Cross-Domain Query Navigation System for Touchscreens by Exploiting Social Search History

Ryo Shimaoka
Faculty of Policy Management Studies
Keio University
5322 Endo, Fujisawa, Kanagawa 252-0882, Japan
s09411rs@sfc.keio.ac.jp

Shuichi Kurabayashi
Faculty of Environment and Information
Keio University
5322 Endo, Fujisawa, Kanagawa 252-0882, Japan
kurabaya@sfc.keio.ac.jp

***Abstract*—Tablets and smartphones have gained immense popularity in recent times, and it is envisaged that they will increasingly be the devices of choice for users accessing the Internet. However, the user interface of conventional Web search engines, which employ keywords that require many taps by the user, are unsuitable for mobile terminals, which are normally equipped with touchscreens. We propose a cross-domain query navigation system that reduces the taps required for inputting queries by providing a content-dependent *word map* that presents the relevance between keywords. This *word map* presents keywords that enable both narrowing action, whereby users append a new keyword to specify the context of a query, and sliding action, whereby users replace a keyword to change the query context. The *word map* is unique in that it recommends queries for narrowing and sliding transitions by computing these two types of directional relevance between input keyword and another keyword in the log. The system is applicable to the existing query logs of search engines, social networking services, and users' browsers, enabling users to control the term recommendation by selecting the logs to be analyzed. The recommendation may be a commonly recognized relevant term from the global query logs of search engines or a personalized term from the user's browser history.**

Keywords-Query Navigation; Personalization; User Interface; Collective Intelligence; Web Search Engine.

I. INTRODUCTION

The recent years have witnessed a rapid rise in the popularity of tablet devices and smartphones, and concomitantly, a widespread increase in the use of the touch-based user interface (UI). Statistics published by Cisco indicate that the global mobile data traffic grew 2.6-fold in 2010, nearly tripling for the third consecutive year [1]. In addition, statistics published by Google Confidential and Proprietary suggest that by 2015, more than a quarter of the mobile traffic will be used for information retrieval and that the number of Internet users not using PC devices will increase to 788 million [2]. Hence, a major shift in use of Internet-connected devices, from PCs to mobile terminals, is currently underway.

A large portion of Internet activity is in the form of queries to search engines. However, many users have difficulty querying a search engine on a complex topic that encompasses several terms, such as "JavaScript and HTML5" or "ActionScript and API," relating to a subject with which they are not familiar. Mobile devices present an additional difficulty: although touchscreens are generally very convenient for other functions, they are not very convenient to use as a typing tool. In particular, queries in Chinese-Japanese-Korean-Vietnamese (CJKV) languages present special difficulties because each CJKV character requires two or three input strokes. In mobile devices, predictive input methods are the predominant method for supporting the input of long sentences and terms. These predictive input methods recommend terms and sentences that can be concatenated to the user's input character sequence. Another conventional method is a keyword suggestion approach, such as Google Suggest. When a user inputs an initial query term, this method suggests related terms by calculating inter-term relevance, exploiting the search engine's query log to recognize the relevant terms.

However, these conventional methods are based on the co-occurrence probability and hence are unsuitable for inputting queries that consists of several cross-domain terms, such as "climbing health care costs." In such cases, predictive input methods may not correctly recommend the next search term, and a cross-domain term-relevance calculation is required. On the other hand, Google Suggest will not tailor the search results to an individual user's interests, because it uses standardized search terms drawn from a universal users' log. Thus, the UIs provided by a conventional Web search engine require users to tap many times, making these UIs unsuitable for mobile terminals.

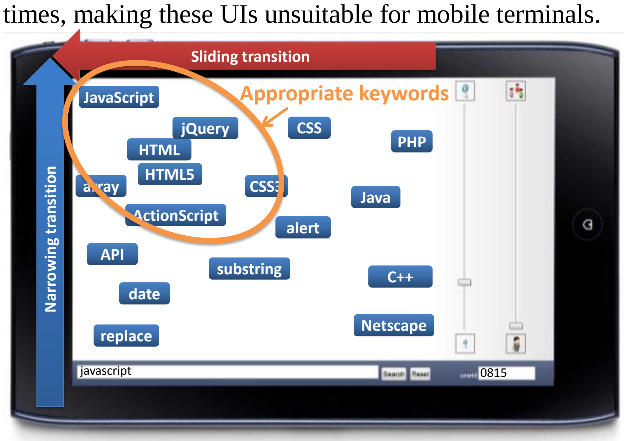


Figure 1. User Interface of a Cross-Domain Query Navigation System.

This paper proposes a cross-domain query navigation system that assists in the input of multiple queries by show

Copyright (c) IARIA, 2012. ISBN: 978-1-61208-200-4

a content-dependent *word map* to present the relevance between keywords. This system allows users to input keywords by selecting an appropriate keyword in a convenient manner, because the word map shows the next coordinate instantly, as shown in Figure 1. Here, we explain an example scenario of query navigation shown in Figure 2. This figure shows the following two types of navigation:

- Narrowing navigation: Users append a new keyword (e.g., “traffic,” or “global”) to specify the context of a query. The appended keyword is at a lower level of abstraction than those of the existing keywords.
- Sliding navigation: Users replace a keyword to change the context of a query. Here, a user removes an existing keyword (e.g., “traffic”) that is not within the scope of the current topic of interest and inserts another one (e.g., “user”) that is relevant to the current topic of interest, thus shifting the focus of the query. In this case, the system recommends a new keyword (e.g., “laptop”) appropriate in the current context.

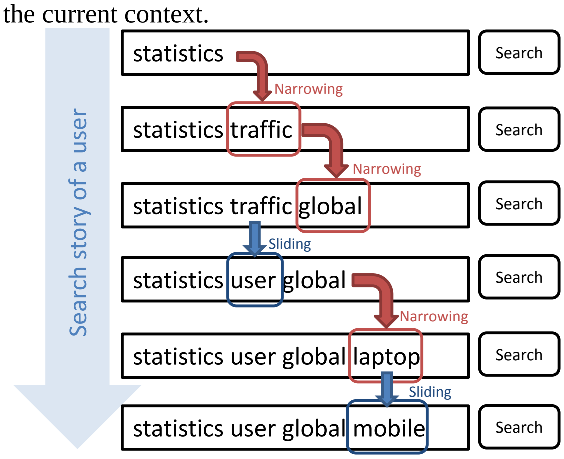


Figure 2. Narrowing and Sliding Transitions in Query Construction.

The advantage of this system is that it obviates the need for users to enter the subsequent search terms themselves; instead, they are able to select from among those that are mapped on the screen. This search story approach makes it possible to reduce taps. For example, when a user wishes to add the search terms “global,” “mobile,” and traffic” to the term “statistics,” which has already been inserted in the search box, only one tap is required for each term, making three in all. The keywords are presented after considering the user’s browser history, which enables personalization, and the other users’ querying history, which supports a user by exploiting collective intelligence. Our system configures the balancing between personalization and collective intelligence support dynamically, that is not possible with conventional search engines.

Another advantage of the system is afforded by the fact that it is also applicable to the search stories of social networks, which include groups of experts in various fields, as shown in Figure 3. This allows users to search within a domain that they are not familiar with, by drawing on the collective knowledge and experience of expert groups through their search stories. Furthermore, the application would also help users construct a query in a language that they do not know well.

II. Related Work

The query expansion method is a well-known means of helping a search engine’s users to input complex queries. The traditional example of query expansions is Google Suggest, which recommends keywords from a uniform set that is derived from all users based on the number of previous searches. Currently, many researchers are focusing on personalization mechanisms in query expansion [3]. For example, Teevan, et al. [4] proposes a personalization method that considers the user’s specific interests by constructing a user profile from the relevance feedback in a ranking. Gauch, et al. [5] proposes an implicit personalization mechanism that generates ontology-based user profiles without user feedback, by monitoring the user’s browsing activities.

An alternative method of query expansion uses the concept of *community*. Smyth, et al. [6] introduces the collaborative filtering method, which exploits a similar relationship between queries and result pages for each community. The method expands a query by referring to a graded mapping between users and items.

The most significant difference between our approach and the conventional ones listed above is the concept of *query dimension*. Our system provides two *dimensions* in the query building process: narrowing and sliding. Narrowing is a typical query building method that allows users to increase the specificity of a query after starting with an initial keyword. Our approach also supports sliding, which suggests cross-domain keywords by computing the implicit relevance of keywords in different domains, such as “climbing,” “health,” “care,” and “costs.” To increase the precision of the sliding process, our approach exploits the search story of a relevant group or community.

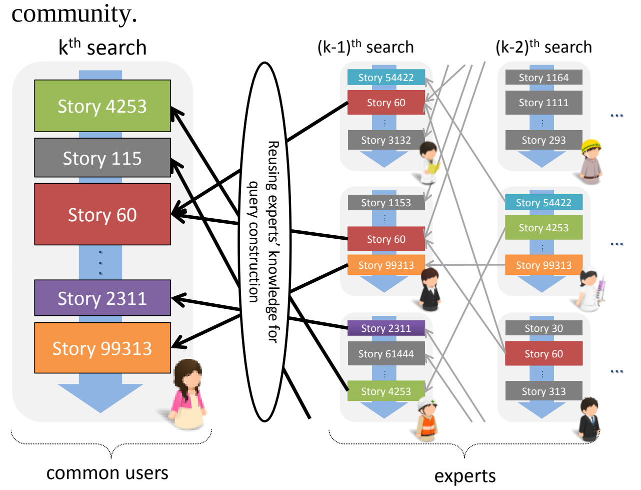


Figure 3. Search Story Sharing among Users Empowers the System’s Cross-Domain Keyword Recommendation.

Copyright (c) IARIA, 2012.
ISBN: 978-1-61208-200-4

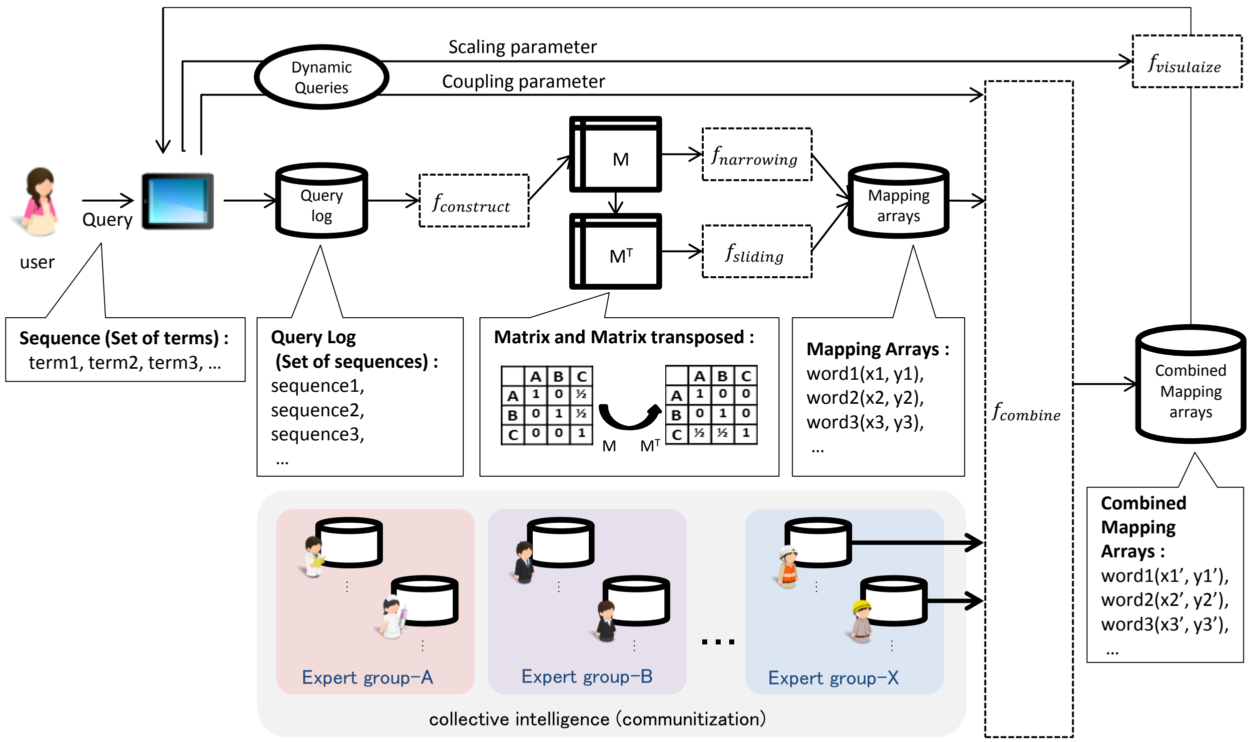


Figure 4. System Architecture of Recommendation of Search Terms on Word Map for Directional Relevance between Input Keyword and Another Keyword in the Log.

III. APPROACH

The narrowing and sliding forms of navigation are based on an *inter-term relationship matrix* constructed from a query log, as shown in Figure 4; the purpose of this matrix is to record the relationship between keywords for each user. The system converts the matrix into recommendation scores, which correspond to the coordinate values for narrowing and sliding as presented on the user interface. The system combines the recommendation scores from the user and from social groups within the domain of interest.

The first stage is for the system to construct the matrix from the query log, a set of keyword sequences recorded when the user inputs a complex query in a search box. This matrix contains scores representing relationships between search terms. It is updated from the query log. In the second stage, the system converts the matrix into two types of recommendation scores: sliding and narrowing. The system calculates these scores based on computation of the inner product of the matrix and the matrix transpose. The final stage is to combine the recommendation scores of the user with that of social groups within the domain of interest. Our concept of social-network-based relevance computation is reusing third parties' knowledge about query construction. This system may distinguish several groups of users by using other SNS's social graph, such as Twitter's follower/followee structure and Facebook's friend structure. The user can also adjust the parameters of the combination process.

A. *Data Structure*

The data structure in this system consists of two data elements—*query log* and *inter-term relationship matrix*—which are now explained in detail as follows.

1) *Query Log*

A query log is a set of sequences that consist of search terms. We define *Log (L)* as a data structure that is determined based on a *Sequence (S)* of keywords inputted by a user. L_i of user *i* is defined by the following equation.

$$L_i := \langle S_0, S_1, \cdots, S_n \rangle$$

where *n* is the number of sequences.

A sequence is a set of searched keywords. Therefore, we define a *Sequence (S)* as a data structure that is determined based on the *keyword (k)*. S_j is defined by the following equation.

$$S_j := \langle k_0, k_1, \cdots, k_n \rangle$$

where *n* is the number of keywords.

2) *Inter-Term Relationship Matrix*

A function generates a relationship matrix from the query log. The relationship matrix contains a set of values that represent the directional relevance between each pair of keywords (the *weight* of the association). This is a square matrix, whose rows and columns each correspond to the same set of keywords. We define the *Matrix (M)* of user i as a data structure that is determined based on the *weight (w)*.

$$M_i := \begin{bmatrix} w_{[0,0]} & \cdots & w_{[n,0]} \\ \vdots & \ddots & \vdots \\ w_{[0,n]} & \cdots & w_{[n.n]} \end{bmatrix}$$

where n is the number of keywords. The system also generates the matrix transpose ${M_i}^T$ for reverse look-up.

B. Functions

The proposed system provides three main functions. The first function constructs the relationship matrix from a query log. The second function converts the matrix into narrowing and sliding scores for recommendations. The final function combines the recommendation scores of the user with those of a social group can provide expertise concerning the user's domains of interest.

1) Constructing a Matrix from a Query Log

The system provides a fundamental function to construct a matrix from a query log. The function is defined as follows.

$$f_{construct}(L_i) \rightarrow M_i$$

where M_i contains a set of values $w_{[l,\, m]}$ that represent the weight of the directional relevance between k_l and k_m.

This function updates the matrix every time the user inputs a query. Thus, we set the *weight (w)* of a sequence (S_j) as the relevance, determined based on the *rank* of *keyword (k)*.

$$w_{[l,m]}(S_j) \rightarrow \left[\frac{1}{rank(k_0 \in S_j)}, \frac{1}{rank(k_1 \in S_j)}, \cdots, \frac{1}{rank(k_n \in S_j)}\right].$$

Figure 5 shows an example of this summation process.

2) Converting a Matrix into Recommendation Scores

The system provides a fundamental function to convert a matrix into mapping arrays. Each mapping array contains the vertical and horizontal scores of a given keyword in relation to the *origin* keyword, i.e., the last term of a query. Thus, the function f_{map} generates two directional relevance scores, such as sliding relevance and narrowing relevance, according to a keyword specified as the origin point. $f_{map}(o)$ is defined as follows:

$$f_{map}(M_i, o) \rightarrow \{\langle p_v, p_h\rangle_0, \ldots, \langle p_v, p_h\rangle_n\},$$

$$\langle p_v, p_h\rangle_k \rightarrow \langle \sum_{j=0}^{n} M_{i[o,j]} \cdot M_{i[j,k]}, \sum_{j=0}^{n} {M_i}^T{}_{[o,j]} \cdot {M_i}^T{}_{[j,k]} \rangle$$

where p_v and p_h are the vertical and horizontal scores, respectively, for the word map. The vertical score corresponds to the directional relevance of a narrowing search, whereas the horizontal score corresponds to the directional relevance of a sliding search.

3) Combining the Recommendation Scores of the User and the Expert Groups

This system uses the collective expertise of other users for its recommendations. This recommendation function merges the matrix of a user and the matrices of other search engine users in a weighted combination. The combination weighting, or *rate*, is set by the user of this system via a slider on the Web page. Thus, we define $f_{combine}$ as a function that is determined based on a *combination rate (r)*.

$$f_{combine}(p, G, r) \rightarrow [\frac{p \cdot (100 - \mathrm{r}) + \frac{\sum_{j=0}^{n} G}{n} \cdot \mathrm{r}}{100}, \cdots]$$

where p is a correlation score and n is number of persons in *group (G)*.

These equations combine the matrix of the main user with the average of the matrices of all users to yield a final score.

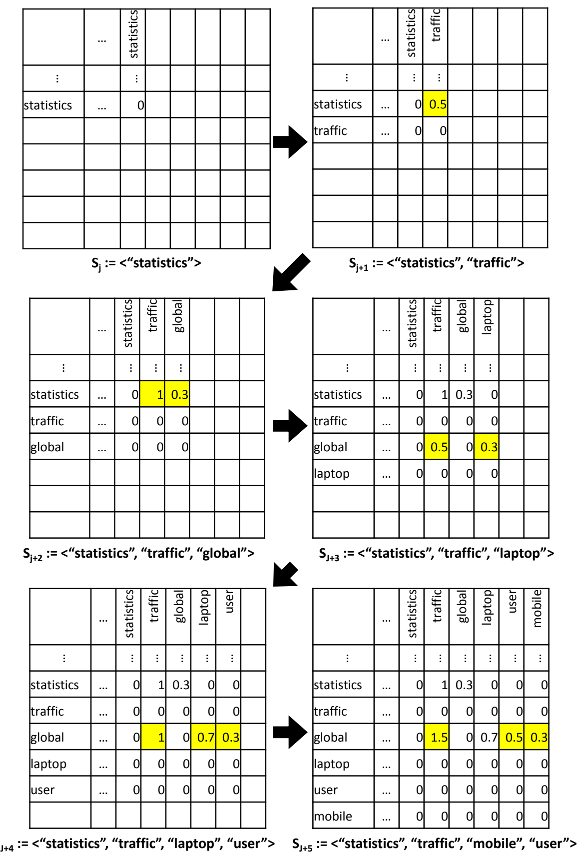


Figure 5. Matrix Composition Process.

IV. IMPLEMENTATION

We have implemented a prototype system for evaluating the recommendation of search terms by analyzing users' query logs. This system has been coded in full-stack JavaScript language, which implies that the server-side and client-side modules are implemented in JavaScript only.

1) Modules

The engine of this system has two main modules for the server side and client side. These modules use the same data structure, which is a user's search story, but they serve two different functions. On the server side, the system provides communitization, whereas on the client side, it provides personalization.

The server-side module of the prototype system outputs four types of arrays: narrowing and sliding scores for both the user and the community. The advantage of these outputs is that the system is able to present search terms with just the client-side module. Therefore, this module is only run when the user inputs a new query.

The client-side module presents search terms on the user interface. The system presents the candidate search terms in a two-dimensional space that is defined by the narrowing axis and the sliding axis. The search terms are positioned in relation to parameters that the user inputs using two sliders, one of which defines the combination rate for other search stories (community) and the other defines the scaling rate (zoom factor) for words.

2) User Interface

The user interface of this system consists of the word map, zoom slider, social slider, search box, and search button (Figure 6). The most important control is the social slider. This slider defines the extent to which the user's search story is combined with the community search stories. The system allows users to discover appropriate keywords by adjusting the combination level of search terms, if no terms are found initially.

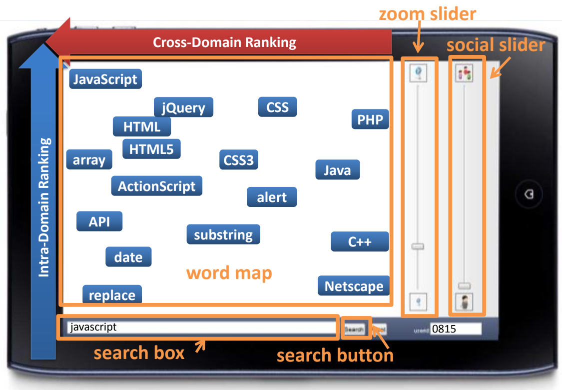


Figure 6. User Interface of Cross-Domain Query Navigation System.

The procedure of this system is as follows:

Step 1: The user inputs the initial keyword of the query in the search box, and the system presents keywords on the word map.

Step 2: The user taps an appropriate keyword. The system displays the keyword in the search box and presents a new set of keywords on the word map (Figure 6 shows only one term in the search box. The system displays the next keyword when the user enters it via the touch interface).

Step 3: If no appropriate keyword is shown, the user may drag the social slider until the combination level generates a satisfactory range of keywords.

Step 4: The user repeats Steps 2 and 3 as necessary. The user taps the search button and the system retrieves the search results.

Figure 7 shows an example of how the word map can be changed using the social slider. The arrows in this figure show how the keywords move when the slider is operated. The origin point (upper left) corresponds to the initial query "JavaScript". The system provides candidate keywords from an expert group of Web designers but not programmers, displaying new candidate keywords that are used by Web designers, such as "sample code," "Web design," and "Flash," but not keywords that are used by programmers, such as "Java" and "C++." The figure shows that the keywords that are more often used by Web designers than by programmers, such as "HTML5," are slightly shifted from lower right to upper left. The figure shows that the keywords that are more often used by programmers than by Web designers, such as "API," are shift from upper left to lower right.

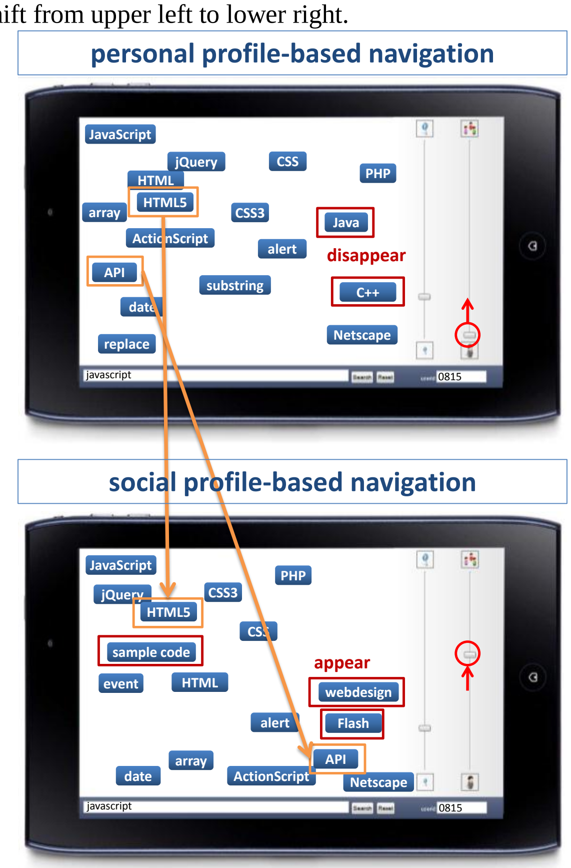


Figure 7. Change in the Keyword Positions on Word Map with Social Slider.

V. EXPERIMENTS

This section presents experimental studies that clarify the effectiveness of our approach. In particular, this experiment evaluates the f_{map} function that converts a matrix into recommendation scores. The experiment evaluates the directional relevance between the input search terms and the candidate search terms, based on the user and the community query logs. The experiment compares the narrowing, which is the legacy keyword recommendation, and the sliding, which is the original feature of this system. Due to the limitation of the space, this experiment clarifies that our approach calculates the appropriate distance between query keywords by asking 10 test subjects to evaluate the navigation results.

A. Overview

This experiment measures the precision of the keyword ranking of the recommended query keywords by comparing the manually-conducted correct result set and our system. We have set up the inter-term relationship

Copyright (c) IARIA, 2012. ISBN: 978-1-61208-200-4

matrix by submitting 952 queries to Google. As a result, we obtained a 108 × 108 matrix. We have designed the three test topics, “design,” “e-book,” and “editorial,” as the initial keywords. These three test cases generate three rankings. We select the top 10 keywords of narrowing / sliding relevancy in each search topics, such as “design,” “e-book,” and “editorial,” as shown in TABLE I.

In order to verify the precision of the results, we had 10 persons evaluate the rankings. 10 test subjects evaluated relevance of 60 keywords and three topics from the viewpoint of *narrowing* and *sliding*. Every subject rated each recommended keyword according to the following 5-point scheme: 0 (completely not relevant), 1 (not relevant), 2 (slightly relevant), 3 (relevant), and 4 (very relevant). We considered the ideal ranking as the average of 10 results.

Table I. NARROWING AND SLIDING KEYWORDS AND THE RANKS.

design			e-book			editorial		
rank	narrowing	sliding	rank	narrowing	sliding	rank	narrowing	sliding
1	editorial	e-book	1	design	editorial	1	color	e-book
2	layout	editorial	2	editorial	research	2	design	design
3	color	research	3	color	implication	3	e-book	research
4	image	history	4	layout	history	4	layout	implication
5	scheme	magazine	5	image	program	5	magazine	hisitory
6	research	book	6	scheme	genre	6	newspaper	program
7	ranking	program	7	research	magazine	7	history	genre
8	magazine	genre	8	ranking	book	8	electronic	magazine
9	newspaper	retrieval	9	search	retrieval	9	television	ranking
10	iPhone	iTV	10	engine	brief	10	iPhone	brief

B. Evaluation Result

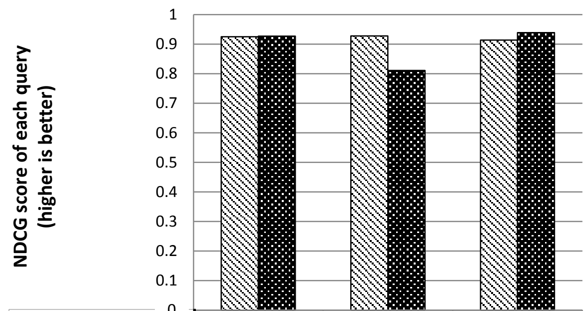
This experiment compares the keyword ranking of narrowing and sliding recommendation with evaluations by test subjects. The evaluation of this experiment applies normalized discounted cumulative gain (NDCG).

$$\mathrm{DCG} = \sum_{\mathrm{i}=1}^{10} \frac{rel_i}{log_2 i},\ \mathrm{IDCG} = \sum_{\mathrm{i}=1}^{10} \frac{rel'_i}{log_2 i},\ \mathrm{NDCG} = \frac{DCG}{IDCG}$$

where rel_i is the average evaluation score given by the test subjects, and $\mathrm{rel'}_\mathrm{i}$ is the average scores in descending order. Figure 8 shows NDCG of narrowing and sliding recommendation for three topics (“design,” “e-book,” and “editorial”). Higher score means a better retrieval precision. The most important result is a score of *sliding recommendation* because the narrowing recommendation is close to the conventional query recommendation method. The NDCG of sliding recommendation is the almost same as that of narrowing recommendation. This result implies that our sliding recommendation achieves highly practical precision, although the sliding recommendation generates different keywords from the narrowing recommendation. By using this system, the user received the precise query keyword, which shared a cross-domain relationship with the initial keyword. This recommendation is a very powerful tool to input a complex query consisting of cross-domain keywords.

VI. CONCLUSION AND FUTURE WORKS

We have proposed the complex query navigation system that exploits search stories of social groups. This system recommends candidates for a next search term by calculating the directional relevance along two conceptual dimensions and performing narrowing and sliding operations. A social combination function enables the user to utilize the knowledge of social groups to facilitate navigation. We implemented a prototype system that is able to retrieve and present candidate keywords for multiple queries while reducing the number of taps required. As a future work, we plan to develop a social-based query recommendation mechanism and to evaluate scalability in a complex query navigation in multiple domains.



Query	Q1: design	Q2: e-book	Q3: editorial
narrowing (conventional method: single domain recommendation)	0.925012297	0.927528221	0.913642768
sliding (our method: cross-domain recommendation)	0.926944072	0.811268464	0.939018707

Figure 8. NDCG of Narrowing and Sliding Recommendation.

REFERENCES

[1] Cisco. “Global Mobile Data Traffic Forecast Update, 2010-2015” - Cisco Visual Networking Index, February 1, 2011, http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-520862.pdf . [retrieved: 5, 2012]

[2] Google. “Admod - Tablet Survey,” March 2011, http://www.ccapitalia.net/descarga/docs/2011-AdMob-TabletSurvey.pdf .

[3] A. Micarelli, F. Gasparetti, F. Sciarrone, and S. Gauch, “Personalized search on the World Wide Web,” in The Adaptive Web, LNCS 4321/2007, 2007, vol. 4321, pp. 195-230.

[4] J. Teevan, S. T. Dumais, and E. Horvitz, “Personalizing search via automated analysis of interests and activities,” in Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval - SIGIR ’05, 2005, p. 449.

[5] S. Gauch, J. Chaffee, and A. Pretschner, “Ontology-based personalized search and browsing,” in Web Intelligence and Agent Systems - IOS Press vol. 1, no. 3-4/2003, pp. 219-234.

[6] 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 Modeling and User-Adapted Interaction, Apr. 2005, vol. 14, no. 5, pp. 383-423.