IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received March 5, 2018, accepted April 1, 2018, date of publication April 18, 2018, date of current version May 24, 2018.

*Digital Object Identifier 10.1109/ACCESS.2018.2828081*

# Relevant Feedback Based Accurate and Intelligent Retrieval on Capturing User Intention for Personalized Websites

YAYUAN TANG[1,2], HAO WANG[3], KEHUA GUO[2,4], YIZHE XIAO[2], AND TAO CHI[5]

[1]School of Electronics and Information Engineering, Hunan University of Science and Engineering, Yongzhou 425199, China
[2]School of Information Science and Engineering, Central South University, Changsha 410083, China
[3]Faculty of Engineering and Natural Sciences, Norwegian University of Science and Technology, N-6025 Ålesund, Norway
[4]Key Laboratory of Information Processing and Intelligent Control, Minjiang University, Fuzhou 350108, China
[5]Key Laboratory of Fisheries Information,Ministry of Agriculture, Shanghai Ocean University, Shanghai 200120, China

Corresponding author: Kehua Guo (guokehua@csu.edu.cn)

This work was supported in part by the Hunan Science and Technology Plan under Grant 2012RS4054, in part by the Natural Science Foundation of China under Grant 61672535 and Grant 61472005, in part by the Key Laboratory of Intelligent Perception and Systems for High-Dimensional Information of the Ministry of Education Innovation Fund under Grant JYB201502, in part by the Key Laboratory of Information Processing and Intelligent Control of the Fujian Innovation Fund under Grant MJUKF201735, in part by the Natural Science Foundation of Hunan Province under Grant 2018JJ3203, in part by the Graduate Student Innovation Project of Hunan Province under Grant CX2016B049, in part by the Research Foundation of Education Bureau of Hunan Province under Grant 17C0679, in part by the Hunan University of Science and Engineering Research Project under Grant 17XKY071, and in part by the key discipline for computer application and technology of the Hunan University of Science and Engineering.

**ABSTRACT** With the rapid growth of networking, cyber–physical–social systems (CPSSs) provide vast amounts of information. Aimed at the huge and complex data provided by networking, obtaining valuable information to meet precise search needs when capturing user intention has become a major challenge, especially in personalized websites. General search engines face difficulties in addressing the challenges brought by this exploding amount of information. In this paper, we use real-time location and relevant feedback technology to design and implement an efficient, configurable, and intelligent retrieval framework for personalized websites in CPSSs. To improve the retrieval results, this paper also proposes a strategy of implicit relevant feedback based on click-through data analysis, which can obtain the relationship between the user query conditions and retrieval results. Finally, this paper designs a personalized PageRank algorithm including modified parameters to improve the ranking quality of the retrieval results using the relevant feedback from other users in the interest group. Experiments illustrate that the proposed accurate and intelligent retrieval framework improves the user experience.

**INDEX TERMS** Intelligent retrieval, real-time location, personalized websites, keywords extraction, implicit feedback.

## I. INTRODUCTION

Cyber-physical-social systems (CPSSs), including the cyber world, physical world, and social world, can provide high-quality personalized services for humans [1], [2]. Such systems take data from the environment, integrate the data and extract valid information. CPSSs take human society from the abstract of philosophy into daily concrete applications. Human flesh search, new media, Wiki and crowd-sourcing mechanisms rapidly enhance human living space and are both data-driven and virtualized. Knowledge can almost be transmitted and accessed at the speed of light through cyberspace via social networks. The emergence of cyber-physical-social systems (CPSSs) [3] has resulted in the explosive growth of networking information. They however also bring about several important challenges in CPSSs.

To address the explosive growth of Internet information, the development of effective retrieval techniques is urgent. Various retrieval approaches have been extensively employed to retrieve massive amounts of Internet information. For example, people can use search engines to conveniently crawl information from the Internet such as through Google and Baidu. However, the retrieval results often contain substantial amounts of unnecessary information, and some required results can be hidden in the back of a webpage; thus, users have to spend a lot of time finding the relevant results. It remains difficult to retrieve more accurate and more special information that satisfies the query intentions of users. Thus, a a special field, retrieval information in personalized websites aims to better account for an individual's requirements than do general search engines [4], [5].

A number of approaches to retrieving information in personalized websites have been presented. The dominant approach primarily focuses on keywords-based techniques. However, a search engine only retrieves information based on keywordss provided by the user and is not in a position to capture the user's search intention. Different users have different search needs on account of their different ages, interests and occupations. For example, the keywords 'orange' could mean a type of fruit or a color. If the search engine process the keywordss in the same way and return the retrieval results to different users, it cannot still change the fact that search engines lack the ability to satisfy the personalized demands of users.

Many approaches have been taken to capture the intentions of users to solve the above problems. The most commonly used approach is to employ keywords-based searching methods to find relevant webpages and provide appropriate ranking strategies [6]. In addition, with the increase in the demands on user satisfaction, vertical search engines have provided certain value information and related services for a particular field, a particular person and a particular demand (e.g., travel searches and educational resource searches) [7]. However, detailed and accurate information is still not able to be obtained by vertical or general search engines. For instance, if we are at Central South University, we need to search today's news, find the location of the fifth canteen, etc. A general search engine will not provide satisfactory results.

Moreover, improving rankings has not been effectively addressed. Personalized search has become a research direction for numerous scientific researchers. User-behavior-based techniques have boosted the ranking performance [8]. For example, click models have been well studied for personalized search [9], where clicks with a reasonable dwell time on a particular document suggest that a user favors this result [10], [11], whereas it might be non-relevant for other users. In this paper, we provide personalized ranking based on user behavior to meet their real-time information needs. During information retrieval, users usually expect to obtain the unknown information. Analyzing the user's historic search data represents a large proportion of research on personalized retrieval but is still unable to solve existing problems.

In this paper, we investigate the above-mentioned problems. The main contributions of this paper are summarized as follows:

1) We propose an accurate and intelligent retrieval framework with real-time location and relevant feedback technology for personalized websites.

2) We predict user retrieval intentions by analyzing the user's real-time location to determine a personalized search range. To improve the retrieval results, we also propose a strategy of implicit relevant feedback based on click-through data analysis, which can obtain the relationship between the user query conditions and the retrieval results.

3) We design a personalized PageRank algorithm including modified parameters to improve the ranking quality of the search results using the relevant feedback from other users in the interest group. The solution ensures that different users obtain different results that are closer to the user's requirements, even with the same keywords search.

The remainder of this paper is organized as follows. Section 2 provides a brief review of related work and comparison with similar problems. In section 3, the model of the search engine is introduced. This section also describes the optimized retrieval strategy as the solution to the problem. In section 4, the proposed framework is implemented, and some experiments are performed, analyzed and compared with other methods through simulation. Finally, the paper is concluded in section 5, and future work is reviewed.

## II. RELATED WORK

Researchers have made great effort to improve the efficiency of information retrieval. The most common approach is based on keywordss. Substantial current research work only considers single keywordss without fully expressing the intentions of users. In expanded research, others use related multi-keywords queries, which make the query results more consistent with the user's requirements [6]. Traditional keywords extraction algorithms come in four types: the LCS algorithm [12], N-Gram algorithm [13], IkAnalyzer algorithm [14] and Nakatsu algorithm. Furthermore, a few algorithms provide typical ranking algorithms for matching results in the search procedure. However, the aforementioned retrieval approaches suffer from several drawbacks. The approach produces many unrelated results, which lead to a massive waste of computational resources.

To solve the above problem, a new generation of search engines is becoming a hot spot of research at home and abroad. Chakrabarti *et al.* [15] implemented a free custom professional spider with storage management. Reinforcement learning was introduced into web spider's learning process, and the hidden structure information obtained by training the link text guided the spider to perform the work [16]. Diligenti *et al.* [17] proposed a search strategy based on a context diagram, which was used to construct a typical context diagram to estimate the distance from the target webpage. Haveliwala [18] proposed a topic-sensitive PageRank algorithm to avoid the theme drift problem of the algorithm. Although these research works have reduced the amount of noise in results, many of them still cannot effectively capture the intentions of users.

To address this issue, current solutions are applied to personalized search. Jing and Baluja [19] designed a personalized information acquisition system in which the web crawler achieved a high acquisition accuracy. Berkhin [20] presented a character-sensitive PageRank calculation formula. Leung *et al.* [21] proposed a personalization approach based on query clustering. Leung *et al.* [22] proposed a new web search personalization approach that captured the user's interests and preferences in the form of concepts by mining search results and their click-through. Leung *et al.* [23] proposed a personalized mobile search engine (PMSE) that captured the users' preferences in the form of concepts by mining their click-through data. Divya and Robin [24] presented an algorithm in the personalization of web searches, called a Decision Making Algorithm, to classify the content in the user history. Gardarin *et al.* [25] discussed an ontology-based web information system (SEWISE) to support web information description and retrieval.

There have been studies on click-through data of relevant feedback being introduced into retrieval systems. Sderlind [26] analyzed the user click behavior when browsing retrieval results. Guo *et al.* [27] compared two users' click-through model on a click chain model (CCM) and a dependent click model (DCM), and the experiment confirmed that the CCM model performed better. Wendt and Lewis [28] introduced user click-through data as measured parameters to improve the data quality of training algorithms. Zhang *et al.* [29] introduced user click-through data into image retrieval to retrieve more accurate results. Cui *et al.* [30] adopted extracting vocabulary in the retrieval results to improve the accuracy. In addition, substantial research has been performed on search result reordering using user click-through data [31], [32] that has confirmed the effectiveness of click-through data [33].

Generally, although most approaches on personalization have achieved good performance, some difficulties, such as real-time performance and user experience, prevent them from being widely applied. Our method provides an accurate and intelligent retrieval framework with real-time location and relevant feedback technology for personalized websites. Our framework captures certain aspects, and the experimental results demonstrate the method's effectiveness.

## III. INTELLIGENT RETRIEVAL FOR PERSONALIZED WEBSITES

In this section, we present the framework of intelligence retrieval and demonstrate how to use real-time location information to assist in retrieval for personalized websites. There are four main parts to the our proposed intelligence retrieval algorithm: (a) real-time location and web configuration, (b) keywords extraction, (c) relevant feedback and (d) personalized ranking. In our proposed method, we assume that the server has already collected some website framework information in a database to guarantee that the server can return appropriate results faster.

### A. FRAMEWORK OVERVIEW

An overview of the intelligence retrieval framework is shown in Fig. 1. The work process mainly consists of four steps: (a) real-time location and configuration personalized website, (b) retrieval, (c) performance optimization and (d) re-retrieval and a list of the final returned results.

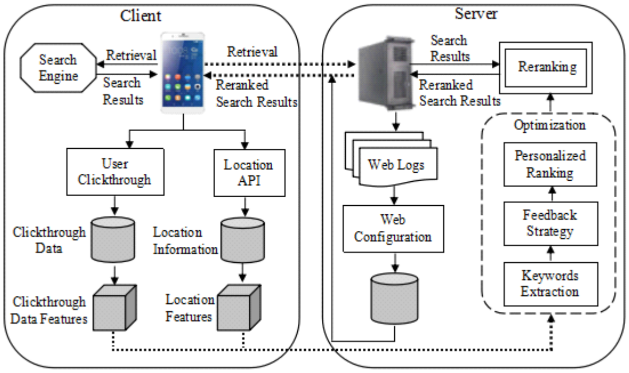

**FIGURE 1. An overview of intelligent retrieval framework.**

In the first step, the user's real-time location information (the latitude and longitude of the user) is obtained from the Location API and uploaded to the server. The user retrieves the current location area after setting the retrieval range, thereby acquiring the name and address information of the nearby building. Then, the information features are processed into a long text using the proposed keywords extraction algorithm to extract the keywordss. The current personalized website is then determined.

In the second step, the server captures information of the initial websites in the web configuration file and calculates the initial PageRank value of all the webpages. The user can modify the local website list in the client and upload it to the sever. The client groups the users according to the keywordss entered the first time. After the server receives the keywords request, the information retrieval is performed on the corresponding personalized website, and the web log also records the information. The retrieval results are then returned to the client.

In the third step, the results returned from the server are not directly displayed to the users. The client uses the click-through data acquisition strategy (feedback strategy) introduced to record user behavior and upload it to the server. The server then analyzes the user's click-through data features and updates the PageRank value through the personalized ranking method proposed to make subsequent retrievals more relevant to users' requirements.

### B. REAL-TIME LOCATION AND WEB CONFIGURATION

Users have different requirements for the same query in different scenarios. Consequently, we consider their location to obtain personalized query results. Real-time location and web configuration are two important issues facing the intelligent retrieval framework. The detailed process of location and web configuration is illustrated in Fig. 2.

Generally, a mobile user obtains location information via GPS, including the current latitude and longitude. We observe two main jobs in extracting real-time location information. First, we can retrieve the information of the current surrounding location by the API of the Baidu map and acquire

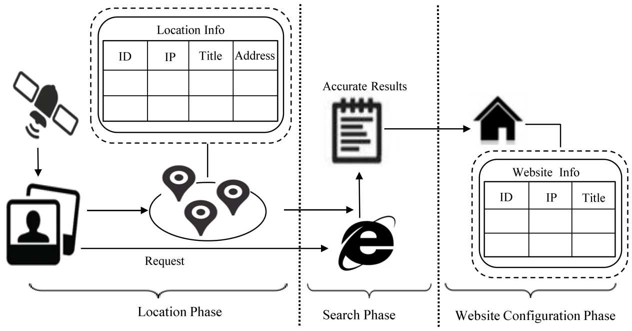

**FIGURE 2.** **Process of location and web configuration.**

detailed information of the surrounding buildings by setting the coordinates (location information and retrieval range as the parameters) such as id, name, and location. Second, we extract the ''title''and ''address'' information of the located surrounding buildings and combine the information into a long text. Thus, we can obtain the website name of the current user's personalized search range.

The frameworks of personalized websites vary, and thus, we design web configuration files to achieve faster retrieval. The user can modify the file according to their own needs for personalized retrieval. The web configuration includes three phases. First, the user sets the search range according to the user's own needs, and then, the corresponding location information is saved to the Location Info. Second, an accurate retrieval result is obtained after the location information and the request are extracted as keywordss; then, the request is sent to the server. Finally, the personalized website is configured automatically in the server according to the Location Info.

### C. KEYWORDS EXTRACTION

This paper proposes an optimized algorithm for the keywords extraction algorithm based on a statistical model, which calculates the frequency of words emerging in the text of important locations such as ''first line'', ''first'' and ''tail''. Therefore, the text is broken into clauses, and a public substring is extracted to calculate the frequency of each word in two clauses one by one when extracting the keywordss.

The text is first broken into clauses, which go with one another by permutation and combination. Then, we use the optimized public substring extraction algorithm to address the clause set. Finally, we extract the keywordss of the text according to the weights of the candidate keywordss, which depend on the word frequency and word length. The public substring extraction algorithm is shown in Algorithm 1.

The optimized keywords extraction algorithm presents improvements in terms of its space and time complexities. In terms of space complexity, it adjusts the construction method of the matrix and changes the string traversal method based on the traditional LCS algorithm. For instance, if two string lengths are p and q, the space complexity of the traditional LCS algorithm is O(pq). Because we adopt the most frequent character (m) as the height of the matrix, the space complexity of the proposed algorithm is improved to O(pm). This is obviously less than O(pq). In addition, this structure mode of the matrix does not affect extracting the public substring and allows the frequency of the public substring to be recorded more easily. In terms of time complexity, the time complexity of the string traversal is still O(pq), but the total running time of the optimized algorithm is reduced to a certain degree due to the height of the matrix being reduced.

**Algorithm 1** Public Substring Extraction Algorithm

```
 input  : str1[],str2[]
output: pstr[]
int rowLength;
index[rowLength][str1.length()];
int row=0;
for $i = 0$ to $str2.length$ do
    for $i = 0$ to $str1.length$ do
        if $str2.getChar(i) == str1.getChar(j)$ then
            if $index[row][j]==-1$ then
                index[row][j]=i;
            end
            else if $index[row][j] > -1$ then
                row++;
                index[row][j]=i;
            end
        end
    end
end
```

### D. RELEVANT FEEDBACK

A practical search engine, especially on the Internet, should provide a convenient user experience. If we want to improve the retrieval performance, the extra overhead of user feedback should be minimized.

We randomly generate user click-through data of top 10 links. The distribution information of a user click-through is shown in Fig. 3. Obviously, when users retrieve information from search engines, usually only the top few retrieval results will attract the attention of the user. If the retrieval results that the user needs are further back, the user cannot obtain useful information from this retrieval. In addition, each user has different concerns about the target link even if they input the same keywordss as a retrieval condition. Therefore, we can obtain a certain value from the information from the user's click-through data. This paper presents a strategy to obtain users click-through data via implicit feedback, which can improve the performance of search engines and the user's satisfaction.

*Definition 1: Click set (CS): Given a query keywords with accessible links and the CS satisfying*

$$CS = (ID, Q, R, C)$$

*where*

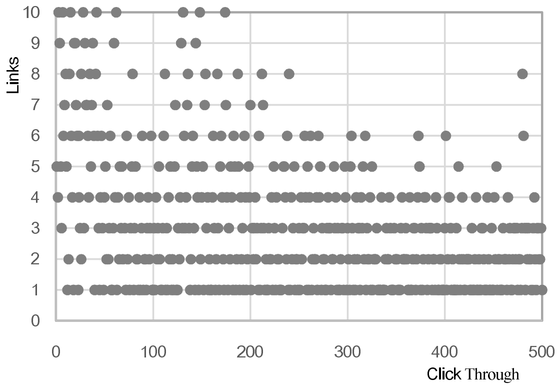

**FIGURE 3.** Distribution information of user click-through.

*ID is number of the user's interest group and used to distinguish users in different groups;*

*Q is a query keywords, which shows the query conditions of the retrieval;*

*R denotes a collection of all links returned from the search engine, in which the order of the links in the set is the display order on the webpage; and*

*C denotes a collection of all links clicked by the user.*

*Definition 2: Feedback Set (FS): The FS is used to indicate the relevant feedback information obtained from the click data analysis, and the FS satisfies*

$$FS = (ID,map)$$

*where*

*map is a relational table that stores relative degrees of correlation between two webpages.*

We analyze user behavior and propose optimized strategies to obtain relevant feedback. In our strategies, the relevant degree is low when a link is in the front and unclicked. If a link is not clicked and the previous link is clicked by the user, the relevant degree of the link is low. If a link is clicked by the user, the relevant degree of the link is higher than the previous unclicked links of the link. The relevant degree is higher between the link clicked more often and query keywordss. The relevant feedback algorithm is shown in Algorithm 2.

N is the number of R sets. link(i) represents the ith link in the linked collection returned from the search engine. The relationship $(l_i,l_j)$ indicates that the relevant degree of link i is higher than that of link j for the keywordss used in this query. num(link(i) represents the number of clicks of link i.

### *E. PERSONALIZED RANKING METHOD*

The traditional PageRank algorithm is implemented based on linkage relations between webpages, but it ignores the importance of the webpages for different users. The paper presents a method whereby the relevant feedback information obtained from the click-through data is introduced into the PageRank algorithm. According to the proposed relevant feedback information extraction strategy, we obtain the map table for the relationship of relevant degrees between links.

**Algorithm 2** Relevant Feedback Algorithm

```
input  : CS(ID,Q,R,C)
output: FS(ID,map)
for $i = 1$ to $n$ do
    for $j = 1$ to $n$ do
        if $1<=j<i<=n$ then
            if $link(i) \in C$ && $link(j) \notin C$ then
                $(l_i,l_j)$ stored in map;
            end
            else if $1<=i<=n-1$ then
                if $link(i) \in C$ && $link(i+1) \notin C$ then
                    $(l_i,l_{i+1})$ stored in map;
                end
            end
            else if $i=n$ then
                if $link(i) \in C$ then
                    for $j = 1$ to $n$ do
                        $(l_i,l_j)$ stored in map;
                    end
                end
            end
            else if $link(i) \in C$ && $link(j) \in C$ then
                if $num(link(i))>num(link(j))$ then
                    $(l_i,l_j)$ is stored in map;
                end
            end
        end
    end
end
```

However, the personalized PageRank value is influenced by not only the relationships among links but also the user click behavior. Thus, we regularly update the map table to more accurately reflect the current retrieval intention for the same group user.

The improvement in the traditional PageRank algorithm consists in adding a vector q, which represents the modification of the PageRank value using the relevant feedback information obtained from the click-through data. During the traversal of the map table, if the relevancy of link $l_i$ for the same keywords is greater than link $l_j$ and the webpage weight of link $l_i$ is less than link $l_j$, we modify the weight stored in the database by the vector q. The calculation is as follows:

$$q[l_i] = \frac{\sum_{(l_i,l_j)}^{(Rank(l_i)-Rank(l_j))}}{2} / N(l_i, l_j) \quad (1)$$

$$q[l_j] = -q[l_i] \quad (2)$$

Rank$(l_i)$ represents the current weight of the link $l_i$ in the database, and N$(l_i,l_j)$ represents the number of relationships in the relevancy table. The click status of a user cannot represent other users; thus, we need to analyze and merge the click-through data of different users, which gradually makes the vector q perfect. Formula (3) represents the accumulation process of the modified vector q.

$$q_{old}[l_i] = k_1 q_{old}[l_i] + k_2 q_{new}[l_j] \tag{3}$$

$q_{old}[l_i]$ represents the original value of the modified vector for link $l_i$. $q_{new}[l_j]$ indicates the modified value calculated based on the relevancy of the newly acquired click-through data. Introducing the modified vector q into the traditional PageRank equation, the following formula (4) is obtained:

$$\forall l_i Rank_{n+1}(l_i) = \sum_{l_j \in B_{l_i}} Rank_n(l_i)/N_{l_j} + q[l_i] \tag{4}$$

$B_{l_i}$ represents the collection of all links in, and $N_{l_j}$ represents the total number of chain links to the webpage. For formula (4), a variable d is added to control the coefficient of the modified vector q and the traditional PageRank value. The calculation is as follows:

$$\forall l_i Rank_{n+1}(l_i) = d * \sum_{l_j \in B_{l_i}} Rank_n(l_i)/N_{l_j} + (1-d)q[l_i] \tag{5}$$

Formula (4) and formula (5) add the modified vector q to the traditional PageRank. The corresponding formula including the webpage access probability C is as follows:

$$\forall l_i Rank_{n+1}(l_i) = \frac{d * [(1-C) + C * \sum_{l_j \in B_{l_i}} Rank_n(l_i)]}{N_{l_j}} + (1-d)q[l_i] \tag{6}$$

The relevant feedback information provided by different users is different, and the value of the modified vector q is different; therefore, the calculated personalized PageRank value also shows significant differences. Therefore, even if users of different groups use the same retrieval keywordss, the retrieved results will be reordered based on the value of the personalized PageRank. The calculation process of this personalized PageRank algorithm is shown in Algorithm 3.

```
Algorithm 3 Personalized PageRank Algorithm
input  : the relation of the link;
         the relevant feedback information
output: personalized PageRank value
while the PageRank value converges do
    calculate PageRank value of webpage;
    calculate the value of related feedback vector
    according to formulas(1), (2), (3);
    calculate personalized PageRank value according to
    formula(6);
end
```

The proposed personalized PageRank value is merged into the webpage weight and user behavior influence factor. For the result ranking, we still add the webpage relevant degree to make the results more accurate.

**TABLE 1.** Test environment.

| Test environment | Name |
|---|---|
| Server OS | Mac OS 10.11.5 |
| Server Memory | 8GB |
| Server Version | Tomcat 7.0 |
| Server Database Version | MySQL 5.7.13 |
| Mobile Terminal Type | Huawei Mate9 |
| Mobile Terminal OS | Android 7.0 |
| Mobile Terminal Database | SQLite 3.8.8 |

## IV. EXPERIMENTAL RESULTS

### A. EXPERIMENTAL SETUP

It is important to establish the experimental database. We perform the experiments in a real Web environment. Table 1 shows the experimental environment. The intelligent retrieval framework provides a convenient operating interface, which is similar to a commercial search engine. Users can type keywordss into the interface and submit the information to the server.

### B. KEYWORDS EXTRACTION EFFICIENCY

The first experiment illustrates the extraction accuracy and time cost between the proposed keywords extraction algorithm and four traditional public substring extraction algorithms. The experimental dataset includes 1000 experimental sets, including the abstract and keywordss extracted from papers in the Baidu academic and CNKI platforms.

#### 1) EXTRACTION ACCURACY

The text length strongly influences the keywords extraction accuracy, and thus, the experiment only tests a single text dozens of times. The accuracy of the keywords extraction algorithm is estimated by the similarity between the results of the algorithms and the real data. Therefore, the keywords extraction accuracy can be defined as follows:

$$accuracy = \frac{1}{n}\sum_{i=1}^{n} Sim(i) \tag{7}$$

where n is the number of keywords extraction results and Sim(i) represents the similarity of the number i with the range of $0 \leq Sim(i) \leq 1$.

From Fig. 4, the accuracy of the keywords extraction under the proposed algorithm is higher than that of traditional public substring extraction algorithms. In this paper, the proposed keywords extraction algorithm does not limit the length of the word or phrase, considering the number of words and phrases in the text. Thus, when extracting keywordss, the "off" phenomenon whereby the keywordss form a meaningless phrase is avoided. In addition, it better matches the theme of the text that the proposed algorithm introduces into the word frequency.

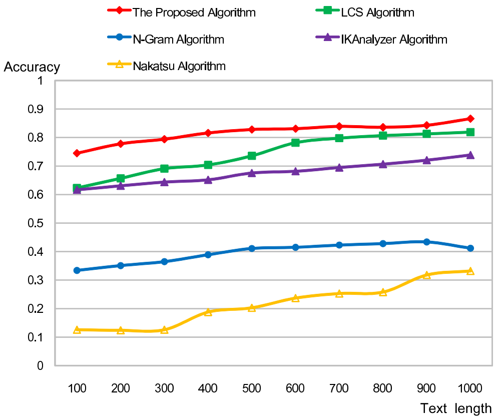

**FIGURE 4.** Relation between keywords extraction accuracy and text length.

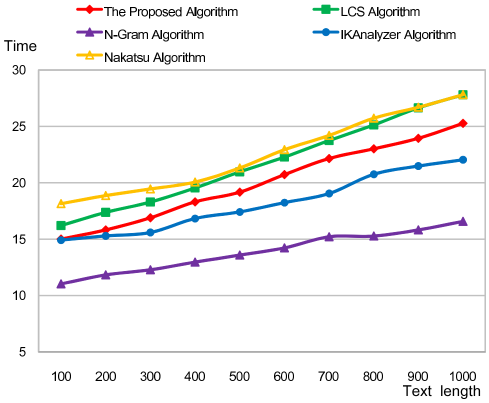

**FIGURE 5.** Relation between keywords extraction time and text length.

## 2) TIME COST

The main influences on the algorithm execution time are the length and quantity of keywordss. The algorithm execution time increases with increasing length and quantity of text, as shown in figures 5 and 6.

From Fig. 5, we draw the following conclusions: (1) the proposed algorithm and the LCS algorithm are similar in time complexity because the two algorithms need to extract public substrings from clauses of the text; however, the proposed algorithm achieves a relatively low space complexity in the presence of high frequencies, and thus, the time consumed by the scanning array is relatively short. The time complexity of the LCS algorithm is still higher than the proposed algorithm. (2) The keywords extraction time of the Nakatsu algorithm was the highest because of the complex steps needed for the high-frequency words. (3) The keywords extraction procedure of the N-Gram algorithm is relatively simple. Only one word segmentation process is performed; therefore, the time complexity is the lowest. (4) The IKAnalyzer algorithm has the same matching process as the N-Gram algorithm; therefore, the time complexity is higher than the N-Gram algorithm.

Fig. 6 shows the relation between the keywords extraction time and text number. Because the text length also affects the execution time of the algorithm, we choose a test text that is approximately 500 characters. When the amount of text increases, the keywords extraction time will also increase. For the same number and length of text, the execution time of the proposed algorithm is in the middle.

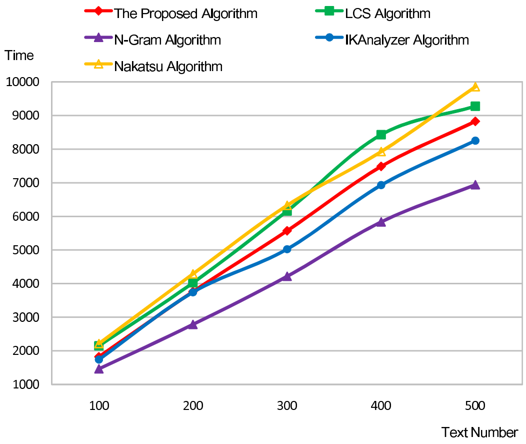

**FIGURE 6.** Relation between keywords extraction time and text amount.

**TABLE 2.** The results of partial data in map.

| no. | results in map |
|---|---|
| 1 | $(l_5, l_2)$ |
| 2 | $(l_5, l_3)$ |
| 3 | $(l_5, l_4)$ |

## *C. RELEVANT FEEDBACK EXTRACTION STRATEGY EFFICIENCY*

The second experiment tests the relevancy between the webpage results and the query keywordss according to the relevant feedback extraction strategy based on the click-through data analysis. In the simulation experiment, we set 20 static links in the android client and 10 links per page. We first set the initial 2-tuple (ID, map) in the database. Then, we click on two links, 1 and 5, of the 20 links in sequence and click once per link. The partial data in the map table are shown in table 2.

From table 2, link 5 is before links 2, 3 and 4, which were not clicked; thus, the relevant degree between the links and the query keywordss is less than link 5.

We then click on five links 1, 6, 8, 11 and 15 successively after truncating the map table, and each link is clicked once. The partial data in the map table are shown in table 3. We find from table 3 that the relevancy of the link that is not clicked and behind the clicked link is low. In addition, the relevancy of link 15 is higher than those all the other clicked links because it is the last link clicked. The results are in agreement with the phenomenon of the proposed strategy.

We click link 1 five times, link 3 two times and link 9 seven times after clearing the map table again. The partial data in

**TABLE 3.** The results of partial data in map.

| no. | results in map |
|---|---|
| 1 | $(l_1, l_2)$ |
| 2 | $(l_6, l_7)$ |
| 3 | $(l_8, l_9)$ |
| 4 | $(l_{11}, l_{12})$ |
| 5 | $(l_{15}, l_{16})$ |
| 6 | $(l_{15}, l_1)$ |
| 7 | $(l_{15}, l_6)$ |
| 8 | $(l_{15}, l_8)$ |
| 9 | $(l_{15}, l_{11})$ |

**TABLE 4.** The results of partial data in map.

| no. | results in map |
|---|---|
| 1 | $(l_1, l_3)$ |
| 2 | $(l_9, l_1)$ |
| 3 | $(l_9, l_3)$ |

the map table are shown in table 4. Table 4 illustrates that the more a link is clicked, the higher the relevancy of the link is. The results are in agreement with the phenomenon of the proposed strategy.

### D. PERSONALIZED RANKING PERFORMANCE

The third experiment tests the performance of the proposed personalized ranking method. We first write 10 simple HTML files as test webpages, and these webpages have the same web structure, that is, the same initial PageRank value. For the same keywordss, each of the 10 webpages will be given different similarity values. We click on 10 webpages many times, where the number of clicks is different, therein recording and analyzing the PageRank value and rankings of the 10 webpages.

The PageRank values of the 10 webpages are first stored. The PageRank value is only related to the webpage structure of the webpage because there is no user click operation. A total of 100 clicks on the 10 webpages are distributed on different webpages. The current PageRank value is shown in table 5.

From table 5, a higher number of clicks results in a greater increase in the personalized PageRank value. Simultaneously, personalized PageRank values with fewer clicks do not fluctuate. The personalized PageRank value of a webpage that is not clicked declines correspondingly.

We increase the number of clicks to 300 for 10 pages and record the change in the personalized PageRank value after each click. In Fig. 7, We randomly extract the personalized PageRank value of 3 pages. From Fig. 7, when the webpage is clicked, the personalized PageRank value of the webpage will correspondingly increase; otherwise, it will be reduced or remain essentially unchanged. Even if the webpages in the rear of the retrieval result list are not clicked, the PageRank value also will not change; however, if some links before the clicked webpage are not clicked, the PageRank value will be reduced because the webpage is not of interest to the user.

**TABLE 5.** The status of personalized PageRank value.

| webpage no. | clicks | initial PageRank value | personalized PageRank value |
|---|---|---|---|
| 1 | 14 | 2.37E-2 | 2.96E-2 |
| 2 | 18 | 2.37E-2 | 3.12E-2 |
| 3 | 9 | 2.37E-2 | 2.76E-2 |
| 4 | 21 | 2.37E-2 | 3.24E-2 |
| 5 | 2 | 2.37E-2 | 2.30E-2 |
| 6 | 3 | 2.37E-2 | 2.31E-2 |
| 7 | 6 | 2.37E-2 | 2.39E-2 |
| 8 | 26 | 2.37E-2 | 3.38E-2 |
| 9 | 0 | 2.37E-2 | 2.17E-2 |
| 10 | 1 | 2.37E-2 | 2.17E-2 |

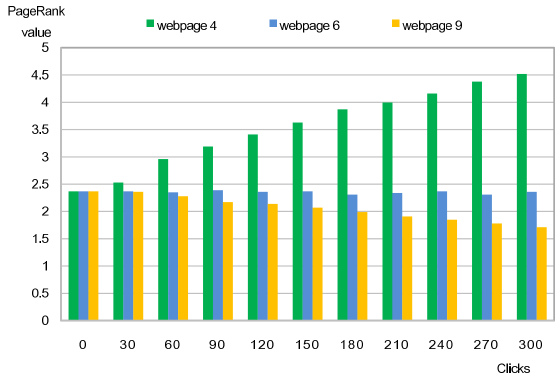

**FIGURE 7.** PageRank value of webpages with different numbers of clicks.

Fig. 8 represents the ranking of the 10 pages in the result list when the user performs 0, 100 and 300 clicks. When the user is not clicking first these webpages are sorted by page similarity, creating a straight line. However, when the user performs multiple clicks, the ranking will change due to the influence of the personalized PageRank value. The webpages with more clicks will be ranked in higher positions. When the number of clicks increases again, the line is basically stable, indicating that the ranking result is in line with the retrieval intentions of the majority of users.

## V. CONCLUSIONS AND FUTURE WORK

In this paper, we propose a new approach for an intelligent retrieval framework with real-time location in CPSSs to resolve ambiguities for general search engines. We first present an intelligent retrieval model for a single field with real-time location. Second, to improve the retrieval results, the paper proposes a strategy for implicit correlation feedback based on click-through data analysis, which obtains the relationship between the user query conditions and retrieval results. Finally, the paper designs a personalized PageRank algorithm including modified parameters to improve the

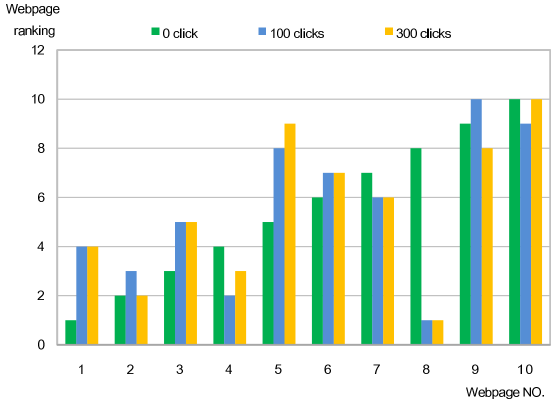

**FIGURE 8.** PageRank value of Webpages in Different Clicks.

ranking quality of the retrieval results using the relevant feedback from other users in the interest group.

We have performed several experiments to evaluate the performance of the proposed framework. Comparisons performed from experiments demonstrate that the proposed framework obtains remarkable retrieval performances with minimum effort and provides a superior user experience.

The proposed framework provides an efficient, intelligence, real-time location-oriented personalized retrieval approach in CPSSs. Although we have proven the efficiency and effectiveness of the proposed framework, in the future, we will concentrate on thoroughly investigating several improvements to the compatibility and usability of the proposed framework.

## ACKNOWLEDGMENT

The authors declare that they have no conflicts of interests.

## REFERENCES

[1] X. Wang, L. T. Wang, X. Xie, J. Jin, and M. J. Deen, "A cloud-edge computing framework for cyber-physical-social services," *IEEE Commun. Mag.*, vol. 55, no. 11, pp. 80–85, Nov. 2017.

[2] X. Wang, L. T. Yang, J. Feng, X. Chen, and M. J. Deen, "A tensor-based big service framework for enhanced living environments," *IEEE Cloud Comput.*, vol. 3, no. 6, pp. 36–43, Nov./Dec. 2016.

[3] J. Zeng, L. T. Yang, M. Lin, H. Ning, and J. Ma, "A survey: Cyber-physical-social systems and their system-level design methodology," *Future Generat. Comput. Syst.*, Aug. 2016.

[4] X. Liu and H. Turtle, "Real-time user interest modeling for real-time ranking," *J. Amer. Soc. Inf. Sci. Technol.*, vol. 64, no. 8, pp. 1557–1576, 2013.

[5] J. Liu and N. J. Belkin, "Personalizing information retrieval for multi-session tasks: Examining the roles of task stage, task type, and topic knowledge on the interpretation of dwell time as an indicator of document usefulness," *J. Amer. Soc. Inf. Sci. Technol.*, vol. 66, no. 1, pp. 58–81, 2015.

[6] R. Li, Z. Xu, W. Kang, K. C. Yow, and C.-Z. Xu, "Efficient multi-keyword ranked query over encrypted data in cloud computing," *Future Generat. Comput. Syst.*, vol. 30, no. 1, pp. 179–190, 2014.

[7] Y. Wu, L. Shou, T. Hu, and G. Chen, "Query triggered crawling strategy: Build a time sensitive vertical search engine," in *Proc. IEEE Int. Conf. Cyberworlds*, Sep. 2008, pp. 422–427.

[8] E. Agichtein, E. Brill, and S. Dumais, "Improving Web search ranking by incorporating user behavior information," in *Proc. 29th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, New York, NY, USA, 2006, pp. 19–26.

[9] A. Chuklin and I. Ḋ. R. M. Markov, *Click Models for Web Search* (Synthesis Lectures on Information Concepts, Retrieval, and Services). San Rafael, CA, USA: Morgan & Claypool, 2015.

[10] S. Xu, H. Jiang, and F. C. M. Lau, "Mining user dwell time for personalized Web search re-ranking," in *Proc. 20th Int. Joint Conf. Artif. Intell.*, Palo Alto, CA, USA, 2011, pp. 2367–2372.

[11] X. Yi, L. Hong, E. Zhong, N. N. Liu, and S. Rajan, "Beyond clicks: Dwell time for personalization,"in *Proc. 8th ACM Conf. Recommender Syst.*, New York, NY, USA, 2014, pp. 113–120.

[12] M. A. Babenko and T. A. Starikovskaya, "Computing the longest common substring with one mismatch," *Problems Inf. Transmiss.*, vol. 47, no. 1, pp. 28–33, 2011.

[13] C. Y. Choong, Y. Mikami, and R. L. Nagano, "Language identification of Web pages based on improved n-Gram algorithm," *Int. J. Comput. Sci.*, vol. 8, no. 3, pp. 47–58, 2011.

[14] Y. Gao, F. Song, X. Xie, Q. Sun, and X. Wu, "Study of test classification algorithm based on domain knowledge," in *Proc. Int. Conf. Cyberspace Technol.*, 2015, pp. 1–5.

[15] S. Chakrabarti, M. H. van den Berg, and B. E. Dom, "Distributed hypertext resource discovery through examples," in *Proc. VLDB*, 1999, pp. 375–386.

[16] J. Rennie and A. Mccallum, "Using reinforcement learning to spider the Web efficiently," in *Proc. 6th Int. Conf. Mach. Learn.*, 1999, pp. 335–343.

[17] M. Diligenti, F. Coetzee, S. Lawrence, C. L. Giles, and M. Gori, "Focused crawling using context graphs," in *Proc. Int. Conf. Very Large Data Bases*, 2000, pp. 527–534.

[18] T. H. Haveliwala, "Topic-sensitive PageRank: A context-sensitive ranking algorithm for Web search," *IEEE Trans. Knowl. Data Eng.*, vol. 15, no. 4, pp. 784–796, Jul. 2003.

[19] Y. Jing and S. Baluja, "VisualRank: Applying pagerank to large-scale image search," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 11, pp. 1877–1890, Nov. 2008.

[20] P. Berkhin, "A survey on pagerank computing," *Internet Math.*, vol. 2, no. 1, pp. 73–120, 2005.

[21] K. W. T. Leung, W. Ng, and D. L. Lee, "Personalized concept-based clustering of search engine queries," *IEEE Trans. Knowl. Data Eng.*, vol. 20, no. 11, pp. 1505–1518, Nov. 2008.

[22] K. W.-T. Leung, D. L. Lee, and W.-C. Lee, "Personalized Web search with location preferences," in *Proc. IEEE Int. Conf. Data Eng.*, vol. 41. Mar. 2010, pp. 701–712.

[23] K. W.-T. Leung, D. L. Lee, and W.-C. Lee, "PMSE: A personalized mobile search engine," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 4, pp. 820–834, Apr. 2013.

[24] R. Divya and C. R. R. Robin, "Onto-search: An ontology based personalized mobile search engine," in *Proc. IEEE Int. Conf. Green Comput. Commun. Elect. Eng.*, Mar. 2014, pp. 1–4.

[25] G. Gardarin, H. Kou, K. Zeitouni, X. Meng, and H. Wang, "SEWISE: An ontology-based Web information search engine," in *Proc. Natural Lang. Process. Inf. Syst., Int. Conf. Appl. Natural Lang. Inf. Syst.*, Jun. 2008, pp. 106–119.

[26] P. Söderlind, "Nominal interest rates as indicators of inflation expectations," *Scandin. J. Econ.*, vol. 100, no. 2, pp. 457–472, 1998.

[27] F. Guo *et al.*, "Click chain model in Web search," in *Proc. ACM Int. Conf. World Wide Web*, 2009, pp. 11–20.

[28] C. Wendt and W. Lewis, "Improving the quality of a customized SMT system using shared training data," *Inproceedings*, 2009.

[29] Y. Zhang, X. Yang, and T. Mei, "Image search reranking with query-dependent click-based relevance feedback," *IEEE Trans. Image Process*, vol. 23, no. 10, pp. 4448–4459, Oct. 2014.

[30] H. Cui, J.-R. Wen, J.-Y. Nie, and W.-Y. Ma, "Probabilistic query expansion using query logs," in *Proc. 11th Int. Conf. World Wide Web*, 2002, pp. 325–332.

[31] B. Smyth, E. Balfe, J. Freyne, P. Briggs, M. Coyle, and O. Boydell, "Exploiting query repetition and regularity in an adaptive community-based Web search engine," *User Model. User-Adapted Interact.*, vol. 14, no. 5, pp. 383–423, 2004.

[32] R. Burke and M. Ramezani, "Matching recommendation technologies and domains," in *Recommender Systems Handbook*. Boston, MA, USA: Springer, 2011, pp. 367–386.

[33] N. Y. Abdullah, H. S. Husin, H. Ramadhani, and S. V. Nadarajan, "Pre-processing of query logs in Web usage mining," *Ind. Eng. Manage. Syst.*, vol. 11, no. 1, pp. 82–86, 2012.

**YAYUAN TANG** is currently pursuing the Ph.D. degree with the School of Information Science and Engineering, Central South University. Her research interests include social computing, ubiquitous computing, and big data retrieval.

**HAO WANG** is currently an Associate Professor and the Head of the Big Data Lab, Department of ICT and Natural Sciences, Norwegian University of Science and Technology, Norway. His research interests include big data analytics and industrial Internet of Things, high-performance computing, safety-critical systems, and communication security.

**KEHUA GUO** received a Ph.D. degree in computer science and technology from the Nanjing University of Science and Technology. He is a currently a Professor with the School of Information Science and Engineering, Central South University. His research interests include social computing, ubiquitous computing, big data, and image retrieval.

**YIZHE XIAO** received the master's degree in computer science and technology from Central South University in 2017. His research interests include green computing and big data retrieval.

**TAO CHI** received the Ph.D. degree from Tongji University, China. He is currently a Professor with the Key Laboratory of Fisheries Information, Ministry of Agriculture, Shanghai Ocean University. His research fields include Internet of Things and ubiquitous computing.