A Novel Customized FARS Recommendation Algorithm based on Search

To cite this article: L Yao et al 2018 IOP Conf. Ser.: Mater. Sci. Eng. 435 012035

View the article online for updates and enhancements.
A Novel Customized FARS Recommendation Algorithm based on Search

Yao L \(^{a}\), Yin Q Q \(^{a}\), Zhong Y \(^{b}\)

\(^{a}\)College of Mathematics and Econometrics, Hunan University, Changsha 410082, China.

\(^{b}\)College of Information Science and Engineering, Hunan University, Changsha 410082, China.

*E-mail: yao@hnu.edu.cn

Abstract: Search engines and recommendation systems work in different ways, but they are designed to handle the same problem: information overload. This paper combines the advantages of the two and proposes a new search-based recommendation algorithm: focused analysis recommendation system (FARS). Inspired by LDA, the FARS model was designed as a three-layer (user-interest-item) Bayesian structure. Based on user-specified keywords, FARS can only perform modeling analysis on relevant users, and return the distribution of interest of these users, which are estimated by the Gibbs sampling method. We treat the top-ranking items in each interest vector as related, and then sort them according to the probability that an item is selected by this user group. Finally, a specified number of recommended items is returned based on the above sorting result. The experimental results show that the proposed model can outperform on Precision@n in comparison with the other baseline models. To some extent, it provides some inspiration for combining search and personalized recommendations.

1. Introduction

With the emergence of Internet, the problem of information overload\(^{[1]}\) has become increasingly fierce. In order to enable users to get the information they need faster and better, the corresponding solution has been upgraded from the search engine to the recommendation system. Search engine is an information search tool that allows individuals with clear goals to search for information they need; Recommendation system (usually known as personalized recommendation system) is a kind of information filtering tool, by analyzing the user's historical behavior data to model the user's interest, actively provides users with information that interests them (note that the user does not need to have a clear goal in this process).

Both search engine and recommendation system are tools to help users get useful information, but in some cases, the results returned by these two methods may not be satisfactory to the user. In practice we found that the user are interested in a type of movie that he has never watched before and wanted to learn more about it. The user needs mentioned above, however, are not likely to excavate from his behavior records. Moreover, the evaluation criteria of the recommendation system include coverage and diversity, so the items they cover are often too extensive and may not even touch the keywords provided by users. Search engine will only return relevant introductions of this movie, video resources, movie review information, and the like.
To solve the above problem, we have carefully analyzed the performance in automatically mining user interests of recommendation system and the focused analysis function of search engine. Combining the advantages of two methods, this paper proposes a new search-based recommendation algorithm: focused analysis recommendation system (FARS). The proposed algorithm can return a collection of recommended items by accurate analysis of user data that contains or relate with keywords.

The rest of the paper is organized as follows: Section 2 describes related research work; Section 3 is about the proposed model and theoretical basis; Section 4 is the experimental results; and the end of the article is related references.

2. Background and related work

2.1. Recommendation system

Recommendation system is one of the hot areas of research and is also popular in industry in recent years. As a practical tool, it has been used in many different e-commerce and social networks[2]. There are many types of data that can be analyzed, but we mainly study the collaborative filtering algorithm[3] which makes recommendations by modeling user behavior data. This recommendation algorithm can be divided into: neighborhood-based[4,5] and model-based[6,7].

Neighborhood-based algorithm is recommend to the user based on the similarity measure results. It can be grouped into two general classes: user-based collaborative filtering algorithms (userCF) and item-based collaborative filtering algorithms (itemCF).

- userCF groups users based on similarity, and members in the same user group are, to a large extent, interested in similar items. So this method will recommend to users similar items that users prefer[4].
- itemCF is grouping items based on the assumption that two items that many people like at the same time are likely to belong to the same class. When the user is interested in an item, the method will recommend the most similar/related items to the user[5].

Model-based algorithm uses user behavior data to learn a model, and then use it to predict user behavior. Like LDA-based recommendation algorithm[7,8] attempts to construct a recommendation model by treated user interest as an implicit feature, which is regarded as the key to associating users with the items they are interested in. This algorithm not only has a good theoretical basis, but also can obtain more flexible recommendation results simply by changing the number of latent features in the algorithm.

2.2. LDA-based Algorithm

LDA (Latent Dirichlet Allocation) is a probabilistic generation model[9]. This model can mine the internal correlation of vocabulary in the corpus, find the hidden semantics of the document, and promote the topic discovery of the text. The advantages above make LDA widely used in document analysis, classification, or clustering[10].

In 2005, LDA was originally introduced into the recommendation system. It was used to model the implicit interrelationships between items, and generate dynamic recommendations for users[7]. Since then, LDA has become a popular optimization tool in the recommendation system. It is used to explore commonalities in data, like user-generated content[11-13](eg, tags, reviews) and the item information[14,15](eg, text Themes, movie categories).

Until 2011, LDA was used to mine user interest, and construct a social media user recommendation system that recommends to a user new friends having similar interests[8]. Later, Chang T M et al. used LDA to model user data and group them, to make better recommendations for a specific user group[16]. In addition, LDA is used to explore the user's potential interests for generating a personalized recommendation of specific user[17].

Rather than simply recommendation based on a single algorithm, some recommendation systems use a hybrid approach by combining LDA-based recommendation algorithm and other. For example:
LDA is used to model the user and the item simultaneously, find out their neighborhood sets respectively, and then recommend according to their relevance[18]; In order to obtain a better recommendation result, some scholars to combine LDA-based recommendation algorithm with matrix decomposition[19].

However, an over-specialization recommendation system makes it difficult for users to customize recommendations according to his or her needs[2]. Similar to other recommendation algorithms, LDA-based recommendation algorithm and its variants cannot return a personalized recommendation list for a single user or user group with the same interests. For the aforementioned example of that moviegoer, this user will never receive a recommendation that he or she have not seen, even if it is the most anticipated movie for user. In order to solve this problem, the search-based related research is considered in the proposed algorithm. The following is a detailed introduction to related work in search field.

2.3. Personalized Search
Search results that include only keywords are not what we need. So this section will make a introduction to some of the personalized search related to the work of this article. For movie websites, there may be multiple reasons why users be interested in a movie, such as a certain director, actor, or the theme of the movie. In order to return better search results, personalized search needs to figure out the user’s true intentions.

Concept-Based retrieval method[20] can better respond to personalized needs in real-time, by embeds feature selection methods into the search process. Lee N et al. proposed a robot-based recommendation service[21] that can generate flexible responses to different language expressions to achieve real-time human-computer interaction recommendations. According to the user's search history, determine the user's true potential intention by LDA, Yan R et al. sorts related document sets according to KL distances, and returns the personalized search document list to the user[22]. Wu J T et al. expands the keyword-related training text sets by incrementally expanding the seed words, and determines keywords in these text sets corresponding to the query words. This method can improve the accuracy of topic clustering and topic diversity[23]. Although these methods achieve a certain degree of personalization, they are not enough to solve the problems mentioned above.

Therefore, we proposes a new algorithm that makes use of the complementarity of search engines and recommendation systems. This algorithm can perform focused analysis on certain relevant aspects according to the user-specified keywords, and return an appropriate recommendation list that is what the user needs. For the case mentioned in the example, we need only to set the appropriate number of interest categories, the proposed algorithm can return a suitable set of recommended items related to this movie.

3. FARS Model
As discussed in the introduction section, our problem statement is that given a collection C of user data, the proposed model can return a recommended list of the relevant aspect specified by the user using a set of keywords S.

![Figure 1: The graphical model of FARS](image)
Inspired by the idea [24] that the items that a user is exposed to is a latent variable that may change for various user/item combinations, we treat the state that whether a user is related to a keyword as a potential random variable. Following previous work, we assume that each user's behavior is a probabilistic event. In other words, the process of a user select an item is actually randomly selecting a point of interest, and then draw an item randomly from the corresponding interest. The element definitions of the FARS model given in Table 1, and the graphical model of FARS is given in Figure 1. For a given keyword, it is easy to determine whether a user data contains the keyword. This state is represented by the variable \( x \): where \( x = 1 \) means that a user's data contains the keyword, otherwise \( x = 0 \). The potential random variable that is used to represent the user data and keyword related state is represented by \( r \): where \( r = 1 \) means the user data is relevant to the target and \( r = 0 \) is irrelevant.

Table 1: Element Definitions of the FARS Model

| Element | Definition |
|---------|------------|
| \( T \)  | the number of interest |
| \( M \)  | the number of user |
| \( N_m \) | the number of items in the \( m \)-th user's data |
| \( V \)  | the number of different items |
| \( r_m \) | the number of the relevance status |
| \( m, z, v \) | user data, interest, item |
| \( w_i, z_i \) | item in position \( i \) (item \( i \)), interest of item \( i \) |
| \( w, z, r \) | all items, assigned interest and relevance status |
| \( x, r \) | keyword indicator, relevance status |
| \( x_m, r_m \) | keyword indicator, relevance status of the \( m \)-th user's data |
| \( \alpha, \beta^r, \beta^s \) | Dirichlet prior for \( \theta, \phi^r, \phi^s \) |
| \( \gamma \) | Beta prior for \( \pi \) |
| \( \theta \) | multinomial distribution over interest |
| \( \pi \) | bernoulli distribution over relevance status |
| \( \phi^r \) | multinomial distribution over interest-item |
| \( \phi^s \) | multinomial distribution over irrelevance interest-item |
| \( z^{(\cdot)} \) | all assigned interest except the item \( i \) |
| \( \beta^{(\cdot)} \) | all selected items except \( v \) under relevance status \( r \) |
| \( \beta^{(\cdot)}_t \) | all selected items except \( v \) under relevance status \( r \) and interest \( t \) |
| \( N_{r,m,v} \) | the frequency of item \( v \) in the \( m \)-th user's data under relevance status \( r \) |
| \( N_{r,m,v}^{(\cdot)} \) | the number of user's data under relevance status \( r \) except the \( m \)-th user |
| \( N_{r,m,v}^{RW} \) | the number of item \( v \) under relevance status \( r \) |
| \( N_{r,m,v}^{RW}^{(-i)} \) | the number of items under relevance status \( r \) and interest \( t \) in the \( m \)-th user's data except item \( i \) |
| \( N_{r,m,v}^{RW}^{(+i)} \) | the number of items \( v \) under relevance status \( r \) and interest \( i \) |

3.1. Generative Process

Then when the keyword is given, each data can be identified as relevant or irrelevant to target. Then for each user \( m \in \{1,2,...,M\} \):

1. Draw a prior distribution \( \pi \sim \text{Beta}(\gamma) \);
2. Draw relevance status \( r \) based on keyword indicator \( x \) and Bernoulli(\( \pi_m \));
3. If the document is relevant to the targeted aspect, that is \( r = 1 \):
   a. Draw \( \theta \sim \text{Dirichlet}(\alpha) \) as a user - interest distribution;
   b. Draw a interest \( z \sim \text{Multinational}(\theta^t) \);
(c) Draw a item distribution $\varphi^i \sim \text{Dirichlet}(\beta^i)$;

(d) Emit a item $w_i \sim \text{Multinomial}(\varphi^i)$;

(4) In contrast the document is irrelevant to the targeted aspect, if $r = 0$:

(a) Draw $\varphi^w \sim \text{Dirichlet}(\beta^w)$ as a item distribution of a irrelevant interest to the keyword;

(b) Emit a item $w_i \sim \text{Multinomial}(\varphi^w)$.

As shown above, there are two related states for each user. When $r = 1$, it means that the user is related, so we need to add this user's behavior data to the candidate data set for analysis. The analysis process is similar to the topic modeling of LDA: a topic $z$ is chosen from $\varphi^i$, after that a item $w_i$ is emitted from the selected topic by $\varphi^i$. When $r = 0$, that is to say this user data is not related to the specified aspect. In this case, we only draw $\varphi^w \sim \text{Dirichlet}(\beta^w)$ as a item distribution of a irrelevant interest to the keyword, and a item $w_i$ is emitted from $\varphi^w$. In other words, the items in irrelevant user are drawn from only one (irrelevant) interest.

The generative process of FARS above can be used to explain the relationship among user data, interests, and items. However, in order to calculate the probability that a user is interested in an item, we also need to compute the user-interest distribution $\theta$ and the interest-item distribution $\phi$ based on the given Dirichlet parameters $\alpha$ and $\beta$.

3.2. Parameter Estimation

Since random variables $\theta$ and $\phi$ are not easily found directly, they are generally estimated using an approximate methods, like variational reasoning EM (expectation maximization)[9] and Gibbs sampling[25]. The former is computationally efficient, but the derivation is complicated; On the contrary, the latter has low computational efficiency, but it is easy to implement, and more tolerant of local optima. Therefore, we choose Gibbs sampling algorithm to estimate $\theta$ and $\phi$ that are polynomial distribution parameters.

First, we sample the relevance status $r$ for user $m$, by the following conditional distribution:

$$P(r_m = c|x_m = d, \varphi^i, \pi, \varphi^w, \beta^i, \beta^w) \propto \begin{cases} 
\frac{N_{R_{m,c}}^{i} + \gamma}{N_{R_{m,c}}^{i} + |R| \gamma} \prod_{r = 1}^{|R|} \Gamma(\beta^i_{r} N_{c,r}^{w} + \beta^i_{r} + N_{c,m}^{w}) , & c = 1 \\
\frac{N_{R_{m,c}}^{i} + \gamma}{N_{R_{m,c}}^{i} + |R| \gamma} \prod_{r = 1}^{|R|} \Gamma(\beta^w_{r} N_{c,r}^{w} + \beta^w_{r} + N_{c,m}^{w}) , & d = c = 0 \\
0 , & \text{otherwise}
\end{cases}$$

(1)

Next, we sample an interest of the item in position $i$ is:

$$P(z_i = t|r^{*}, r, \varphi^i, \beta^i, \beta^w) \propto \begin{cases} 
\frac{N_{RMT_{i,t}^{m,d}}^{i} \alpha + 1}{N_{RMT_{i,t}^{m,d} + |T| \alpha} \times \prod_{r = 1}^{T} \Gamma(\beta^i_{r} N_{i,j,r}^{RTW_{t}^{m,d}} + \beta^i_{r} + N_{i,j,r}^{RTW_{t}^{m,d}})} , & r_i = 1 \\
\frac{N_{RMT_{i,t}^{m,d}}^{i} \alpha + 1}{N_{RMT_{i,t}^{m,d} + |T| \alpha} \times \prod_{r = 1}^{T} \Gamma(\beta^w_{r} N_{i,j,r}^{RTW_{t}^{m,d}} + \beta^w_{r} + N_{i,j,r}^{RTW_{t}^{m,d}})} , & r_i = 0
\end{cases}$$

(2)

The user-interest distribution $\theta$ and the interest-item distribution $\phi$ of related user groups as following, and the procedure of learning FARS by Gibbs sampling show as Figure 2.
\[
\begin{align*}
\theta &= \frac{N_{t,m,d}^{RMT(i)} + \alpha}{N_{t,m,d}^{RMT(i)} + |T| \alpha} \\
\varphi &= \frac{N_{t,m,v}^{RTW(i)} + \beta_{d,v}}{N_{t,m,v}^{RTW(i)} + |V| \beta_{d,v}}
\end{align*}
\]

(3)

Figure 2: The procedure of learning FARS by Gibbs sampling

The process diagram of FARS is shown in Figure 3, and the textual description of FARS is described as follow:
1. For a given keyword, the proposed algorithm will determine the user's relevance;
2. Expand the keyword set to better express the interest classification related to the keywords, by use the commonality of relevant users;
3. Sample the relevant user using the Gibbs sampling formula related to the keywords, to find more relevant user sets;
4. Reduce words that are not representative enough in the keyword set, add some better words to it, and iterate until convergence;
5. According to the sampling results, estimate the user-interest distribution \( \theta \) and the interest-item distribution \( \varphi \);
6. Sort related items by the product of \( \theta \) and \( \varphi \) and return a list of recommendations.

Figure 3: The process diagram of FARS

4. Results and Evaluation

Data and Keyword: MovieLens dataset is used in our experiments, these data were created by 671 users between January 09, 1995 and October 16, 2016. Users were selected at random for inclusion. All selected users had rated at least 20 movies. Instead of modeling with tags or contexts, our method takes only the user data and itemID to construct a recommendation model. Therefore we removed other information in the data set, such as: rating, time stamp, and so on. We will choose from several films to be analyzed as keywords, some of them are more frequent, like Titanic (1997) (movieID is 1721); some are infrequent, like Safe (1994) (movieID is 190).

4.1. Experimental Setting
Parameter Setting: For the hyper-parameter setting, we place: $\gamma=1$, $\alpha=1$, $\beta^n=0.001$; and $\beta^*\{0,1\}$ (for a keyword $s \in S$, the $\beta^*_s=1$). The models for comparison are also with the same setting.

Baseline Models for Comparison: To evaluate our proposed FARS, we compare it with the following baseline models.

Model1: We use LDA for user interests generation via analyze all user data[17], and inspection to all resulting interest class from LDA, to find the subset of relevant items. Return a set of recommended items by their probability.

Model2: Group users according to conditions that the user data includes keywords, and using LDA to model interest in relevant user groups. Return the most popular movies in this user group.

Model3: We runs TF-IDF only on the user data that contains one or more keywords from S, and uses this result to expand the keyword to obtain the keyword set $S'$. Return the most popular ten movies in the relevant user group[21] just like Model2.

Since LDA is a full-analysis model, it analyzes all user data and does not directly generate keyword-related interest categories and item recommendations. Therefore, in the experiment, we set the number of interest categories $T$ of Model1 to 15 or 20, find the appropriate item from its related subset, and return a recommendation list according to the sort of probability; In other methods, the value of interest classification $T$ is set to 5-10. Since we do not have a priori information about the number of user interests, the size of $T$ is determined based on the popularity of the movie being searched (eg, popular movies such as Titanic may be seen in most user data). In general, $T$ is set to 5 for infrequent targets, and $T$ is set to 10 for other more frequent targets. Note that we will use the same $T$ value to compare the results given by these methods.

4.2. Experimental Results and Evaluation

According to the characteristics of top-k recommendation, Precision@n (or P@n for short) were used to assess the recommendation’s effect in the experiment. P@n can give the precision results at different rank position n. Specifically, we use a normalized form of precision that can estimate the proportion of related items in the recommendation list.

$$P_{(i)@n} = \frac{N_{(i)\text{(relevanted}@n)}}{N\text{(items}@n)} \quad (4)$$

In Equation 4, $P_{(i)@n}$ indicates the Precision@n for model(i). $N_{(i)\text{(relevanted}@n)}$ is the number of relevant items in the recommendation list returned by model(i). $N\text{(items}@n)$ is the length of the recommended list. The average performances of different models about P@n at the rank position of 5 and 10 are displayed in Figure 4 and Figure 5.

![Figure 4: P@5](image)

![Figure 5: P@10](image)
As can be seen from the picture, both P@n at the rank position of 5 or 10, and in terms of infrequent or frequent, our proposed FARS model is significantly outperforms other models. Compared to other models, FARS can identify user data related to the keyword through a random variable. This variable can continuously iterate while modeling user interest, and optimize the model results, which is also the main reasons for the higher scores.

Among them, Model3 obtains the second best scores. Because Model3 can use TF-IDF for keyword expansion, and analyze the subset of user data associated with the expanded set of keywords. Model3 can find more or better interest for the relevant aspect than Model2, because Model2 discards many relevant user data, resulting in information loss. Model2 can rules out the irrelevant user data that can interfere with good topic identification. But then it also loses relevant information as we discussed above. Thus, which results in it has better performance compared to model1, and inferior than Model3.

Compared with FARS and Model3, the experimental results of the previous two models in infrequent aspect are worse. Because there are few user data containing the keywords about infrequent aspect, the amount of information contained in these data is insufficient to illustrate the true interest of such user groups; That is to say, FARS and Model3, to a certain extent, extend the set of user-specified keywords, which makes more data for analysis. The worst performance is model1. Because, instead of directly analyze the user's interest for the target aspect, Model1 is required to conduct modeling analysis on all raw data. When it comes to some infrequent movie, this model returned user interest are often too coarse and may not even be on target.

By comparing P@5 and P@10, it is easy to find that we proposed FARS recommends related items ranked relatively high. That is, the algorithm can better identify related items and recommend them to users more preferentially. Although Model2 and Model3 scored similarly in the frequent aspect, the difference was that Model3 had a slightly higher P@5 score than P@10, indicating that the expanded keyword set could have a certain impact on the ranking of related items.

5. Conclusion
In this paper, we studied the problem of search-based customize recommendation, and propose FARS model. Instead of analyzing user interest based on all data, FARS combines user requirements (search based on keywords) and recommendations (analysis of preferences of related user groups). Compared with the baseline model, our proposed model can not only return more relevant recommendation items, but also return related item rankings more forward.

Although the proposed model performs well, there are still many other optimizations that can be done: visualization, interactivity, interpretability, and so on. More specifically. Visualization: Shows users all possible results in an intuitive way, and the proportion of various results. Interactivity: If a user is interested in a certain type of item in the recommendation list, he can perform deeper or shallower analysis on such items without having to re-enter other keywords. Interpretative: Make some explanation for the recommendation result so that the user can judge whether the item is worth choosing according to his own needs.

Acknowledgment
This research was financially supported by the National Natural Science Foundation of China under Grant No. 61502159.

References
[1] Yang C C, Chen H and Hong K 2003 Visualization of large category map for internet browsing. Decision Support Systems, 35(1), 89-102.
[2] Adomavicius G, Tuzhilin A 2005 Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. IEEE Transactions on Knowledge & Data Engineering, 176, pp.734-749.
Collaborative filtering recommender systems. The adaptive web Vol.22, pp.291-324. Springer-Verlag.

[4] Wang J, Vries Arjen P D, Reinders Marcel J T 2006. Unifying user-based and item-based collaborative filtering approaches by similarity fusion. Proc. of, ACM SIGIR Conference on Information Retrieval pp.501-508.

[5] Sarwar B, Karypis G, Konstan J and Riedl J 2001 Item-based collaborative filtering recommendation algorithms. International Conference on World Wide Web Vol.4, pp.285-295. ACM.

[6] Jenatton R, Roux N L, Bordes A and Obozinski G 2012 A latent factor model for highly multi-relational data. International Conference on Neural Information Processing Systems Vol.4, pp.3167-3175. Curran Associates Inc.

[7] Jin X, Zhou Y, Mobasher B 2005 A maximum entropy web recommendation system: combining collaborative and content features. Eleventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Chicago, Illinois, USA, August pp.612-617. DBLP.

[8] Pennacchiotti M, Gurumurthy S 2011 Investigating topic models for social media user recommendation. International Conference on World Wide Web, WWW 2011, Hyderabad, India, March 28 - April pp.101-102. DBLP.

[9] Blei D M, Ng A Y, Jordan M I 2003 Latent dirichlet allocation. J Machine Learning Research Archive Vol.3, pp.993-1022.

[10] Blei D 2011 Probabilistic topic models. ACM SIGKDD International Conference Tutorials Vol.27, pp.1-1. ACM.

[11] Xu Y, Chen Z, Yin J, Wu Z and Yao T 2015 Learning to Recommend with User Generated Content. WAIM Vol.9098, pp.108-113.

[12] Krestel R, Fankhauser P, Nejdl W 2009 Latent dirichlet allocation for tag recommendation. ACM Conference on Recommender Systems, Recsys 2009, New York, NY, USA, October pp.61-68. DBLP.

[13] Zhai C, Pang J 2017 Mining Latent Features from Reviews and Ratings for Item Recommendation. International Conference on Computational Science and Computational Intelligence pp.1119-1125. IEEE.

[14] Chang T M, Hsiao W F 2013 Lda-based personalized document recommendation.

[15] Wang S, Wen W D 2017 A hybrid recommendation method based on features. Computer & Digital Engineering.

[16] Chang T M, Hsiao W F, Lin Y L 2014. LDA-based group document recommendation.

[17] Xie W B, Dong Q, Gao H 2014 A probabilistic recommendation method inspired by latent dirichlet allocation model. Mathematical Problems in Engineering, 201424.

[18] Gao M, Jin C Q, Qian W N, Wang X L and Zhou A Y 2014 Real-time and personalized recommendation on microblogging systems. Chinese Journal of Computers.

[19] Cao Y, Li W, Zheng D 2018 A hybrid recommendation approach using lda and probabilistic matrix factorization. Cluster Computing 1, 1-11.

[20] Egozi O, Markovitch S, Gabrilovich E 2011 Concept-based information retrieval using explicit semantic analysis. Acm Transactions on Information Systems, 292, 1-34.

[21] Lee N, Kim E, Kwon O 2017 Combining tf-idf and lda to generate flexible communication for recommendation services by a humanoid robot. Multimedia Tools & Applications, 774, 1-16.

[22] Yan R, Li S j 2018 Document Retrieval Algorithm Based on Query Intent Identification and Topic Modeling. Computer Engineering, 443: 189-194.

[23] Wu J T, Li T N 2017 Keyword recommendation method and system based on latent dirichlet allocation model. WO 2017114019 A1.

[24] Liang D, Charlin L, Mcinerney J and Blei D M 2015 Modeling user exposure in recommendation. 951-961.

[25] Griffiths T L, Steyvers M 2004 Finding scientific topics. Proc Natl Acad Sci USA, 101 Suppl 11, 5228-5235.