiling on the number of topics, in addition to using 25 and 50 topic numbers in the experiment, also adding an experiment with a topic number of 80. The results of the proposed method are recorded as TTHDP. The number of iterations of the HDP model in the experiment is 10000. Other parameters are applicable to the setting of Yee et al. [9].

The HDP model needs to train the semantic relationship of the data mining vocabulary in order to avoid the influence of introducing training data from the outside. We refer to the setting in [15] to use a 10-fold cross-validation method on the SogouC1 dataset, using 90% of the data for training and 10% of the data for testing until all test data is traversed. Due to the small amount of SogouC2 data, there is a
problem of too little training data by using the 10-fold cross-validation method, so only one text is used for testing at a time, and the rest is used for training until all test texts are traversed.

Experiments were performed on the Sogou C1 data set, and the results are shown in Table 3. It can be seen from the table that the segmentation result of the TTHDP algorithm is significantly superior to the classic TextTiling, C99 and TopicTiling algorithms.

In contrast, it is found that using the topic ID instead of the vocabulary and the word vector to represent the text for text segmentation can improve the effect of text segmentation. From the comparison between TTHDP and TopicTiling, TopicTiling performs better when the number of topics is set to 80 and 50, especially when the number of topics is set to 80. Because 80 is close to the optimal number of topics in the dataset, TopicTiling can better exploit the performance of its model. On the other hand, when the number of subjects of TopicTiling is set to 50 and 20, the error rate of segmentation increases continuously, which means that when the number of topics keeps deviating from the optimal number of topics, the result will continue to deteriorate. Therefore, this type of method is not suitable for processing data with unknown topic numbers.

Table 3. Experimental results on SougouC1 dataset

| Method | T3-11 | T3-5 | T6-8 | T9-11 |
|--------|-------|------|------|-------|
|        | $P_k$ | WD   | $P_k$ | WD   | $P_k$ | WD   | $P_k$ | WD   |
| TextTiling | 0.263 | 0.283 | 0.277 | 0.297 | 0.234 | 0.257 | 0.218 | 0.224 |
| C99    | 0.275 | 0.287 | 0.298 | 0.302 | 0.247 | 0.267 | 0.231 | 0.249 |
| TopicTiling($k = 80$) | 0.178 | 0.182 | 0.187 | 0.196 | 0.167 | 0.192 | 0.184 | 0.192 |
| TopicTiling($k = 50$) | 0.198 | 0.204 | 0.219 | 0.237 | 0.194 | 0.206 | 0.193 | 0.203 |
| TopicTiling($k = 20$) | 0.215 | 0.231 | 0.214 | 0.216 | 0.203 | 0.217 | 0.204 | 0.213 |
| TTHDP  | 0.187 | 0.223 | 0.198 | 0.214 | 0.213 | 0.254 | 0.191 | 0.196 |

Experiments were performed on the SogouC2 data set, and the results are shown in Table 4. It can be seen from Table 4 that the average error rate of TTHDP is lower than other methods, which indicates that the proposed method has stable performance when dealing with different data sets and good robustness to new data. In addition, the optimal result of TopicTiling is not in the experimental group with the number of subjects $k = 80$, which means that the optimal number of topics for different dataset topic models is different. The method that depends on the number of topics set will affect the result when the new data is processed because the number of topics is not suitable.

Table 4. Experimental results on SougouC2 dataset

| Method | 1 | 2 | 3 | Average |
|--------|---|---|---|---------|
|        | $P_k$ | WD | $P_k$ | WD | $P_k$ | WD | $P_k$ | WD | $P_k$ | WD |
| C99    | 0.213 | 0.221 | 0.203 | 0.276 | 0.200 | 0.312 | 0.221 | 0.242 |
| TopicTiling($k = 80$) | 0.178 | 0.183 | 0.181 | 0.192 | 0.169 | 0.175 | 0.176 | 0.183 |
| TopicTiling($k = 50$) | 0.184 | 0.192 | 0.193 | 0.205 | 0.198 | 0.219 | 0.191 | 0.205 |
| TopicTiling($k = 25$) | 0.203 | 0.214 | 0.201 | 0.211 | 0.213 | 0.229 | 0.205 | 0.218 |
| TTHDP  | 0.156 | 0.162 | 0.152 | 0.173 | 0.162 | 0.178 | 0.156 | 0.171 |

5. Conclusion
In this paper, the text segmentation method based on the topic model cannot automatically determine the number of topics, so this paper proposes a text segmentation method based on HDP model. The method firstly uses the HDP model to model the segmented text, obtains the representation of the text
under the theme model, and then uses the topic vector for the TextTiling segmentation algorithm to implement text segmentation. The results of experiments show that the proposed method can not only reduce the error segmentation rate of text, but also further simplify the calculation of traditional topic-based model segmentation algorithm, which greatly improves the efficiency of text segmentation. What needed in the future is to focus on improving the performance of the TTHDP algorithm and efficiently implement text segmentation.

Acknowledgments
This work was financially supported by the State Grid Corporation of China Technology Project (5211XT17000F)

References
[1] Wu J W, Tseng J C R, Tsai W N. A hybrid linear text segmentation algorithm using hierarchical agglomerative clustering and discrete particle swarm optimization[J]. Integrated Computer Aided Engineering, 2014, 21(1):35-46.
[2] Cardoso P C F, Taboada M, Pardo T A S. Subtopic Annotation in a Corpus of News Texts: Steps Towards Automatic Subtopic Segmentation[J]. Chemical Communications, 2014.
[3] Hearst M A. TextTiling: segmentation text into multi-paragraph subtopic passages[J]. MIT Press, 1997, 23(23): 33-64
[4] Choi F Y Y. Advances in domain independent linear text segmentation[J]. Proc. NAACL-00, 2000:26-33.
[5] Riedl M, Biemann C. How text segmentation algorithms gain from topic models[C]// Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Association for Computational Linguistics, 2012:553-557.
[6] Riedl M, Chris B. Text Segmentation with Topic Models[J]. Journal for Language technology and Computational Linguistics, 2012, 27(1):47-69
[7] Riedl M, Biemann C. TopicTiling: a text segmentation algorithm based on LDA[C]. //ACL 2012 Student Research Workshop. Association for Computational Linguistics, 2012 :37-42
[8] Jing S, Guozhong D. Text Segmentation Based on PLSA Model[J]. Journal of Computer Research & Development, 2007, 44(2):242-248.
[9] Yee WT, Michael IJ, Matthew JB, et al. Hierarchical Dirichlet Processes[J]. Journal of the American Statistical Association, 2006, 101(476):1566-1581
[10] Zhimin Z, Shenyong G. A SURVEY ON HIERARCHICAL DIRICHLET PROCESS PRINCIPLE AND ITS APPLICATIONS[J]. Computer Applications & Software, 2014.
[11] Beeferman D, Berger A, Lafferty J. Statistical Models for Text Segmentation[M]. Kluwer Academic Publishers, 1999.
[12] Pevzner L, Hearst M A. A critique and improvement of an evaluation metric for text segmentation[M]. MIT Press, 2002, 28(1):19-36
[13] Jian Z, Mao-Sheng Z, Li M. Method of chinese text segmentation model acquisition and its optimization[J]. Journal of Nanchang University, 2011.
[14] Yao L, Yuan-Hua S, Xing-Wei G, et al. Study on Text Segmentation Based on Domain Ontology[J]. Computer Science, 2018.