A Cross-Domain Query Navigation and Visualization System for Touchscreens that Exploits Social Search History

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

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

Abstract—**Tablets and smartphones have become the most popular Internet client system for end-users. However, conventional web search engines employ Hyper Text Markup Language (HTML) text input systems that require many individual characters. These are unsuitable for mobile terminals, which are normally equipped with touchscreens. We propose a cross-domain query navigation and visualization system that assists the query input process by providing a context-dependent *word map* that presents the relevance between keywords. This *word map* enables both a narrowing action, whereby users append a new keyword to specify the context of a query, and a sliding action, whereby users replace a keyword to change the query context. The *word map* is unique in that it recommends queries for both narrowing and sliding transitions by computing the directional relevance between the input keyword and another keyword in the social knowledge base. The system is applicable to existing search engine query logs, social networking services, and web browsers, enabling users to control the term recommendations by selecting the logs to be analyzed.**

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

I. INTRODUCTION

This paper describes a cross-domain query navigation and visualization system [1] and its implementation framework towards modern smart devices, such as smartphones and tablets. The name of our system is *Query Map (Q-MAP)*, and it assists in the input of multiple queries for web search engines by showing a context-dependent *word map* to present the relevance between keywords by considering the current user's input. We implement this system on top of an existing web search engine by utilizing modern HTML5 technologies.

Recent years have witnessed a rapid rise in the popularity of tablet devices and smartphones, and concomitantly, a widespread increase in the use of touch-based user interfaces (UIs). Statistics published by Cisco indicate that global mobile data traffic grew 2.6-fold in 2010, nearly tripling for the third consecutive year [2]. In addition, statistics published by Google Confidential and Proprietary suggest that, by 2015, more than a quarter of mobile traffic will be used for information retrieval and the number of Internet users not using personal computers (PCs) will increase to 788 million [3]. Hence, a major shift in the type 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 Application Program Interface (API)," relating to a subject with which they are not familiar. Mobile devices present an additional difficulty: although touchscreens are generally very convenient, they are not particularly well adapted for use as a typing tool. In particular, queries in Chinese, Japanese, Korean, and Vietnamese (CJKV) present special difficulties because each CJKV character requires two or three input strokes. In mobile devices, predictive methods are the predominant means of 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 the inter-term relevance, exploiting the search engine's query log to recognize the relevant terms. However, these conventional methods are based on co-occurrence probabilities, and are thus unsuitable for inputting queries that consist of several cross-domain terms, such as "climbing healthcare costs." In such cases, predictive input methods may not correctly recommend the next search term, and a cross-domain term-relevance calculation is required. Thus, the UIs provided by conventional web search engines require users to tap the keyboard or screen many times, making them unsuitable for mobile terminals.

This paper proposes a cross-domain query navigation system that assists in the input of multiple queries by forming a content-dependent *word map* to present the relevance between keywords. This system allows users to select appropriate keywords in a convenient manner, because the word map shows the next coordinate instantly, as shown in Figure 1. In this paper, we extend the system proposed in [1] to support a large-scale dataset and perform practical experiments to evaluate our model. In addition, this paper shows a new prototype system implementation running on the iPad and Android smart devices. This implementation utilizes modern touchscreens as a user interface for inputting queries with small number of touch operations.

This *word map* approach makes it possible to reduce the number of keyboard or screen 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 other users' querying history, which supports users by exploiting collective intelligence. Our system configures the balance between personalization and collective intelligence support dynamically, which is not possible with conventional search engines. This configuration mechanism can be applied to protect users' privacy by setting all the query logs to be stored in a client local storage.

Another advantage of our system is its applicability to the search histories of social networks, which include groups of

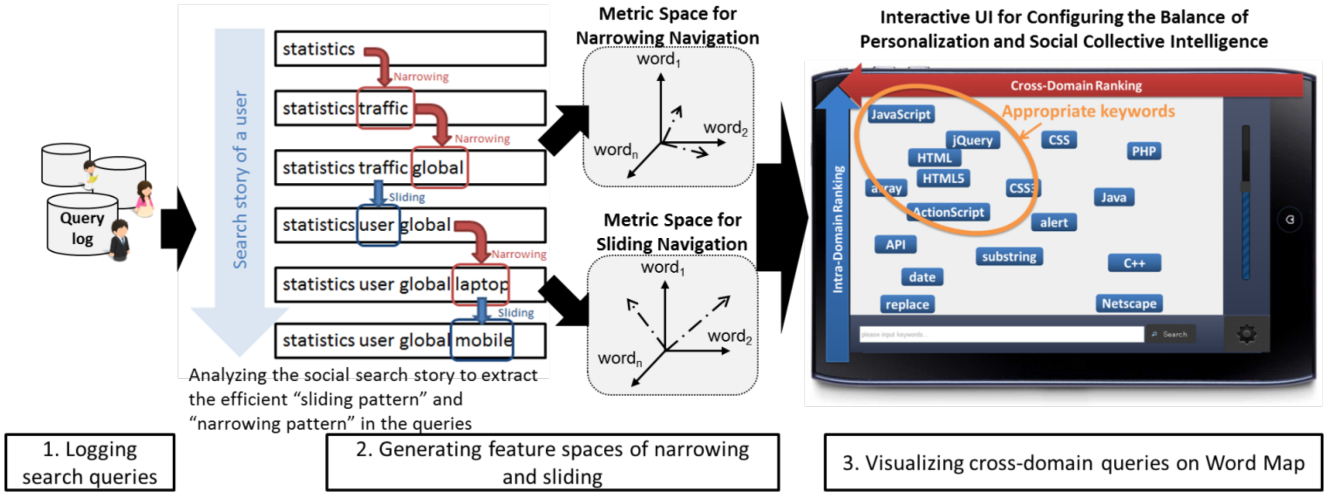


Figure 1. User Interface of a Cross-Domain Query Navigation System.

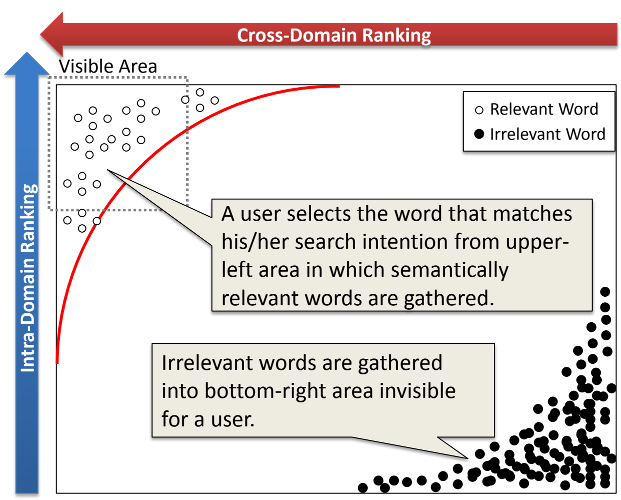


Figure 2. Visual Metric Space for Selecting Relevant Words.

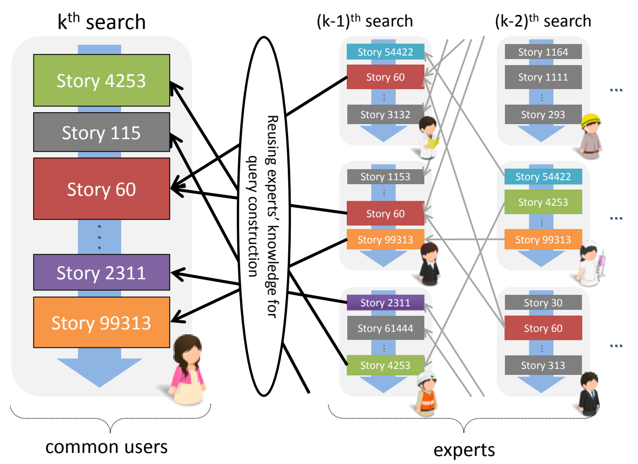


Figure 3. Search Story Sharing among Users Empowers the System's Cross-Domain Keyword Recommendation.

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 can also help users to construct a query in a language that they do not know well.

The remainder of the paper is structured as follows. Section II demonstrates several advantages of our system. In Section III, we discuss several related studies. and Section IV presents an architectural overview of our system. Section V introduces the prototype system, which is then evaluated in Section VI. Finally, Section VII gives our concluding remarks and some ideas for future work.

II. Advantages of Cross-Domain Query Navigation

Here, we explain the example scenario of query navigation shown in the center of Figure 1. This figure shows the following types of navigation:

- Narrowing: 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: Users replace a keyword to change the context of a query. Here, the 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") as being appropriate in the current context.

The advantage of this system is that it obviates the need for users to enter subsequent search terms themselves; instead, they are able to select from among those that are mapped on the screen. Figure 2 shows a metric space for visualizing the narrowing and sliding navigation methods. Our system computes the semantic distance between words for each navigation type and allocates each word to the two-dimensional map according to the calculated distance. The most important factor of this visualization is that it eliminates irrelevant words from the visible upper-left area, because the system orders the retrieved words by using the correlation scores between each word. The correlation-based distance enables us to locate irrelevant words into bottom-right corner. This combination mechanism of visualization and correlation computation is our unique approach. The user is able

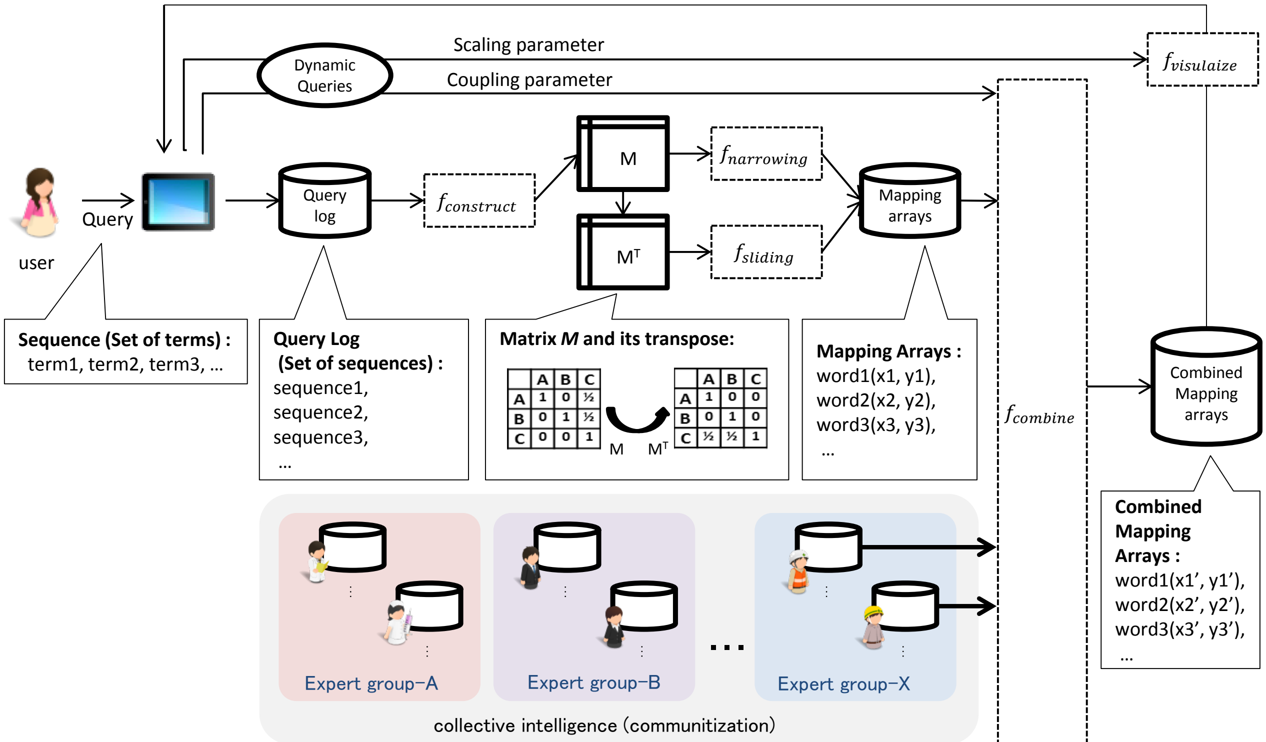


Figure 4. System Architecture for the Recommendation of Search Terms on Word Map for Directional Relevance between Input Keyword and Another Keyword in the Log.

to search something for making trial and error while watching the change of search results. The horizontal axis corresponds to the cross-domain ranking where the keywords are ordered according to the sliding navigation. The vertical axis corresponds to the intra-domain ranking where the keywords are ordered according to the narrowing navigation. Users can select the words that match their search intention from the upper-left area in which semantically relevant words are clustered. In contrast, irrelevant words are mapped to the bottom-right.

This system utilizes the query logs, which are submitted by experts, as a knowledge base for providing search navigations. This application reuses the experts' query as a successful search story. For example, a novice user submits a query "asthma, how-to cure." By using the conventional search method, the novice user has difficulty finding the sufficient information. On the other hand, our system reuses the query logs, submitted by experts who know the filed well, to navigate a user to use "asthmatic remission".

Our navigation algorithm is independent of a search engine's relevance computation method such as PageRank. This is because our system calculates context-dependent relevance between query words by analyzing co-occurrence of those words in a query and temporal relevance between words in a sequence of queries submitted for a specific purpose. Thus, our system can be implemented as a meta-level system or a wrapper interface, which is installed in modern smart devices including smart TV and the latest gaming console, for the legacy search engines such as Google and Microsoft's Bing. This is highly advantageous for realizing a novel UI suitable for touchscreens because we do not have to re-implement or to modify the existing Web search engine infrastructure. It is valuable to mention that our system can be a special-purpose search UI for the specific site. For example, by applying our system to an online bookstore, our system will navigate users to input book titles, authors, and publisher's name.

III. Related Work

The query expansion method is a well-known means of helping a search engine's users to input complex queries [4]. The traditional example of query expansion is Google Suggest, which recommends keywords by analyzing query logs based on the number of previous searches. The significant difference between the conventional keyword suggesting system and our Q-MAP system is a smart visualization mechanism that integrates query logs analysis and intuitive noise reduction mechanism for eliminating irrelevant words from the visible area. Our system provides a novel UI designed for modern touchscreen devices. Kelly, Gyllstrom, and Bailey [5] proposed the combination method of term suggestion that helps users to add terms to their original query to clarify the semantics of the initial input. Specialization and parallel movement query suggestion (SParQS) [6] is a query suggestion system that provides two query reformulation methods: a specialization method to make the query more specific, and a parallel movement method to change an entity contained in the query. These approaches utilize a query log collected by a web search engine. Wen et al. [7] proposed a query clustering method based on content similarity for detecting frequently asked queries.

Currently, many researchers are focusing on personalization mechanisms in query expansion [8]. For example, Teevan et al. [9] proposed 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. [10] developed an

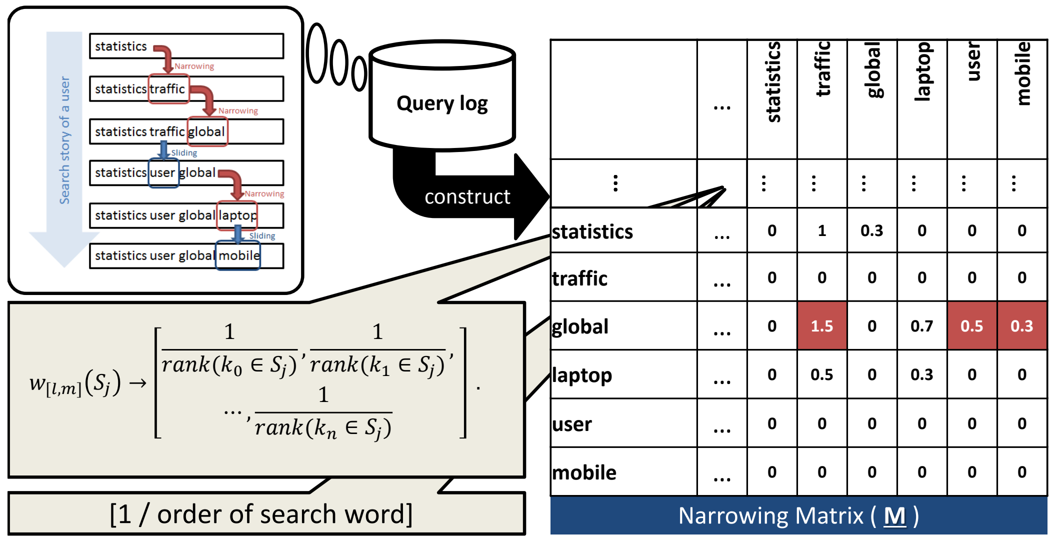


Figure 5. Composition Process of Narrowing Matrix.

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. [11] introduced the collaborative filtering method, which exploits a similar relationship between queries and results for each community. The method expands a query by referring to a graded mapping between users and items.

In the development of search UIs, many approaches have analyzed the user's search activities. For example, classified or faceted search results are well-known techniques for organizing expanded queries [12], [13].

The most significant difference between our approach and those listed above is context-dependent reuse of query stories shared in social networking service (SNS). Our system focuses on 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," "healthcare," and "costs." Unique feature of our system is a method to increase the precision of the sliding and narrowing query specification process by exploiting the search history of a relevant group or community.

IV. Cross-Domain Query Navigation 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 UI. The system combines the recommendation scores from the user with those from social groups within the domain of interest.

The first stage is for the system to construct the matrix from the query log, which is the 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 recommendation scores: one for sliding and the other for narrowing. The system calculates these scores based on the inner product of the matrix and its transpose. The final stage is to combine the recommendation score of the user with those of social groups within the domain of interest. Our concept of computing social network-based relevance is the reuse of third-party knowledge about query construction. This system may distinguish between several groups of users according to social graphs, such as Twitter's follower/followee structure and Facebook's friend structure. Users can also adjust the parameters of the combination process.

A. Data Structure

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

1) Query Log

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

$$L_i := \langle S_0, S_1, \cdots, S_n \rangle \tag{1}$$

where m is the number of sequences.

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

$$S_j := \langle k_0, k_1, \cdots, k_m \rangle \tag{2}$$

where n is the number of keywords.

2) Inter-Term Relationship Matrix

We generate a relationship matrix from the query log. The relationship matrix contains a set of values that represents 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 based on the *weight (w)*.

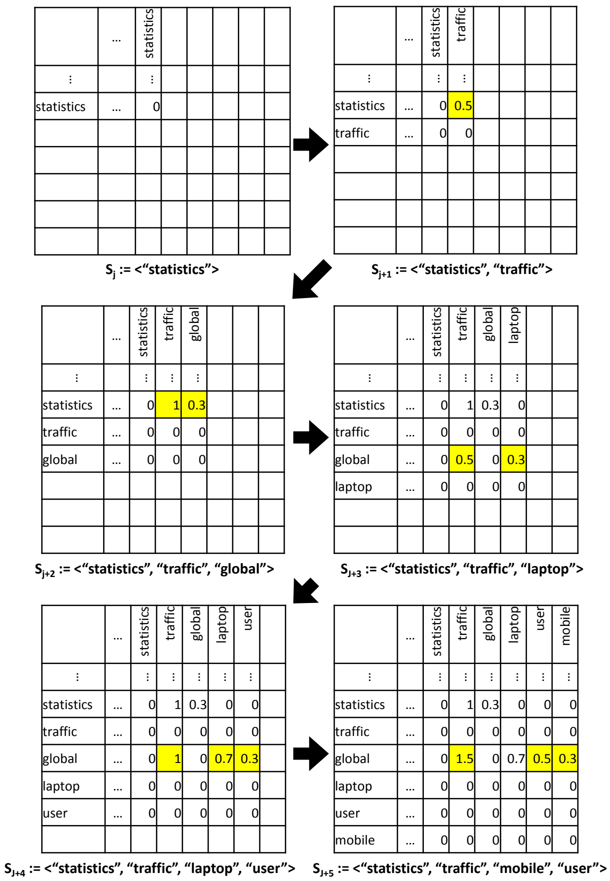


Figure 6. Example of Matrix Composition Process for Cross-Domain Query Navigation.

$$M_i := \begin{bmatrix} w_{[0,0]} & \cdots & w_{[q,0]} \\ \vdots & \ddots & \vdots \\ w_{[0,q]} & \cdots & w_{[q.q]} \end{bmatrix} \tag{3}$$

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

B. Primitive Functions

The proposed system provides three main functions. The first constructs the relationship matrix from a query log. The second 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 that 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:

$$f_{construct}(L_i) \rightarrow M_i \tag{4}$$

where M_i is a matrix for i-th user and contains a set of values $w_{[l,m]}$ representing the weights 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 sequence S_j as the relevance based on the *rank* of *keyword k*.

$$w_{[t,u]}(S_j) \rightarrow \left[\frac{1}{rank(k_0 \in S_j)}, \frac{1}{rank(k_1 \in S_j)}, \cdots, \frac{1}{rank(k_n \in S_j)}\right]. \tag{5}$$

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 (z)*, i.e., the last term of a query. Thus, the function f_{map} generates sliding and narrowing relevance scores according to a keyword specified as the origin point. We define $f_{map}(M_i, a)$ that inputs an origin word a and a user matrix M_i as follows:

$$f_{map}(M_i, z) \rightarrow \{\langle p_v, p_h\rangle_0, \ldots, \langle p_v, p_h\rangle_q\} \tag{6}$$

$$\langle p_v, p_h\rangle_k \rightarrow \langle \sum_{x=0}^{q} M_{i[a,x]} \cdot M_{i[x,k]}, \sum_{x=0}^{q} M_i{}^T{}_{[a,x]} \cdot M_i{}^T{}_{[x,k]} \rangle \tag{7}$$

where p_v and p_h are the vertical and horizontal scores, respectively, for the word map. Thus, $\langle p_v, p_h\rangle_k$ corresponds to relevance of k-th word. 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. $M_{i[g,j]}$ denotes the value at a point of g-th row and j-th column.

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 user matrix with those of other search engine users in a weighted combination. The system user sets the combination weighting (or *rate*) via a slider on the web page. Thus, we define $f_{combine}$ based on a *combination rate (r)*.

$$f_{combine}(p, G, r) \rightarrow \left[\frac{p \cdot (100 - \mathrm{r}) + \frac{\sum_{y=0}^{e} G}{e} \cdot \mathrm{r}}{100}, \cdots\right] \tag{8}$$

where p is a correlation score and e is the number of people in a *group (G)*. These equations combine the matrix of the main user with the average matrix of all users to yield a final score.

C. Query Navigation Methods using Primitive Functions

As shown in Figure 4, our system executes the cross domain query navigation by applying the following seven steps.

Step 1: The system receives an initial query keyword from a user. The initial keyword is required to start our query navigation. We have been recognizing several novices have difficulties to input an initial keyword. In this case, a novice user can combine voice recognition method installed in the modern smartphones and tablets.

Step 2: The system applies $f_{construct}$ primitive function to generate the query matrix M and the corresponding transposed matrix M^T.

Step 3: The system applies f_{map} primitive function, which internally invokes the function $f_{\mathrm{narrowing}}$ and the function $f_{sliding}$, into the matrix M and the transposed matrix M^T in order to generate the mapping arrays.

Step 4: The system invokes the primitive function f_{combine} to integrate a user's mapping array and social mapping array, generated from other users' query logs. And then, the system uses $f_{\text{visualize}}$ to render query candidate keywords on the touchscreen by using the weighted mapping arrays.
Step 5: The system receives next keyword input.
Step 6: The system iterates from Step-2 to Step-5 until a user complete describing a query.
Step 7: When a user click a "search" button, the constructed query is submitted to the existing web search engine.

V. Implementation

We implement a prototype system to evaluate the recommendation of search terms by analyzing users' query logs. The system is 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 our system has two main modules for the server side and client side. These modules use the same data structure of a user's search history, but they serve two different functions. On the server side, the system provides the socially shared matrix, whereas on the client side it provides the personalized query expansion using the personal matrix.

The server-side module of the prototype system outputs four 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 the two-dimensional space defined by the narrowing axis and the sliding axis. The search terms are positioned according to two user-input parameters: one defines the combination rate for other search histories (community) and the other defines the scaling rate (zoom factor) for words.

2) *User Interface*

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

The following procedure describes the use of the system:
Step 1: The user inputs an initial keyword for 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 selects it via the touch interface.)
Step 3: If no appropriate keywords are 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. Once an appropriate query has been formed, the user taps the search button and the system retrieves the search results.

Figure 7 shows how the word map can be changed using the social slider. The arrows illustrate 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 suggested by web designers, such as "sample code," "Web design," and "Flash," but not those used by programmers, such as "Java" and "C++." The figure shows that keywords used more often by web designers than programmers, such as "HTML5," are shifted slightly towards the 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.

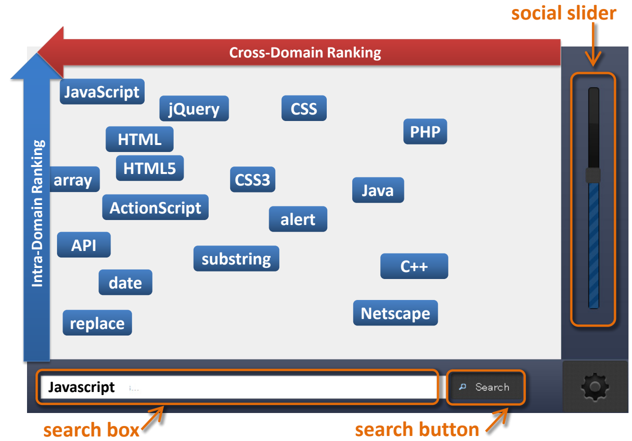


Figure 7. User Interface of Cross-Domain Query Navigation System.

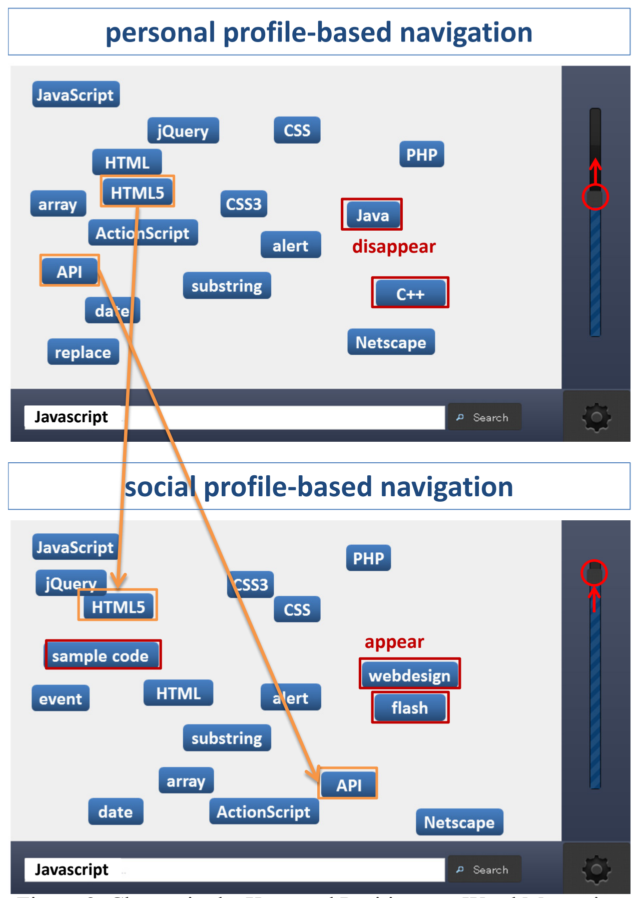
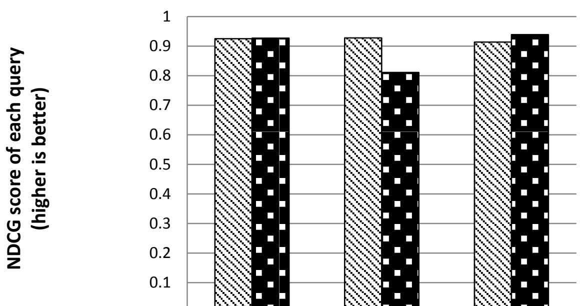


Figure 8. Change in the Keyword Positions on Word Map using the Social Slider.

Table 1. Narrowing and Sliding Keywords and their 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



Query	Q1: design	Q2: e-book	Q3: editorial
narrowing (conventional method: single domain recommendation)	0.925	0.927	0.913
sliding (our method: cross-domain recommendation)	0.926	0.811	0.939

Figure 9. NDCG of Narrowing and Sliding Recommendations.

VI. EXPERIMENTS

This section evaluates the effectiveness of our Q-MAP system when applied to an existing search engine. For this, we implement the Q-MAP prototype system as a Google interface. Two evaluation experiments are performed: Experiment 1 evaluates the precision of the narrowing and sliding relevance computations, and Experiment 2 evaluates the effectiveness of our system by measuring the precision and recall of query expansion. In both experiments, we conduct multiple tests to create a suitable data set.

A. *Experiment 1: Overview*

Experiment 1 investigates the f_{map} function for converting 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 community query logs. We compare narrowing, which is a legacy keyword recommendation, and sliding, which is an original feature of this system. We set up the inter-term relationship matrix by submitting 952 queries to Google. As a result, we obtain a 108 × 108 matrix. The three test topics "design," "e-book," and "editorial" are chosen as the initial keywords. These test cases generate three rankings, and we select the top ten keywords for the narrowing and sliding relevancy from each search topic, as shown in Table 1.

This experiment clarifies that our approach calculates the appropriate distance between query keywords. We ask ten test subjects (seven male and three female) to evaluate the relevance of the 60 keywords from the viewpoint of *narrowing* and *sliding*. Keywords are rated according to the following five-point scheme: 0 (completely irrelevant), 1 (irrelevant), 2 (slightly relevant), 3 (relevant), and 4 (very relevant). We consider the ideal ranking as the average of ten results.

B. *Experiment 1: Evaluation Result*

To evaluate this experiment, we compute the normalized discounted cumulative gain (NDCG).

$$DCG = \sum_{i=1}^{10} \frac{rel_i}{log_2 i} \qquad (9)$$

$$IDCG = \sum_{i=1}^{10} \frac{rel'_i}{log_2 i} \qquad (10)$$

$$NDCG = \frac{DCG}{IDCG} \qquad (11)$$

where rel_i are the average evaluation scores given by the test subjects, and rel'_i are the average scores in descending order.

Figure 8 shows the NDCG of the narrowing and sliding recommendations for our three topics ("design," "e-book," and "editorial"). A higher score implies a better retrieval precision. The most important result is the *sliding recommendation* score, because the narrowing recommendation is similar to conventional query methods. The NDCG of the sliding recommendation is almost the same as that of the narrowing recommendation for each query. This implies that our sliding recommendation achieves highly practical precision, although it generates different keywords from the narrowing recommendation. Using this system, users received precise query keywords that shared a cross-domain relationship with the initial keyword. This recommendation is a very powerful tool for inputting a complex query consisting of cross-domain keywords.

C. *Experiment 2: Overview*

Experiment 2 considers the visualization function that displays the word map according to recommendation scores. The experiment evaluates the precision and recall of the upper-left area of the word map showing the relevant query words. We ask five test subjects (two male and three female) to evaluate the word map, and plot the change in precision and recall while expanding the visible area. By plotting this information, we can evaluate the appropriate size of the visible area to satisfy the user requirements for query expansion precision. The precision p and recall r with a visible area size of k are calculated as follows:

$$p(k) = \frac{V(k) \cap C}{V(k)} \qquad (12)$$

$$r(k) = \frac{V(k) \cap C}{C} \qquad (13)$$

where $V(k)$ denotes the words displayed in area k and C denotes the set of correct words. The visible area size k is the rectangle containing the top-k narrowing words and the top-k sliding words. As preprocessing for this experiment, we ask five subjects to perform several web searches to find as much information as

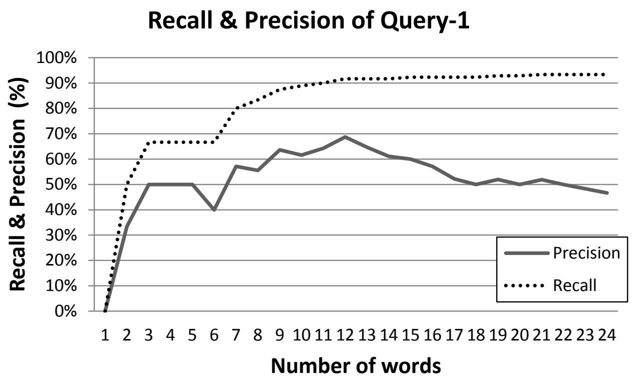


Figure 10. Recall & Precision of Query-1. Initial Keyword is "search."

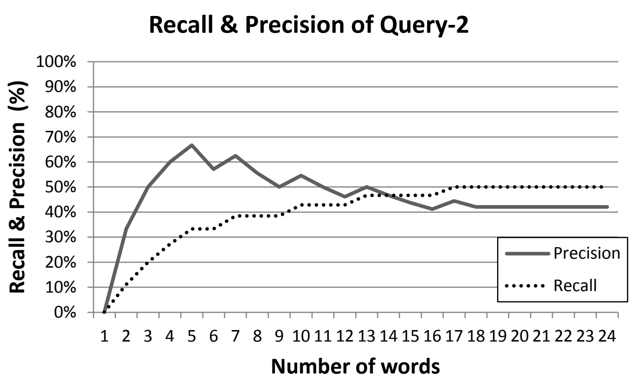


Figure 11. Recall & Precision of Query-2. Initial Keyword is "retrieval."

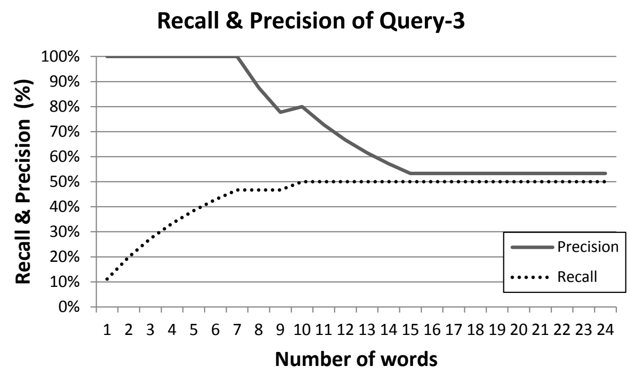


Figure 12. Recall & Precision of Query-3. Initial Keyword is "interactive."

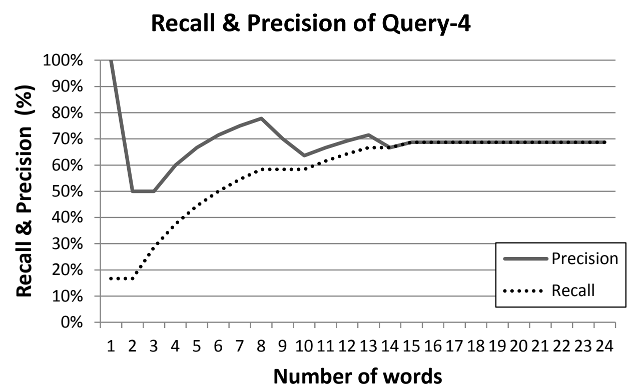


Figure 13. Recall & Precision of Query-4. Initial Keyword is "words."

possible related to the article [1]. We then ask five different test subjects to perform the same search. We set up the inter-term relationship matrix by submitting 808 queries, which contain 93 keyword variations, to Google. As a result, the system has updated totally 34,865 cells in a 108 × 108 matrix.

D. Experiment 2: Evaluation Result

We have plotted eight evaluation results corresponding to the initial query keywords "search," "retrieval," "interactive," "words," "tablet," "smartphone," "typing," and "input" (Figures 10–17). As shown by these graphs, our approach is highly effective at recommending cross-domain keywords when users can select an appropriate initial keyword (Figure 10, Figures 14–17). In contrast, our system could not recommend appropriate keywords when users select an initial keyword that is too common (Figure 11). Figure 12 and 13 show that our system requires sufficiently large query logs to compute the relationship between the initial keywords and other keywords.

Figure 10 shows the recall and precision of Query-1, in which "search" was the initial keyword. In this case, the recall is increasing towards 100% as the number of words *k* increases. In contrast, the precision has saturated after reaching the top-12 ranking.

Figure 11 shows the results for Query-2 (initial keyword: "retrieval"). This result is not so good because the precision score has saturated beyond the top-5 ranking. This is because the query log does not contain a search history that starts with the keyword "retrieval".

Figure 12 and 13 illustrate the limitation of our approach. Figure 12 corresponds to the initial keyword "interactive" and Figure 13 corresponds to the initial keyword "words." In both cases, our system could not compute the relationship between the initial keywords and other keywords due to the small log size. We recognize that our system requires a sufficient log size in order to accept a wide variety of initial keywords. Currently, our experimental data set is relatively small.

Figures 14–17 show that our system provides good keywords for the cross-domain relevance computation. Specifically, Figure 14 shows the recall and precision for Query-5 (initial keyword: "tablet"). In this case, the precision score remains high while the recall score is increasing. It is important to mention that the reason why the precision score remains high is that the system does not show irrelevant keywords on the screen, instead gathering them into the bottom left area of the metric space, which is not visible to users. Figure 15 shows the recall and precision for Query-6 (initial keyword: "smartphone"). The results are similar to those in Figure 14, but the precision score decreases as more words are included. This is because several irrelevant words move to the center of the metric space. Our metric space may include some irrelevant words in the central area.

We consider this to be a trivial matter because many users do not scroll the metric space or look at the central area. Figure 16 and 17 exhibit ideal results from our system.

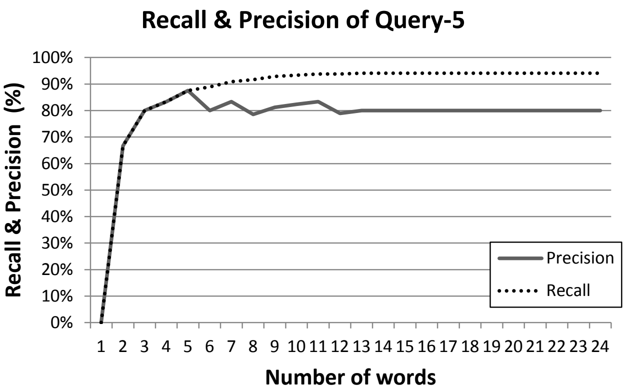


Figure 14. Recall & Precision of Query-5. Initial Keyword is "tablet."

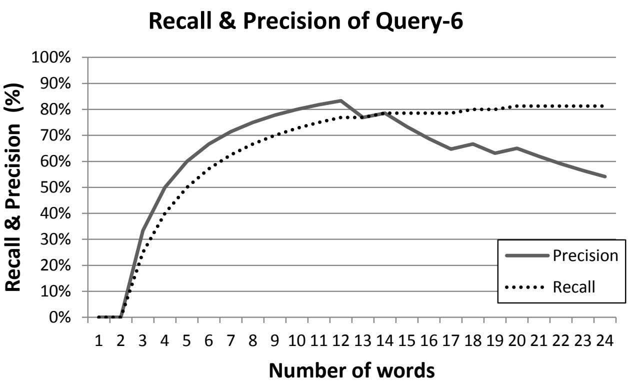


Figure 15. Recall & Precision of Query-6. Initial Keyword is "smartphone."

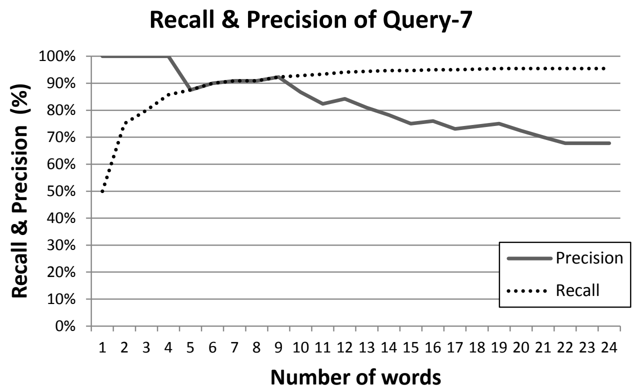


Figure 16. Recall & Precision of Query-7. Initial Keyword is "typing."

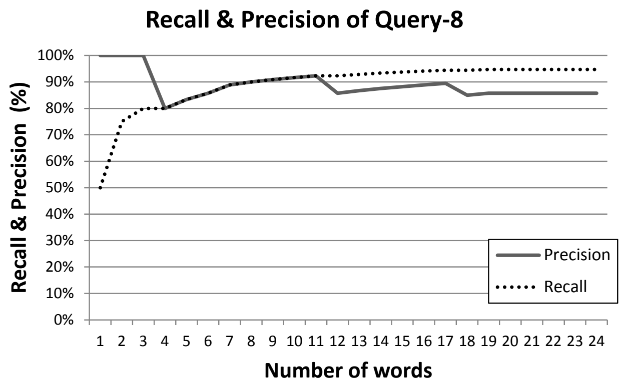


Figure 17. Recall & Precision of Query-8. Initial Keyword is "input."

VII. CONCLUSION AND FUTURE WORK

We have proposed a complex query navigation system that exploits the search history of social groups. This system recommends candidates for the next search term by calculating the directional relevance along two conceptual dimensions and performing narrowing and sliding operations.

The unique feature of our system is that it combines visualization techniques and semantic correlation computing methods to provide an intuitive user interface dedicated to the modern touchscreens. A social combination function enables the user to utilize the knowledge of social groups to facilitate navigation. We have implemented a prototype system that was able to retrieve and present candidate keywords for multiple queries while reducing the number of touchscreen taps required. Our implementation system is running on the modern slate devices such as iPad and Android.

As future work, we plan to develop a social network-based query recommendation mechanism and evaluate the scalability of complex query navigation in multiple domains. In addition, we are working on developing a context-dependent noise reduction mechanism for words appearing the map because we have recognized that several irrelevant words may appear in the visible area of the word map.

REFERENCES

[1] Ryo, S. and Shuichi, K., "Cross-Domain Query Navigation System for Touchscreens by Exploiting Social Search History," in Proceedings of the Seventh International Conference on Internet and Web Applications and Services (ICIW 2012), Stuttgart, Germany, May 27–June 1, 2012, pp. 178-183.

[2] 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: December 6, 2012]

[3] Google, "Admod - Tablet Survey," March 2011. http://www.ccapitalia.net/descarga/docs/2011-AdMob-TabletSurvey.pdf, [retrieved: December 6, 2012]

[4] Harman, D., "Relevance feedback revisited," in Proceedings of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 1–10, 1992.

[5] Kelly, D., Gyllstrom, K., and Bailey. E., "A Comparison of Query and Term Suggestion Features for Interactive Searching," in Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR2009), 2009, pp. 371–378.

[6] Kato, M.P., Sakai, T., and Tanaka, K., "Structured Query Suggestion for Specialization and Parallel Movement: Effect on Search Behaviors," in Proceedings of the 21st

International Conference on World Wide Web (WWW2012), ACM, 2012, pp. 389–398.

[7] Wen, J., Nie, J., and Zhang, H., "Clustering User Queries of a Search Engine," In Proceedings of the 10th International Conference on World Wide Web (WWW2001), 2001, pp. 162–168.

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

[9] Teevan, J., Dumais, S. T., and Horvitz, E., "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, pp. 449-456.

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

[11] Smyth, B., Balfe, E., Freyne, J., Briggs, P., Coyle, M., and Boydell, O., "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.

[12] Dumais, S., Cutrell, E., and Chen. H., "Optimizing Search by Showing Results in Context," in Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI2001), 2001, pp. 277–284.

[13] Yee, K., Swearingen, K., Li, K., and Hearst, M., "Faceted Metadata for Image Search and Browsing," in Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI2003), 2003, pp. 401–408.