# Recommendations of Personal Web Pages Based on User Navigational Patterns

Yin-Fu Huang and Jia-Tang Jhang

***Abstract*—In this paper, we propose a web recommendation system where user navigational patterns can be extracted from web logs. First, the recommendation system discovers user concepts from web logs step-by-step, and then extracts the navigation patterns among these concepts. These navigational patterns are then used to generate recommendation web pages by matching the navigation behavior of a user personal knowledge base. The pages in a recommendation list are ranked according to their hub scores which are computed based on page connectivity information. The experimental results show that the web pages recommended by our system are of better quality and acceptable for humans from various domains, based on human evaluators ranking as well as quality-value-based performance measures.**

***Index Terms*—Information retrieval, recommendation system, web mining, web search.**

## I. Introduction

As more and more information is available on the Internet, web search has become an essential tool for users. However, sometimes a user query is ambiguous and cannot clearly describe what he/she wants. For example, if the keyword specified in a user query is "apple," it cannot indicate fruit or computer science explicitly. In this paper, we propose a solution to this problem by constructing a web recommendation system which integrates personal knowledge bases over different domains to help users to extract desired information.

A web recommendation system is an online information system that recommends relevant items to users. The recommended items could be products, movies, or even on-line resources such as web pages. Typically, a web recommendation system is composed of an off-line module and on-line module. The off-line module discovers user navigation patterns from web logs, while the on-line module matches the navigation behavior of a current user with discovered navigation patterns to produce a recommendation list.

In recent years, many recommendation systems have been developed to facilitate web search. However, the common problem with these recommendation systems is that they generally require a great amount of documents collected from the visited websites by users and the information from user interactions. In this paper, we propose a novel approach instead of collecting a huge amount of data. In general, a user session has more than one query to be fulfilled [1]. Thus, the queries of a user session in web logs can be clustered based on the probabilities in different domains, and these clusters are called concepts. In other words, a concept is a session with coherent information, which is constructed dynamically and not pre-defined. Then, these concepts are augmented with their connected neighborhoods and finally generate navigation patterns. Next, the navigation patterns already captured are compared with the semantic graph in a personal knowledge base. Finally, we recommend the most relevant web pages in the matched clusters.

The remainder of the paper is organized as follows. First, related basic concepts and personal knowledge bases are introduced in Section II. Then, Section III describes each component in the system architecture. Section IV discusses the experimental results. Finally, we make conclusions in Section V.

Manuscript received February 13, 2014; revised May 12, 2014. This work was supported by National Science Council of R.O.C under Grant NSC100-2218-E-224-011-MY3.

Yin-Fu Huang is with National Yunlin University of Science and Technology, Yunlin, Taiwan 640 (e-mail: huangyf@yuntech.edu.tw).

Jia-Tang Jhang is with Tornado Technologies Co., Ltd, Taipei, Taiwan 106 (e-mail: 9917711@yuntech.edu.tw).

## II. Basic Concepts

Some techniques and approaches are used in our work. Here, we would briefly review them in the following subsections.

### A. Query Classification

The classification/categorization of web queries is usually an investigated issue in computer science. The task is to assign a web query to one or more predefined categories, based on its topics. The importance of query classification was emphasized by many services provided by web search. A direct application is to provide better web pages for users with multi-category interests. For example, a user issuing a web query "apple" might expect to browse the web pages related to fruit "apple", or he/she may prefer to look over the products or news related to the company "Apple". Online advertisement services rely on query classification results to promote products more accurately. Web pages can be grouped together according to the categories predicted by a query classification algorithm. However, the computation of query classification is non-trivial. Different from document classification tasks, the queries submitted by web users are usually short and ambiguous, and their meanings are evolving over time. Therefore, query classification is much more difficult than traditional document classification tasks [2]-[4].

### B. Fuzzy Clustering

Clustering is the task of assigning a set of objects into groups (called clusters) so that the objects in the same cluster

DOI: 10.7763/IJMLC.2014.V4.429

are more similar (in some sense or another) to each other than to those in other clusters. Typically, clustering techniques come in two notions: hard and soft. In hard clustering [5], data is divided into distinct clusters, where each data element belongs to exactly one cluster. In soft clustering (or fuzzy clustering) [6], each data element has a certain probability of belonging to each of the clusters, as shown in Fig. 1. One can think of hard clustering as a special case of soft clustering where these probabilities only take values 0 or 1.

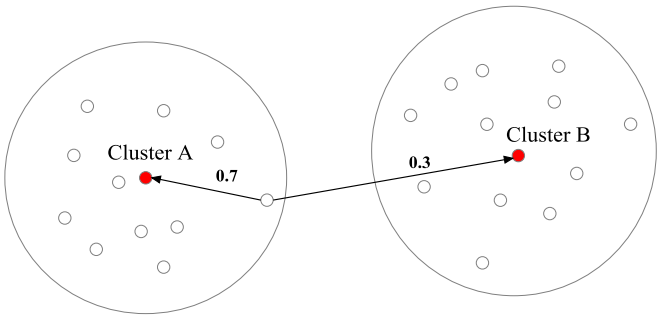


Fig. 1. Fuzzy clustering.

### C. Link Analysis Algorithm

The analysis of hyperlinks has been instrumental in the development of web search. There are several algorithms of the link analysis; e.g., the PageRank algorithm and HITS (i.e., Hypertext Induced Topic Selection) algorithm are two popular algorithms. PageRank considers the hyperlink weight normalization and the balance distribution of random surfers as the citation score [7]. HITS makes the differentiation between hubs and authorities, and estimates them in a mutually reinforcing way.

### D. Personal Knowledge Base

A personal knowledge base (PKB) is knowledge repository used to store the personal knowledge of an individual. A PKB differs from a traditional database where it contains subjective material particular to the owner. Importantly, a PKB consists primarily of knowledge, rather than information; in other words, it is not a collection of documents or other sources that an individual has encountered, but rather an expression of the distilled knowledge that the owner has extracted from those sources [8].

## III. System Architecture

### A. Overview

The architecture of the web recommendation system, as shown in Fig. 2, consists of five components: 1) preprocessor, 2) query classification, 3) concept identification, 4) HITS algorithm, and 5) recommendation engine. Since different users have their own interests in different domains such as science, art, sports *et al.*, the recommendation system discovers user concepts from web logs step-by-step, and then extracts the navigation patterns among these concepts. Afterwards, the recommendation engine identifies the navigation behavior of a current user by his/her personal knowledge base [9] and matches the behavior with the discovered navigation patterns. If navigation patterns are found, the engine would recommend a list of web-pages for his/her browsing.

### B. Preprocessor

First, the preprocessor parses or splits up web logs into three files: 1) query file, 2) user visiting file, and 3) page connectivity information. All of these file records information regarding users' request to web logs. The query file collects all of queries recorded in web logs. The user file includes anonymous user IDs, queries issued by users, and URLs clicked on search results. The page connectivity information records the connectivity of these URLs in web logs.

### C. Query Classification

Typically, the queries issued by users consist of a few words. Since these queries are usually very short and ambiguous, how to interpret the queries in terms of multiple domains is the major problem of concept identification. Here, we use the taxonomy-bridging algorithm [3] to solve this problem, which classifies a user query $Q_k$ into a set of $n$ categories $\{C_1, C_2, \ldots, C_n\}$. As a result, we can represent a query as a vector $Q_k = (p(C_{k1}), p(C_{k2}), \ldots, p(C_{kn}))$ where $p(C_{ki})$ is the probability of query $Q_k$ belonging to category $C_i$.

An example of the target categories is illustrated in Fig. 3. Because no data are provided to define the contents and semantics of a category and query, a straightforward method is to submit them to a search engine for extracting related pages. The extracted pages can help determine the meanings of the categories and queries. In [3], Shen *et al.* connect the target categories and queries by taking intermediate categories as a bridge. As illustrated in Fig. 4, the squares in the left part denote the queries to be classified; the tree in the right part represents a hierarchy organized by the target categories; the tree in the middle part is an existing intermediate taxonomy used in the Open Directory Project (ODP) [10]. The thickness of the dotted lines reflects the similarly relationship between two nodes. For example, given a target category $C_i^T$ and a query $q_k$, we can judge the similarity between them by the distributions of their relationship to the intermediate category $C_j^I$ and $C_k^I$.

For the following formula used in the taxonomy-bridging algorithm, $p(C_i^T|q)$ is the conditional probability of a target category $C_i^T$, given a query $q$. Similarly, $p(C_i^T|C_j^I)$ and $p(q|C_j^I)$ are the conditional probabilities of a target category $C_i^T$ and a query $q$ respectively, given an intermediate category $C_j^I$. $p(C_j^I)$ is the prior probability of a intermediate category $C_j^I$, which can be estimated from the web pages in the intermediate categories $C^I$. If a target category $C_i^T$ is represented by a set of words $(w_1, w_2, \ldots, w_n)$ where each word $w_k$ appears $n_k$ times, $p(C_i^T | C_j^I)$ can be calculated as $\prod_{k=1}^{n} p(w_k | C_j^I)^{n_k}$. Finally, $p(q|C_j^I)$ can be calculated in the same way as $p(C_i^T|C_j^I)$. Thus, the formula can be

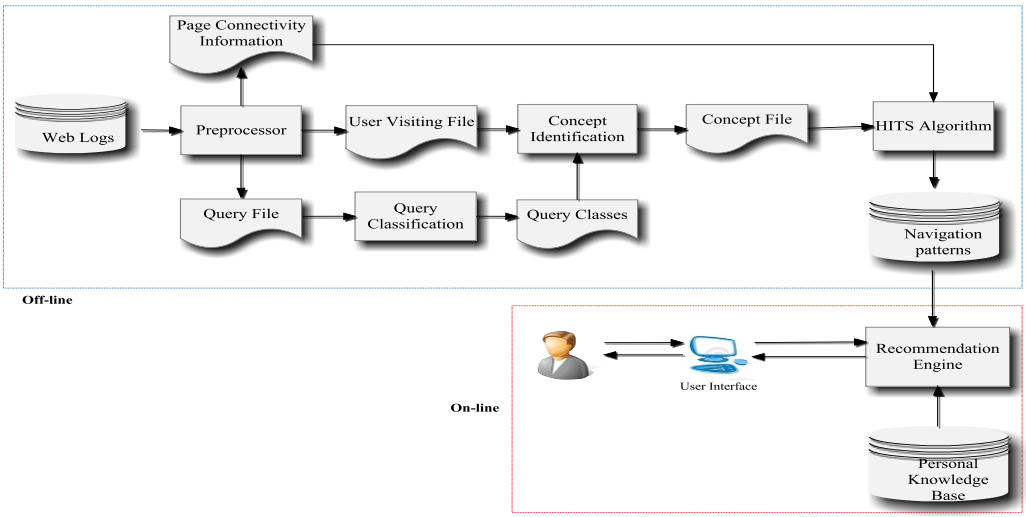


Fig. 2. System architecture.

expressed as follows:

$$p(C_i^T \mid q) = \sum_{C_j^I} p(C_i^T \mid C_j^I) \frac{p(q \mid C_j^I) p(C_j^I)}{p(q)} \quad (1)$$

where $p(C_i^T \mid C_j^I) = \prod_{k=1}^{n} p(w_k \mid C_j^I)^{n_k}$

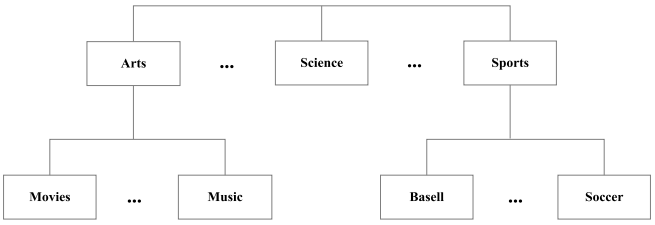


Fig. 3. Example of the target categories.

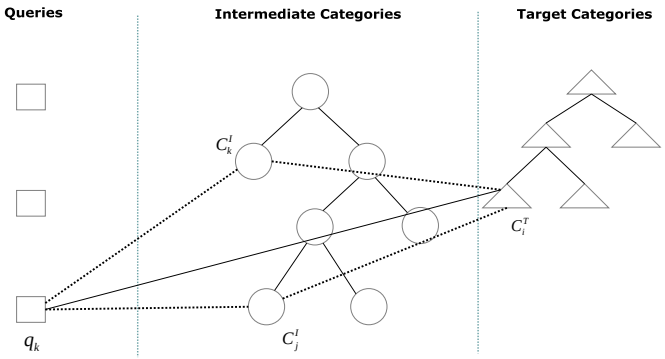


Fig. 4. Illustration of the taxonomy-bridging.

### D. Concept Identification

A web query log contains anonymous user IDs, queries issued by users, clicked URLs, and other related information. In general, session detection is done by considering the time length of a user session, and a user session has more than one query to be fulfilled [1]. Here, we propose a novel method where the queries of a user session are clustered based on the vectors of the probabilities in different target categories, and we call these clusters concepts; i.e., a concept is a session with coherent information, which is constructed dynamically and not pre-defined. For example, the concept baseball can be represented by a set of queries {American League, Boston Red Sox, Oakland Athletics, …, Major League Baseball}.

#### 1) Query Clustering

As described in Section III, a query is regarded as a vector. Then, all queries of a user session can be clustered based on their semantics from vectors. A concept models queries related to one of user intentions, and a concept can overlap other concepts since a query is usually ambiguous. One of the popular fuzzy clustering is the Fuzzy C Means (FCM) algorithm which allows each query to belong to more than one cluster. This method was developed by Dunn in 1973 [11] and improved by Bezdek in 1981 [12], and it is frequently used in pattern recognition. The FCM algorithm is based on minimizing the following objective function:

$$J_m = \sum_{i=1}^{N} \sum_{j=1}^{c} u_{ij}^m \| x_i - c_j \|^2 , \; 1 \le m < \infty \quad (2)$$

where $m$ is any real number greater than 1, $N$ is the number of measured data, $u_{ij}$ is the degree of membership of $x_i$ in the cluster $j$, $x_i$ is the $i_{th}$ d-dimensional measured data, $c_j$ is the d-dimension center of the cluster $j$, and ||*|| is the norm expressing the similarity between the measured data and center. Fuzzy partitioning is carried out through an iterative optimization of the objective function shown above, with the update of membership $u_{ij}$ and the cluster center $c_j$ by:

$$u_{ij} = \frac{1}{\sum_{k=1}^{c} \left( \frac{\| x_i - c_j \|}{\| x_i - c_k \|} \right)^{\frac{2}{m-1}}} \quad (3)$$

$$c_j = \frac{\sum_{i=1}^{N} u_{ij} \cdot x_i}{\sum_{i=1}^{N} u_{ij}^m} \quad (4)$$

The iteration will stop when $\max_{ij} \left\{ | u_{ij}^{(k+1)} - u_{ij}^{(k)} | \right\} < \varepsilon$, where $\varepsilon$ is a termination criterion between 0 and 1, and $k$ is the

iteration step. Finally, this procedure converges to a local minimum or a saddle point of $J_m$. In summary, the algorithm is composed of the following steps:

1. Initialize $U=[u_{ij}]$ matrix; i.e., $U^{(0)}$.
2. At $k_{th}$-step: calculate the center vectors $C^{(k)}=[c_j]$ using $U^{(k)}$.
3. Update $U^{(k)}$ into $U^{(k+1)}$.
4. If $| U^{(k+1)} - U^{(k)} |<\varepsilon$ then STOP; otherwise return to step 2.

Since the FCM algorithm is very sensitive to initial cluster centers, we use the ant based algorithm [13] to obtain initial cluster centers.

*2) Concept Merging*

After clustering the queries, each concept is represented by a set of queries; i.e., concept = {query$_1$, query$_2$, …, query$_n$}. Since the concepts may be similar to each other, we use query-based similarity to judge whether a pair of concepts are close enough. If a concept has strong similarity with another concept, then two concepts will be merged. The query-based similarity function is defined as follows:

$$Similariry_{query}(c_1,c_2)=\frac{C(c_1,c_2)}{Max(l(c_1),l(c_2))}>\gamma \qquad (5)$$

where $l(c)$ is the number of queries in the concept $c$, $C(c_1,c_2)$ is the number of common queries in two concepts, and γ is a threshold to judge whether a pair of concepts are close enough.

### E. HITS Algorithm

For a concept, each query is involved with at least one URL so that we can measure these URLs by link analysis. In this section, we use the HITS algorithm to measure these URLs [14] where each URL has both a hub score and an authority score. A good hub page is one that points to many good authorities; a good authority page is one that is pointed to by many good hub pages. More precisely, let $h$ and $a$ denote the vectors of all hub and all authority scores, respectively. Given a concept with a set of URLs, first, the HITS algorithm produces an n-by-n adjacency matrix $A$ whose element $A_{ij}$ is 1 if URL$_i$ references to URL$_j$ and 0 otherwise. Then, the HITS algorithm iterates the following equations:

$$\vec{h} \leftarrow A\vec{a} \qquad (6)$$

$$\vec{a} \leftarrow A^T\vec{h} \qquad (7)$$

where $A^T$ denotes the transpose of the matrix $A$. Since the iterative updates capture the intuition of good hubs and good authorities, the high-scoring URLs would be good hubs and authorities for the concept. Finally, we can regard the concept with the vectors of all hub and all authority scores as navigation patterns.

### F. Recommendation Engine

In the on-line phase, we extract top-weighted key-phrases concerning a specific category from the personal knowledge base of a current user [9] where the number of top-weighted key-phrases and the target category are specified by the current user through GUI. Then, we use these key-phrases (or keywords) to match the navigation patterns previously discovered in the HIST algorithm. For matching patterns, we use the same query-based similarity function mentioned in Section III to judge their relationship. If they are close enough (or the similarity value is more than $\gamma$), the high-scoring (or top-ranked) URLs in matched concepts would be recommended to the current user. Here, the number of recommended URLs (or top-$k_i$) from each matched concept can be expressed as follows:

$$k_i = round\left(w_i \times N\right) \qquad (8)$$

where $w_i$ is the normalized similarity ratio of concept $i$ and $N$ (also specified by the current user) is the total number of recommended URLs. In other words, the matched concepts with more similarity values would contribute more recommended URLs.

## IV. Experiments

This section describes how to evaluate the web recommendation system using discovered user concepts. Five evaluators are invited to judge the list of pages recommended by our system; i.e., assigning a numeric score to each list. The web recommendation system is implemented in Java, and the experiments are conducted on an Intel Core i7 2.93GHz CPU with 4G main memory in Window XP professional.

### A. Dataset

In this paper, we use the dataset from AOL (i.e., American Online), which is available on the Web and includes more than 30 million (non-unique) web queries collected from more than 650,000 users over three months. This dataset was sorted by user IDs and sequentially ordered. For each request, there is also information about when the query was issued, when a link was clicked, the ranks of links, and the URLs of links.

### B. Performance Measures

Here, we invite five evaluators majoring in computer sciences to rate the web pages recommended by our web recommendation system. The personal knowledge bases (in 10 specific domains) of each evaluator have been built from the web pages which he/she selects through Google and Yahoo search engines. These domains are diverse, including baseball, music, computer networking, military, dancing, gambling, car, comic & animation, investing, and religion & spirituality.

In order to observe the effectiveness of our recommendation system, we use the acceptable percentage measure defined by Zhang *et al.* [15] for rating each page as a 1-to-5 scale (1: not related, 2: poorly related, 3: fairly related, 4: well related, and 5: strongly related). Besides, we also use a quality value measure to evaluate the quality of the pages recommended by our system as follows.

The acceptable percentage measure:

$$m_1=\frac{n_3+n_4+n_5}{\sum_{i=1}^{5} n_i} \qquad (9)$$

The quality value measure:

$$m_2 = \frac{\sum_{i=1}^{5} n_i \times i}{\sum_{i=1}^{5} n_i} \quad (10)$$

where $n_1$, $n_2$, $n_3$, $n_4$, and $n_5$ are the number of recommended web pages with a score of 1, 2, 3, 4, and 5, respectively.

### C. Experimental Results and Discussions

In the experiments, these five testers evaluate the system by using top-weighted key-phrases under different Top-N recommendations (i.e., Top-5, Top-10, and Top-15). Besides, the threshold γ is empirically set to 0.7 so that the system can judge whether a pair of concepts are close enough.

Here, only the results of using 10 top-weighted key-phrases under different Top-15 recommendations are shown in Fig. 5 and Fig. 6. We found that the acceptable percentages of most domains are more than 80% and the quality values of most domains are more than 3.5. This means that our system works well on various domains.

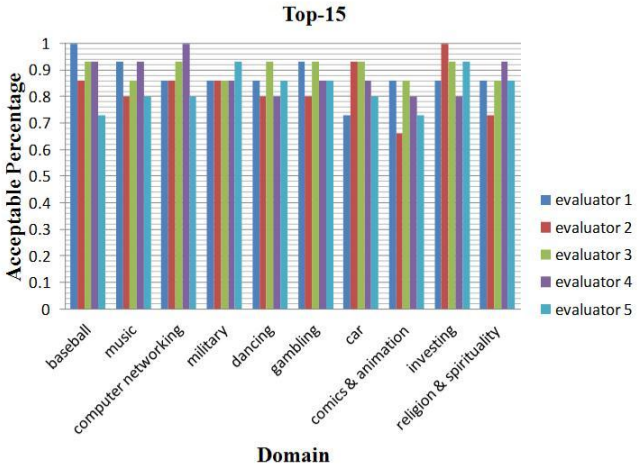


Fig. 5. Acceptable percentages of all domains for Top-15.

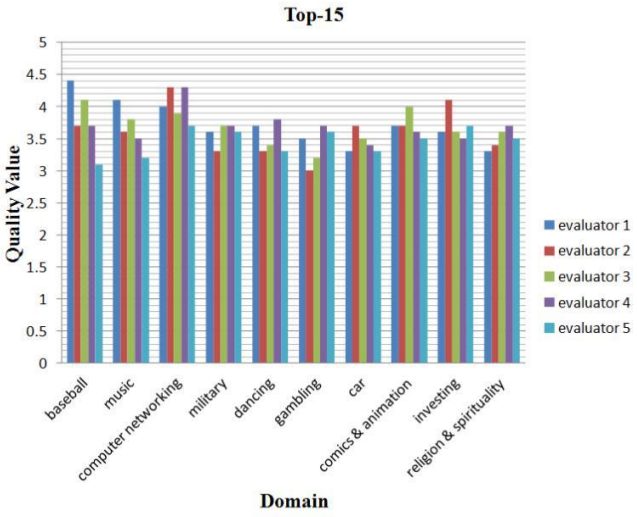


Fig. 6. Quality values of all domains for Top-15.

For each evaluator evaluating our method as shown in Fig. 7 and Fig. 8, we found that the average acceptable percentage is more than 83% and the average quality value is more than 3.4. All five evaluators are very satisfied with their own lists of the recommended web pages. Besides, we also compare our method with the method only using personal knowledge bases. The results indicate that our method always performs better than the method only using personal knowledge bases. This verifies that the navigation patterns extracted from the discovered concepts definitely facilitate recommending web pages.

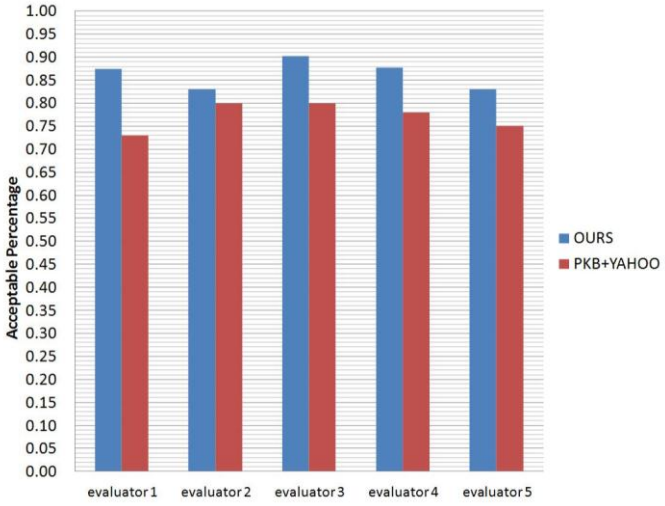


Fig. 7. Average acceptable percentage for each evaluator.

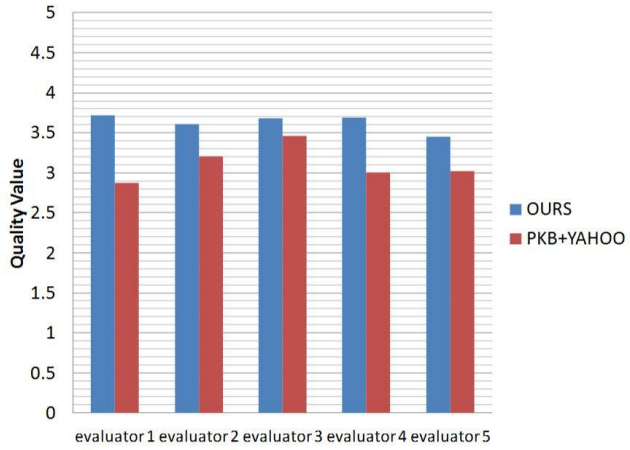


Fig. 8. Average quality value for each evaluator.

## V. Conclusions

In this paper, we propose a web recommendation system where user navigational patterns can be discovered from web logs. These navigational patterns are then used to generate recommendation web pages by matching the navigation behavior of a user personal knowledge base. The pages in a recommendation list are ranked according to their hub scores which are computed based on page connectivity information. The experimental results show that the web pages recommended by our system are of better quality and acceptable for humans from various domains, based on human evaluators ranking as well as quality-value-based performance measures.

## REFERENCES


[1] D. He and A. Goker, "Detecting session boundaries from web user logs," in *Proc. the 22nd BCS-IRSG Annual Colloquium on Information Retrieval Research*, 2000, pp. 57-66.

[2] S. M. Beitzel, E. C. Jensen, D. D. Lewis, A. Chowdhury, and O. Frieder, "Automatic classification of web queries using very large unlabeled query logs," *ACM Transactions on Information Systems*, vol. 25, no. 2, article 9, 2007.

[3] D. Shen, J. T. Sun, Q. Yang, and Z. Chen, "Building bridges for web query classification," in *Proc. the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2006, pp. 131-138.

[4] J. R. Wen, J. Y. Nie, and H. J. Zhang, "Query clustering using user logs," *ACM Transactions on Information Systems*, vol. 20, pp. 59-81, 2002.

[5] H. J. Zimmermann, *Fuzzy Set Theory and Its Applications*, Kluwer Academic Publishers, 2001.

[6] A. Banerjee, S. Merugu, I. S. Dhillon, and J. Ghosh, "Clustering with Bregman divergences," *Journal of Machine Learning Research*, vol. 6, pp. 1705-1749, 2005.

[7] C. Ding, X. He, P. Husbands, H. Zha, and H. D. Simon, "PageRank, HITS and a unified framework for link analysis," in *Proc. the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2002, pp. 353-354.

[8] S. Davies, "Still building the memex," *Communications of the ACM*, vol. 54, pp. 80-88, 2011.

[9] Y. F. Huang and C. S. Ciou, "Constructing personal knowledge base: automatic key-phrase extraction from multiple-domain web pages," *Lecture Notes in Computer Science*, vol. 7104, 2012, pp. 65-76.

[10] The Open Directory Project. [Online]. Available: http://www.dmoz.org/

[11] J. C. Dunn, "A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters," *Journal of Cybernetics*, vol. 3, pp. 32-57, 1973.

[12] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*, Kluwer Academic Publishers Norwell, 1981.

[13] P. M. Kanade and L. O. Hall, "Fuzzy ants as a clustering concept," in *Proc. the 22th International Conference of North American Fuzzy Information Processing Society*, 2003, pp. 227-232.

[14] J. M. Kleinberg, "Authoritative sources in a hyperlinked environment," *Journal of ACM*, vol. 46, pp. 604-632, 1999.

[15] Y. Zhang, E. Milios, and N. Zincir-Heywood, "Narrative text classification for automatic key phrase extraction in web document corpora," in *Proc. 7th Annual ACM International Workshop on Web Information and Data Management*, 2005, pp. 51-58.


**Yin-Fu Huang** received the B.S. degree in computer science from National Chiao-Tung University in 1979, and the M.S. and Ph.D. degrees in computer science from National Tsing-Hua University in 1984 and 1988, respectively. He is currently a professor in the Department of Computer Science and Information Engineering, National Yunlin University of Science and Technology. Between July 1988 and July 1992, he was with Chung Shan Institute of Science and Technology as an assistant researcher. His research interests include database systems, multimedia systems, data mining, mobile computing, and bioinformatics.

**Jia-Tang Jhang** received his B.S. degree in computer sciences from National United University and M.S. degree in computer sciences from National Yunlin University of Science and Technology in 2010 and 2012, respectively. He is currently serving in Tornado Technologies Co., Ltd. His major areas of interests are database systems and data mining.