Evaluating an Associative Browsing Model for Personal Information

Jinyoung Kim, W. Bruce Croft, David A. Smith and Anton Bakalov
Department of Computer Science
University of Massachusetts Amherst
{jykim,croft,dasmith,abakalov}@cs.umass.edu

ABSTRACT

Recent studies [3] [2] suggest that associative browsing can be beneficial for personal information access. Associative browsing is intuitive for the user and complements other methods of accessing personal information, such as keyword search. In our previous work [9], we proposed an associative browsing model of personal information in which users can navigate through the space of documents and concepts (e.g., person names, events, etc.). Our approach differs from other systems in that it presented a ranked list of associations by combining multiple measures of similarity, whose weights are improved based on click feedback from the user.

In this paper, we evaluate the associative browsing model we proposed in the context of known-item finding task. We performed game-based user studies as well as a small scale instrumentation study using a prototype system that helped us to collect a large amount of usage data from the participants. Our evaluation results show that the associative browsing model can play an important role in known-item finding. We also found that the system can learn to improve suggestions for browsing with a small amount of click data.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: [Information Search and Retrieval]

General Terms

Algorithms

Keywords

Associative Browsing, Personal Information Management, Known-item Finding, Human Computation Game

1. INTRODUCTION

Associative browsing, the process of going through personal information by following a chain of associations, has several benefits. First of all, studies in cognitive psychology [5] [15] suggest that people remember facts primarily by associations, which explains the intuitive appeal of associative browsing. Also, Teevan et al. [14] suggest that many people tend to find information by a series of small steps (orienteering) instead of using keyword search.

Recently [9], we proposed a conceptual model of associative browsing for personal information. When keyword search fails to bring the desired item into the top results, a user can try browsing to the item by clicking on a result and following associations between that and other items. These associations are calculated based on various features, which are combined to create a ranking of associated items. This ranked list of items are presented to the user and the click from the user is used as a feedback to improve the weighting between features. The work improved on previously suggested models of associative browsing [3] [2] in that we proposed using more general measures of association (e.g. textual similarity and co-occurrence), and that we introduced the idea of click-based training of the feature weights.

In this paper, we introduce a learning framework for ranking suggestions for browsing, and evaluate the associative browsing model in the context of known-item finding, which is the most common task in personal information access [6]. The known-item finding task also has a well-defined structure with a concrete target item, which allowed us to use a novel evaluation method.

Specifically, we performed a game-based user study in which participants were asked to find a set of target documents by combining keyword search and associative browsing. The study shows that the participants often choose to use associative browsing. It also reveals insights on their known-item finding behavior.

Using the click data collected during the user study, we show that it is beneficial to combine many similarity measures for ranking browsing suggestions, and that the system can improve the quality of the ranked list with a small amount of click data. Moreover, the analysis of user's behavior during a game-based user study shows that people choose to use browsing when search results are only marginally satisfactory.

The rest of this paper is organized as follows. In the next section, we provide an overview of related work (§2). Then we introduce the associative browsing model (§3) and the learning method for ranking suggestions (§4), followed by the evaluation methods based on game-based user study (§5). Finally, we present experimental results(§6).

CIKM'11, October 24–28, 2011, Glasgow, Scotland, UK.

2. RELATED WORK

Since the early days of IR, researchers have been interested in the combination of search and browsing for accessing document collections. Lucarella [11] described a system with a network of concepts and documents which provides search and browsing capability in a complementary manner. The I^3R system developed by Croft et al. [4] also assumes a scenario where documents returned by a user's initial query provide a starting point for subsequent browsing.

Kaplan et al. [8] described a navigation scheme that adapts to user behavior. Smucker et al. [13] found that similarity browsing can improve retrieval effectiveness when used as a search tool. Compared to these systems, our proposed approach is novel in that it suggests a feature representation of links between items. The weights of these links are trained using the click feedback from the user. Our model also has the notion of *concept*, which provides an extra access mechanism to documents. Finally, while they assumed that the user has already found a relevant document, we do not make such assumption here.

Human computation games [16] have recently been suggested as a method for obtaining a large amount of human annotations in a way that motivates participants. In the context of IR research, Ma et al. [12] introduced PageHunt, which is a game designed to collect web search log data by asking participants to find pages that they were shown. Recently, Kim et al. [10] developed a variant of PageHunt to collect the data for the evaluation of desktop search algorithms. This work is different in that we designed a game in which search and browsing are supported at the same time. We also analyzed session logs to gain insights into the use of the system, whereas previous work mostly used the data at the query level.

3. ASSOCIATIVE BROWSING MODEL

In this section, we briefly introduce the associative browsing model we proposed earlier [9], focusing on its application to known-item finding. At a high level, our associative browsing model is composed of information *items* and the *associations* between them. An item is a fundamental unit of our data model, which can be either the *documents* collected from many sources (e.g., desktop files, emails, calendar items), or the *concepts* (e.g., person names, events, etc.).

One distinctive part of our model is *concepts*, which denote entities and terms of interest to the user. They are similar to labels or facets in that they provide an abstract layer of organizing documents, yet they are distinctive in that they form a space of associations on their own.

Another feature of our data model is a rich link structure between items as seen in Figure 1. The associations between concepts and documents are created based on the occurrence of a concept within a document (e.g. an *email* and its *sender*). While this provides natural connections between documents and concepts, creating associations between documents and between concepts is harder, since there is no single method that gives both high coverage and precision.

Previous research solves this problem by asking the users to create associations manually, or by extracting the associations automatically between a limited set of items. In our work, we address this issue by ranking candidates for browsing using the combination of many similarity metrics. The top k items are then presented to the user as suggestions for browsing, and the system learns to improve suggestions based on the click feedback from the user. The details of ranking suggestions for browsing are provided in Section 4.

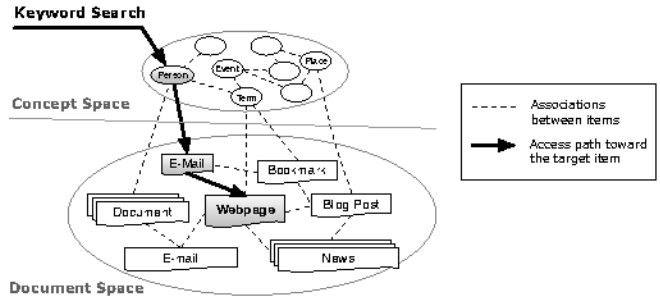

Figure 1: An illustration of how the suggested associative browsing model can be used for known-item finding.

3.1 Using the Model for Known-item Finding

There can be many use cases for this associative browsing model. For instance, such a rich network of association would be suitable for exploratory search [17] in personal information. Another common use case is known-item finding, where associative browsing can provide a back-up strategy for keyword search. In this paper, we focus on evaluating the model in the known-item finding scenario, since it is the most common task in personal information access and the well-defined structure allows us to use the evaluation methods introduced in Section 5.

Here, we provide an example on how associative browsing can be combined with keyword search for known-item finding. Imagine a user who is trying to find a webpage she has seen. Further assume that she cannot come up with a good keyword for search, yet she remembers the sender of related email. Using our model, as shown in Figure 1, the user can first use keyword search to find a relevant concept (person), and then browse into the target document (webpage) through another document (email) associated with both the concept and the target document. Here, dotted lines represent the associations between documents and concepts. Directed lines denote how a user can access the target webpage by using keyword search and associative browsing.

4. RANKING SUGGESTIONS FOR ASSOCIATIVE BROWSING

A core component of our model is the ranked list of related concepts or documents, generated by combining multiple measures of association. Another important task is finding appropriate weights for each feature.

In this section, we introduce a learning framework for creating and improving suggestions for browsing. As is the case of our data model, this learning framework is sufficiently general to be used beyond the domain of personal information. We first explain the features we used for representing the associations between two items, followed by the methods we employed to learn feature weights. Since many features are similarity measures, we will use the term *similarity* interchangeably with *association*.

4.1 Features

The following subsections describe the features we used to rank suggestions for browsing. Note that some of features are applicable only for the ranking of concepts or documents. If that is the case, then the text in brackets after the name of the feature will reflect that.

4.1.1 Term Vector Similarity

We can create a term vector for each item based on the text in the title or content fields. Since many concepts do not have any text in their content fields, we use the documents in which the concepts occur. The term vector similarity score of two items is just the cosine similarity of the corresponding term vectors.

4.1.2 Tag Overlap

Since concepts and documents have tags associated with them, we can consider two items with common tags to be similar. Given two vectors of tags, we compute the tag overlap score using the cosine similarity.

4.1.3 Temporal similarity

Intuitively, two items are deemed to be close to one another if the system indexes them or if the user creates them within a short period of time. Therefore, the closer the creation of two items is in time, the higher their temporal similarity score. We got the feature value by taking the reciprocal of the difference in creation time (in seconds).

4.1.4 String Similarity (concept)

We compute the string-level similarity by dividing the Levenshtein distance between the titles of two concepts by the square root of the product of the title lengths as follows:

4.1.5 Co-occurrence (concept)

This feature counts how many times each concept pair occurs together in the collection's documents. It captures the semantic distance between two concepts. This metric is available only for the calculation of concept similarity.

4.1.6 Occurrence (concept)

This feature counts the number of times a concept has occurred in the document collection in log scale. Although all the other features measure some kind of similarity, this metric is intended to capture the popularity of a concept, since such concepts are likely to be clicked by a user.

4.1.7 Topical Similarity (document)

This feature relies on the topic model Latent Dirichlet Allocation (LDA) [1]. LDA is a hierarchical Bayesian model, which allows us to model a text document as a mixture of topics. To measure the similarity between two documents, we calculate the cosine similarity between the distribution of topics associated with each document. This is similar to computing the similarity of term vectors, except that each document is mapped to a vector of latent topics instead of terms.

4.1.8 Path / Type Similarity (document)

Since each document has a URI, we can compute a similarity score between two documents based on the *path*. Specifically, we calculate the similarity between two path strings by counting the word-level overlaps from the beginning of the path, normalized by the number of words. Also, since each document has a *type* (e.g., email, pdf, etc.), we developed a binary feature based on whether two documents are of the same type.

4.1.9 Concept Overlap (document)

This feature is similar to tag overlap in that it considers two documents with common concepts to be similar. Unlike tags, since we can measure the strength of association between any two concepts, we can use it to measure the similarity between documents. In other words, even if two documents are linked to different sets of concepts, we can consider them to be similar if the concepts that each of them has are strongly associated.

4.2 Learning Feature Weights

One key step of our system is learning the weight of each feature. We examined two algorithms – iterative grid search and RankSVM [7]. The two learning methods used here have different characteristics. As far as the objective function is concerned, grid search simply finds the set of parameters that maximizes the target metric, whereas the goal of RankSVM is to predict the pairwise preference relation with highest accuracy. There is another aspect in which the two methods differ. While grid search uses each click as a relevance judgment, RankSVM interprets each click as a pairwise preference. We investigate the performance of the two learning methods in Section 6.1.

5. GAME-BASED EVALUATION METHOD

The method we employed for our evaluation is a game-based user study where we asked people to perform known-item finding tasks using both search and browsing capabilities. Using the data from the user study, we analyze the user's behavior in finding known-items, and evaluate the algorithms for ranking suggestions for browsing.

We call this a game because participants were competing against one another for how well they find known items. In addition to providing usage data in a controlled environment, this game-based evaluation method has advantages in terms of reusability—the whole collection and usage logs can be made public without privacy concerns.

Now we describe the design of the game-based user study. From the player's perspective, the purpose of the game is to find a target document by combining keyword search and associative browsing. We use the term 'session' to denote the process of finding each target document, and each game is composed of 10 sessions.

The sequence of interaction for each session, and the corresponding user interface is shown in Figures 2 and 3. As a starting point, the system shows two candidate documents to the users for a certain period of time, and then randomly chooses one target document. The user then combines keyword search and associative browsing to find the item. Each keyword query or click on the ranked list is considered a trial, and users are given 10 trials for each session. The score is determined by the rank position of each target document in the final rank list — the higher the position, the higher the score.

The rationale behind showing multiple target documents is to simulate the state of vague memory for the target document. For instance, if the user is shown an *email* and a *webpage* in a row and then asked to find it, he or she might get confused about the content of two documents. We assumed that this kind of confusion would be similar to the memory of a typical known-item searcher. We leave it as a future work to verify whether this kind of trick realistically simulates the state of memory for known-item finders.

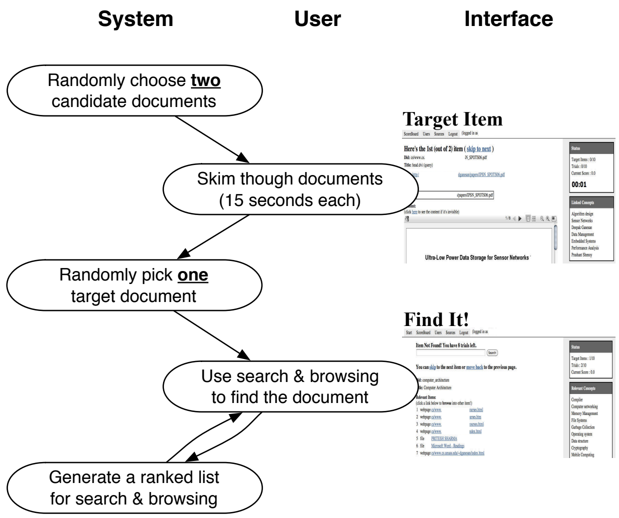

Figure 2: The sequence of interaction for each session during a game.

We ran two rounds of user studies with slightly different settings. In the first round, users were asked to find the target document using only keyword search of documents and associative browsing between documents. In other words, they did not have access to the concept space. In the second round, concepts were available for searching and browsing, thereby providing a full access to the model. We chose this two-stage design to evaluate the role of each system component and to help users gradually familiarize themselves with the system.

6. EVALUATION RESULTS

In this section we present the evaluation results. We used three document collections to evaluate the system. The first two collections are based on a preliminary study we did with two users, and we used the click data from these collections for training and evaluating ranking methods for browsing suggestions. We built the third collection, and used it to run the game-based user study. We describe the details of the collections below.

For creating the first and the second collections, two student volunteers in our department, Person 1 and Person 2, deployed the system in their machine and used it over the period of two weeks. They were encouraged to use the system for everyday information access tasks. The former contains 8,841 documents and 368 concepts and the latter contains 9,441 documents and 945 concepts. Both collections are mostly composed of emails, webpages they visited and desktop files.

As for concepts and tags, users created them as they use the system. No specific task was given to them, except that they were encouraged to use the system for their information access. At the end of data collection period, Person 1 clicked 145 times on the ranked list of concepts and 58

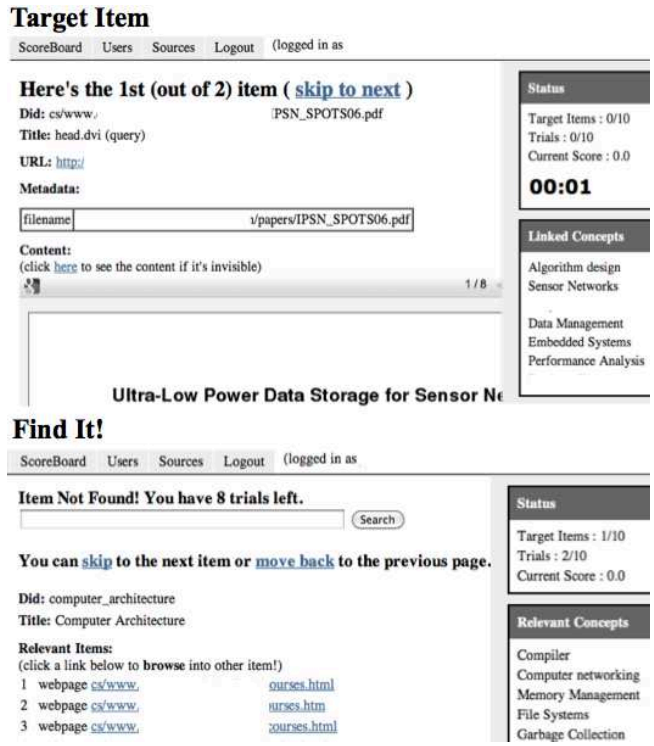

Figure 3: Game user interface. Top: a target document is being shown along with related concepts. Bottom: the user interface for finding a target document by searching and browsing.

Table 1: Number of documents, concepts, and clicks in the case of document similarity and concept similarity experiments for each of the collections we used.

	#Items		#Clicks	
	Document	Concept	Document	Concept
Person 1	8841	368	58	129
Person 2	9411	945	204	196
CS/Top1	7984	650	145	42
CS/Top5	”	”	309	220

times on the ranked list of documents. Person 2 had 196 clicks and 204 clicks on concepts and documents, respectively. We also found that only Person 2 actively created tags (56 unique tags created in total), whereas Person 1 created only a handful of tags.

The third dataset (called CS collection) contains public emails, webpages, publications and lectures crawled from the computer science department website of the authors. As for concepts, we selected the names and terms related to the computer science department and the domain of computer science in general — the name of people, lab and conferences. We also created 42 tags and assigned to each item, mostly based on its categorical information (e.g., ‘student’, ‘professor’ and ‘staff’ for person names).

The CS collection was created as a reasonable simulation of personal information, since we experimented with participants from our department, who had some knowledge of these documents and concepts. This collection is composed of 7,984 documents and 650 concepts.

We had 30 participants for the game-based user study within our department, who were mostly graduate students. They played 53 games in total, although some of the games were not completed (less than 10 sessions were recorded). Regarding click data, since most people contributed only a few clicks, we used the data from a user with the highest number of clicks (CS/Top1). To find the effectiveness of ranking when the click data pooled from many people, we also experimented with the aggregate data from the five users with most clicks (CS/Top5). The number of items and clicks are summarized in Table 1.

For learning methods, we used our own implementation of Iterative Grid Search and SVM^{rank} [7], which is a popular implementation of RankSVM. To facilitate the training of SVM^{rank}, each feature value was scaled to values that were approximately between 0 and 1. We also used 10-fold cross validation for training feature weights and evaluating the system.

In order to measure retrieval performance, we used the mean reciprocal rank (MRR), which is the average of the reciprocal of click positions. We also used clicks as relevance judgments, because the goal of the ranking is to show the items that the user is likely to click on the top. In other words, we wanted the ranking to adapt to user's subjective notion of relevance. In what follows, we first describe the results on the quality of the suggestions generated by the learning framework. We then present the analysis of users' behavior in known-item finding.

6.1 Evaluating Suggestions for Browsing

We present the evaluation results on the quality of browsing suggestions, with the goal of evaluating the effectiveness of click-based feature combination as well as individual feature. We compared the performance obtained when each feature was used by itself and when three combination methods were used—feature values with equal weights (*Uniform*), weights obtained with grid search (*Grid*) and with RankSVM (*SVM*), respectively. Note also that *title* and *content* are term vector similarity features, and the title and the content field was used for constructing term vectors, respectively.

Table 2 shows the concept ranking results for each feature and combination method. Regarding the single-feature results, different features turned out to be the most effective ones for each collection. Specifically, we found that *co-occurrence* is the most effective feature in Person 1's and Person 2's collection, while *occurrence* and *tag* was the best in CS/Top5 and CS/Top1, respectively. From this we can conclude that there exists a considerable variation in the value of each feature depending on the collection and the click behavior.

Among the combination methods, RankSVM performed the best for all collections except for Person 2's, where Grid Search performed the best. Another observation is that even the naive uniform combination of features produced better results than any of the ones obtained by using a feature by itself. In summary, different features perform the best for each collection, yet combination results are consistently better than singe-feature results. These results imply that feature combination is beneficial for ranking concepts for browsing.

As far as the document ranking task is concerned, Table 3 shows a slightly different trend. Term vector similarity using the *content* field is far more important than any other features in the case of Person 1 and Person 2. This makes intuitive sense because documents typically contain more textual content. This results in more accurate term vectors and subsequently better term vector similarity estimates. The best feature for the CS collection was the *topic* similarity.

In the case of combination methods, grid search performed better than any feature used by itself, while RankSVM was not as effective as it was in concept ranking. Although the performance margin between combination and single-features methods is small, given that it is hard to know which feature would work best a priori, we can conclude that feature combination should be used here as well.

For both concept and document ranking tasks, another observation is that using the click data from one user (CS/Top1) showed better performance than the one we obtained using the click data from the top five users. Since more click data usually leads to better performance, this unexpected drop in ranking quality seems to suggest that learning from the data of each user is important for improving performance. We investigate this point in the following subsection.

6.2 Analyzing How Users Find Known-items

Here we analyze user's behavior in known-item finding using the data from the game-based user study. As described in Section 5, we performed two rounds of game-style user studies in which participants were asked to find a set of randomly chosen target documents using the game interface. While each session had to be initiated by keyword search, participants had an option of browsing by clicking on a document in ranked list.

We first focus on the role of browsing in known-item finding by analyzing the portion of sessions where browsing was used, and how much of them was successful (target document found at Top 10). Table 4 presents the results. We have 290 sessions from Round 1 and 142 sessions from Round 2. The percentage of sessions during which users chose to browse as well as search is 14.5% (or 42 sessions) for Round 1 and 30.2% (or 43 sessions) for Round 2.

Although the percentage of sessions with browsing was not as high as was expected, since we envisioned the browsing as a complementary method to keyword search, what seems more important is the portion of success with browsing. Furthermore, the fact that we have more browsing in the second round seems to suggest that the concept space provided further motivation for browsing.

Table 4: The ratio of the sessions where users chose to use browsing, and the choice of browsing led to success.

Round	Total	Browsing used	Successful
1st	290	42 (14.5%)	15 (35.7%)
2nd	142	43 (30.2%)	32 (74.4%)

Let's look at the success ratio of the sessions with browsing. Out of the 42 sessions in Round 1 involving both searching and browsing, 35.7% (or 15 sessions) of them were successful, i.e., the user found the required document. For Round 2, this percentage is 74.4 (or 32 sessions). Given that users turn to browsing only when initial search is not successful, this success rate can be considered moderately high.

Also, the higher successful rate in the second round can be attributed to the presence of the concept layer. In fact, many users commented that they could find the target document using the concept as an intermediate step. Another

Table 2: Concept ranking performance (MRR) for the single-feature and combination methods. 10-fold cross-validation was used for grid search and RankSVM (SVM).

Collection	title	content	tag	time	string	cooc	occur	Uniform	Grid	SVM
Person 1	0.097	0.229	0.194	0.136	0.136	**0.241**	0.151	0.243	0.236	**0.277**
Person 2	0.037	0.350	0.403	0.221	0.310	**0.516**	0.234	0.506	**0.581**	0.509
CS/Top1	0.142	0.179	**0.289**	0.235	0.107	0.191	0.195	0.306	0.255	**0.433**
CS/Top5	0.184	0.127	0.170	0.155	0.100	0.158	**0.222**	0.293	0.301	**0.340**

Table 3: Document ranking performance (MRR) for the single-feature and combination methods. 10-fold cross-validation was used for grid search and RankSVM (SVM).

Collection	title	content	tag	time	topic	path	type	concept	Uniform	Grid	SVM
Person 1	0.392	**0.480**	0.063	0.296	0.229	0.274	0.183	0.264	0.404	**0.500**	0.494
Person 2	0.334	**0.564**	0.268	0.372	0.184	0.137	0.092	0.187	0.512	**0.592**	0.478
CS/Top1	0.074	0.097	0.065	0.114	**0.140**	0.098	0.070	**0.140**	0.109	**0.156**	0.098
CS/Top5	0.081	0.138	0.05	0.114	**0.151**	0.132	0.062	0.129	0.139	**0.150**	0.133

comment was that the process of browsing was helpful in suggesting good query words.

In summary, the analysis of user's behavior shows that users find associative browsing helpful for known-item finding, and the use of concept layer makes the interaction more effective.

7. CONCLUSIONS

In this paper, we evaluated an associative browsing model we proposed in the context of known-item finding. In ranking suggestions for browsing, we showed that the value of each association measure varies depending on the collection and on the user behavior, and that the weighted combination of individual features improves the quality of suggestions in all cases. he game-based user study also suggests that the model is useful for the known-item finding task, especially when concepts are used in addition to documents.

As far as future work is concerned, we plan to evaluate our model in a more realistic setting. Although we found initial evidence that associative browsing is helpful for the known-item finding task, a long-term study with actual users could further verify our claims. For learning methods, we are going to examine whether incorporating more features and click data leads to better performance.

8. ACKNOWLEDGEMENTS

This work was supported in part by the Center for Intelligent Information Retrieval and in part by the Central Intelligence Agency, the National Security Agency and National Science Foundation under NSF grant #IIS-0326249 . Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the sponsor.

9. REFERENCES

[1] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, 2003.

[2] D. H. Chau, B. Myers, and A. Faulring. What to do when search fails: finding information by association. In *CHI '08*, pages 999–1008, New York, NY, USA, 2008. ACM.

[3] J. Chen, H. Guo, W. Wu, and W. Wang. imecho: an associative memory based desktop search system. In *CIKM '09*, pages 731–740, New York, NY, USA, 2009. ACM.

[4] W. B. Croft and R. H. Thompson. I^3R : A new approach to the design of document retrieval system. Technical report, Amherst, MA, USA, 1987.

[5] G. Davis and D. Thomson. Memory in context: Context in memory. England, 1988. Wiley.

[6] D. Elsweiler and I. Ruthven. Towards task-based personal information management evaluations. In *SIGIR '07*, pages 23–30, New York, NY, USA, 2007. ACM.

[7] T. Joachims. Optimizing search engines using clickthrough data. In *KDD '02*, pages 133–142, New York, NY, USA, 2002. ACM.

[8] C. Kaplan, J. Fenwick, and J. Chen. Adaptive hypertext navigation based on user goals and context. *User Modeling and User-Adapted Interaction*, 3(3):193–220, 1993.

[9] J. Kim, A. Bakalov, D. A. Smith, and W. B. Croft. Building a semantic representation for personal information. In *In Proceedings of CIKM'2010, Toronto, Ontario, Canada*, 2010.

[10] J. Kim and W. B. Croft. Ranking using multiple document types in desktop search. In *In Proceedings of SIGIR '10*, pages 50–57, New York, NY, USA, 2010. ACM.

[11] D. Lucarella. A model for hypertext-based information retrieval. pages 81–94, 1992.

[12] H. Ma, R. Chandrasekar, C. Quirk, and A. Gupta. Improving search engines using human computation games. In *CIKM*, pages 275–284, 2009.

[13] M. D. Smucker and J. Allan. Find-similar: similarity browsing as a search tool. In *SIGIR '06*, pages 461–468, New York, NY, USA, 2006. ACM.

[14] J. Teevan, C. Alvarado, M. S. Ackerman, and D. R. Karger. The perfect search engine is not enough: a study of orienteering behavior in directed search. In *CHI '04*, pages 415–422, New York, NY, USA, 2004. ACM.

[15] E. Tulving and D. Thomson. Encoding specificity and retrieval processes in episodic memory. In *Psychological Review*, pages 352–373, England, 1973.

[16] L. von Ahn and L. Dabbish. Designing games with a purpose. *Commun. ACM*, 51(8):58–67, 2008.

[17] R. W. White and R. A. Roth. *Exploratory Search: Beyond the Query-Response Paradigm.* Morgan & Claypool, 2009.