

Task Tours: Helping Users Tackle Complex Search Tasks

Ahmed Hassan
Microsoft Research
One Microsoft Way
Redmond, WA 98052 USA
hassana@microsoft.com

Ryen W. White
Microsoft Research
One Microsoft Way
Redmond, WA 98052 USA
ryenw@microsoft.com

ABSTRACT

Complex search tasks such as planning a vacation often comprise multiple queries and may span a number of search sessions. When engaged in such tasks, users may require holistic support in determining the required task activities. Unfortunately, current search engines do not offer such support to their users. In this paper, we propose methods to automatically generate *task tours* comprising a starting task and a set of relevant related tasks, some or all of which may be necessary to satisfy a user's information needs. Applications of the tours include helping users understand the required steps to complete a task, finding URLs related to the active task, and alerting users to activities they may have missed. We demonstrate through experimentation with human judges and large-scale search logs that our tours are of good quality and can benefit a significant fraction of search engine users.

Categories and Subject Descriptors

H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval – *selection process, search process*

Keywords

Task tours; Task graph; Search task support

1. INTRODUCTION

Web search engines return lists of items ranked by estimates of their query relevance. Information retrieval (IR) researchers have worked extensively on algorithms to effectively rank documents (c.f. [19]). In addition, some Web search engines now offer manually-curated lists of sites for particular tasks created by human editors. The Editors' Picks feature of the Microsoft Bing search engine (bing.com/editors-picks) is one example of such functionality. However, individual items are often insufficient to satisfy complex needs such as understanding medical conditions, planning a vacation, or buying a home [2]. When attempting such tasks, people may need support that extends beyond a ranked list, and alerts them to the steps required for task completion [10]. For example, prior work [7][8] has shown that trails comprising a filtered set of documents arranged in sequence can help searchers.

Tours and trails are frequently generated by human trailblazers [8][20]. While this approach aligns well with Vannevar Bush's original vision in his seminal article entitled *As We May Think* [4] it may not scale to the broad range of search tasks that people can perform with search engines. Although trails can be generated algorithmically [10], the methods proposed to date involve restricted domains such as particular websites or hypertext corpora rather than Web search [7][22], specific URL paths [24] rather than more general level of abstraction (e.g., topical categories) which may be more widely applicable, or recommend one step at

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

CIKM'12, October 29–November 2, 2012, Maui, Hawaii, USA.

Copyright 2012 ACM 1-58113-000-0/00/0010...\$10.00.

a time [10][14] rather than providing a holistic view of the tour, which might be useful to searchers, especially novices in the domain of interest. We need to address these shortcomings and provide users with scalable support for search task completion.

In this paper we present and evaluate methods to automatically create multi-step *task tours* that can help users perform complex (multi-stage) search tasks. A task tour comprises a trigger task and set of other following tasks that the user can attempt in any order. For example, a task tour for the query [buying a home] as generated by our algorithm is: *Real Estate Search* (the trigger); *Find a Realtor*; *Financial Services*; *Online Maps*; *Public Education*, indicating many of the key tasks required when purchasing a new residence. When the trigger task is detected, say via a query match, search engines can show the tour could be shown on the search engine result page (SERP) accompanied by popular URLs or domains for each task, indications of what tasks remain to be completed, etc. As we will show, human judgments show that the task tours we generate are of high quality and a log analysis predicts significant utility from the deploying task tours at scale.

The remainder of this paper is structured as follows. Section 2 describes related work in areas such as modeling search activity and creating guided tours. Section 3 provides details on how we create the task tours. In Section 4 we describe the evaluation methodology and in Section 5 we describe our experimental results. We conclude in Section 6.

2. RELATED WORK

There are several areas of related work that are relevant to the research presented in this paper: (i) modeling search activity beyond basic querying, (ii) leveraging interaction sequences from search logs, (iii) creating trails and guided tours through information spaces, and (iv) modeling search interests to provide step-by-step recommendations to searchers. We now describe work in each of these related areas in more detail.

Modeling Search Activity: Models of information seeking have been developed that illustrate the value and necessity of moving well beyond the search result page. O'Day and Jeffries [12] proposed an orienteering analogy to understand users' information-seeking strategies. Their qualitative study relates to ours in describing the benefits of a system that considers the entirety of users' trails through information spaces. Other models of information search behavior have been proposed. Two well-known examples of such models are berrypicking [2] and information foraging [15]. Berrypicking describes the movement between information sources associated with dynamic information needs. Information foraging, derived from foraging for food in the wild, highlights how information seekers can use cues left by previous visitors to find patches of information in a collection, and then consume this information to satisfy their needs.

Mining Search Sequences from Logs: Search logs containing the search engine interactions of thousands or millions of users have been mined extensively to enhance search-result ranking [1][9]. Moving beyond the use of aggregate interactions for a

single-query, Radlinski and Joachims [16] used connected sequences of similar queries as implicit feedback to improve result ranking. Rich log data, from sources such as browser toolbars, offer insight into user behavior beyond search engines. Trails comprising query and post-query page views can be mined from these logs [23] and used to help guide future searchers.

White and Huang [24] performed a log-based study to assess search trails followed by users, and showed that users benefited from the intermediate pages as well as the origin and destination pages (terminal trail URLs). Bilenko and White [3] studied full trails mined from logs, including the origin, intermediate, and destination pages. They found that treating the pages in these trails as endorsements improved their ranking in search engines.

Guided Tours and Trailblazing: Guided tours and trails have been proposed to help users situate themselves and navigate within information spaces [4]. This approach has been used within the hypertext community or to support navigation within a Website [17]. Tours and trails can be created manually or automatically.

Manually Generated: Hammond and Allison [8] and Trigg [20] proposed guided tours in hypertext to ease problems of user disorientation. These tours comprised a connected sequence of cards that were presented to users in a pre-determined order. Wexelblat and Maes [21] introduced annotations called footprints that reveal trails through a Web site assembled by the site’s designer.

Automatically Generated: Dispensing with human intervention, tours and trails can also be generated automatically. Guinan and Smeaton [7] generated a tour for a given query based on term-matching for node selection and inter-node relationships (e.g., is_a, precedes) for node ordering. In a user study based on a hypertext collection of lecture materials, they found that users followed these trails closely—40% of the time participants did not deviate from the suggested trail. Wheeldon and Levene [22] proposed an algorithm for generating trails to assist in Web navigation through a particular website. Study participants found trails to be useful and noted that seeing the inter-link relationship.

We extend previous research in a number of different ways. First, we automatically generate task tours rather than relying on humans to generate them explicitly by creating them manually, or implicitly by mining previously followed trails from log data. Second, tours are created at the category level rather than the document level, allowing us to detect higher-level patterns in search behavior, while also being insensitive to Web dynamism (e.g., changing content, dead links). Third, we focus directly on supporting Web search rather than assisted Website navigation or navigation through restricted domains such as Wikipedia. Finally, we propose and utilize new methods for evaluating the quality, utility, and potential benefit of tours without costly user studies.

3. TOUR GENERATION

In this section we provide an overview of the data that we used to generate the tours, the labeling of the URLs with the topical category information necessary to identify tasks, the construction of the task graph that relates the tasks to each other, the building of the tours using the graph, and finally the identification of triggers. Note that we assume that the order of the tasks following the trigger task is undetermined. Imposing a sequence ordering on tour elements may be unnecessarily restrictive for the complex tasks that lack temporal dependence between their component steps.

3.1 Data

The data we used for this study was a sample of the anonymized logs of all URL visits by users who opted in to provide data logs

through a widely-distributed browser toolbar. We used a sample of 10,000 different users over a two-month time period from February 2012 to March 2012 (drawn from a larger sample of millions of users). The data from one month was used to create the task tours and the data from the other month was used in a log-based study to validate the findings for unseen data (more on this later in the paper). While constructing the dataset, we excluded users with few page visits (less than 50 per month). We also excluded a small number of users who had too many visits (over 1000 per day) to make sure that our analysis is not biased toward the behavior of a small set of users. The dataset contained millions of visited URLs and was segmented into sessions.

3.2 Labeling URLs

There have been many studies in the literature on task type or task intent classification (e.g., [11]). Those studies have developed schemes to classify tasks into categories such as: Fact finding, Information gathering, Undirected browsing, Transaction, etc. While these categories may be useful for different scenarios, they are not particularly useful for our scenario. The main reason for this is that these are very high level categories that correspond to task types rather than to the information need behind the task.

Other studies have looked at page topicality, but with millions of pages in our dataset, it is impractical to download and use the text in the Web pages. Conversely, we could have looked at URLs or domains but that would be very limited due to data sparseness. To address this challenge, we used the Open Directory Project (ODP), also referred to as dmoz.org. ODP is an open Web directory maintained by a community of volunteer editors. It uses a hierarchical scheme for organizing URLs into categories and subcategories. Many previous studies have used ODP to assign topical categories to URLs (e.g., [18]). These studies have focused on the top two level categories (e.g. *Recreation: Travel*). This is sufficient for page topicality, but it is not enough for our purposes. Instead, we used the top three level categories or the top four level categories if they exist. This results in breaking a task like *Recreation: Travel* into many more granular tasks, e.g., *Recreation: Travel: Transportation: Air*, *Recreation: Travel: Lodging*, etc.

Given the large number of URLs in our set we needed to label them automatically. We performed automatic classification of URLs into ODP categories via an approach similar to [24]. URLs in the directory were directly classified according to the corresponding categories. Missing URLs were incrementally pruned one level at a time until a match was found or a miss declared.

3.3 Task Graph

Recall that our goal is to find a good task tour to support the current search task. A good tour will contain many coherent tasks that together help the user achieve her complex task. A natural thing to do would be to construct a graph over the tasks and find a coherent set of tasks using this graph. Since, there are no edges between tasks; we needed to figure out a way to connect relevant tasks. Two tasks A and B are relevant for our purpose if users are likely to perform both A and B together.

We construct a graph $G = (T, E, w)$ where T is the set of all tasks in the dataset. $E = T \times T$ is the set of possible associated tasks. $w : E \rightarrow [0..1]$ is a function that assigns to every pair of tasks (i, j) a weight $w(i, j)$ representing their association strength.

There are many measures that have been used in the literature to assess the association between two variables. One of the widely used measures is the pointwise mutual information (PMI). The PMI of any two discrete variables x and y quantifies the discrepancy between the probability of their coincidence given their joint

distribution and the probability of their coincidence given only their individual distributions, assuming independence. Formally $pmi(x, y)$ can be defined as:

$$pmi(x, y) = \log \frac{p(x, y)}{p(x)p(y)}$$

The PMI value is 0 if the two variables are independent. Positive values of PMI indicate positive association while negative values indicate negative association.

Now that we have an association measure, we discuss how we create a variable corresponding to every task that appears in our dataset. We construct a set of records where every record consists of N binary attributes, where N is the number of tasks in the dataset. Every record corresponds to a user u and a time period t resulting in $m \times n$ records where m is the number of time periods and n is the number of users. An attribute is set to one if the corresponding user performed this task in the corresponding time period. We set the length of the time period to two days in our experiments which resulted in 30 overlapping time periods for March and 28 for February.

Given these records, we treat every binary attribute as a random variable and proceed with calculating the normalized pointwise mutual information. Before measuring NPMI, we discard all pairs that occurred for less than 100 times in the dataset. This is used to filter out infrequent and noisy associations.

3.4 Building Task Tours

Given the task graph we described in our previous section, let us revisit our overall objective which is building tours of tasks to guide users toward the completion of complex Web tasks. One way to do that would be to start at an arbitrary task and then move to the most strongly connected neighbor and so on.

However, this simple method does not necessarily yield a good tour. Suppose the user was searching transportation to a certain destination, she may get tours that resemble the following:

< *Travel Transportation; Maps; Business Directories* >
< *Travel Transportation; Weather; Read the News* >

These tours are certainly not useful because there is no strong association between looking for travel transportation and business directories, neither is there one between travel transportation and reading the news. Note that each transition when checked out of context is reasonable. For example many people need to check the weather when traveling and many people read the news after checking the weather, often on the same website.

The problem here is that links between pairs are coherent but the whole tour is not. To find more coherent tours, we need to extract groups of nodes that have denser connections internally and sparser connections between groups. This is a typical community finding problem where tasks are divided into coherent disjoint groups. The drawback of this approach is that every task can be a member of one group only. Common tasks like “*Checking the Weather*” or “*Check Online Maps*” can naturally belong to more than one tour. An alternative approach would be to extract overlapping communities from the task graph. Figure 2 shows how the task graph could be divided into overlapping communities representing tours. In the figure, nodes correspond to tasks, edges are the association between tasks, and the shaded regions are tours.

To find overlapping communities we used a modified clique template rolling technique based on [13]. Before we describe the algorithm, we define some of the terms that we will use.

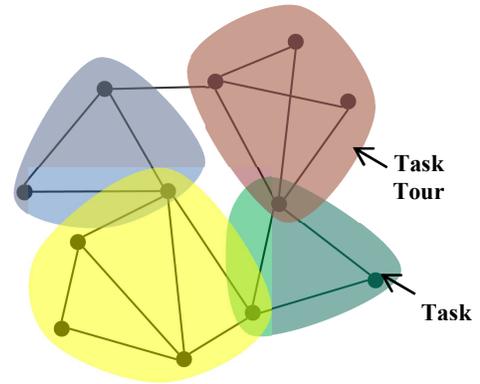


Figure 1: Overlapping communities in the task graph.

Our algorithm first extracts all cliques of size k (k -clique) from the task graph. Once the cliques are identified, a clique-clique overlap matrix is constructed. The clique overlap matrix is a symmetric matrix with each row and column representing a clique. The matrix elements are 1 if the corresponding cliques share $k - 1$ common nodes and 0 otherwise. We then run component analysis on the clique-clique overlap matrix to identify connected clique components. Every component is a maximal union of adjacent cliques. We select $k = 3$ which means that tours will be given by the union of triangles that can be reached from one another through a series of shared edges. To allow tours of size 2, we add to the pool of tours cliques of size 2 if they are not sub components of tours with larger sizes.

Obviously not all tasks can trigger a tour (e.g. “*Check the Weather*” should not trigger the travel tour). To solve this problem, we define the triggering score of a task t as the sum of the conditional probabilities of all other tasks given t . The task with the highest score is designated as the trigger.

4. TOUR EVALUATION

As a result, we evaluated our methods by building task tours using real data and leveraging human judgments to assess the validity and usefulness of our findings. We are interested in assessing the correctness and usefulness of the extracted tours. As a result, we conducted user studies to answer the following research questions:

RQ1. Validity of the task graph: Do the links in the task graph connect relevant tasks?

RQ2. Coherence of the tours: Are the tasks that together define the tours coherent?

RQ3. Utility of the tours: Do the tours cover most of the tasks related to the complex task?

We attempt to answer these questions by conducting a rating study using Amazon Mechanical Turk (MT) to rate the results. As is necessary with a study on a remote crowdsourcing platform, we took several precautions to maintain data integrity. We restricted annotators to those based in the US because our logs came from users based in the US. We restricted annotators to those who performed more than 1000 tasks and achieved more than 95% approval rate for their previous work. Moreover, we used hidden quality-control questions to filter out unreliable workers. Two annotators work on every instance and report their average scores.

4.1.1 The Task Graph

The objective of the first experiment was to evaluate the quality of the links in the task graph. We generated edges connecting related tasks using the following techniques:

- **Proposed Method:** As described in Section 3, we used normalized pointwise mutual information to assess the association between tasks. Two tasks were deemed relevant if the $npmi$ value was greater than 0.1. Generally, tasks are considered related if the value is greater than 0. We used a slightly higher threshold to filter out barely related tasks if any exists.
- **ODP Hierarchy:** The ODP data imposes a hierarchy over all categories. We used this hierarchy to find related tasks. Two tasks are considered related if they share a common ancestor other than the root node.
- **Random:** This is a simplistic baseline that links tasks in the task graph randomly. Note that this was restricted to the nodes that already existed in the task graph.

For every task in the task graph, we showed an external annotator the ODP label and a list of the top three most frequent domains related to this label the first month of log data, and asked the annotator to come up with one sentence describing the task. For example, when presented with:

Label: Recreation: Travel: Transportation: Air
URLs: www.delta.com, www.aa.com, www.united.com

The annotator devised the following description: “*Search for an air travel fare*”, which may then be generalized into the task “*Air Travel*” for inclusion in the tour.

The MT workers, hereafter referred to as “turkers,” were given a pair of tasks and asked to judge the relevance of the two tasks. The turkers had the description of every task and top 3 most frequent URL domains related to it. The workers were asked to judge the relevance of the tasks on a three-point scale:

- **Highly Relevant (rating=3):** Users interested in one task are very likely to be interested in the other as well.
- **Somewhat Relevant (2):** Users interested in one task are somewhat likely to be interested in the other as well.
- **Not Relevant (1):** Users interested in one task are unlikely to be interested in the other as well.

We used Cohen’s kappa to assess the annotator agreement. The value was 0.78 for the task relevance annotation task. This is considered an “excellent” agreement according to [6].

4.1.2 The Tours

The objective of the second experiment was to evaluate the quality of the generated tours generated with the following methods:

- **Proposed Method:** As described in Section 3, a clique template rolling technique was applied to the task graph.
- **Task Graph Path:** For every tour generated using the proposed method, we generated another tour that has the same size and starts with the same trigger. Additional tasks are added to the tour by selecting the task with the strongest connection to the current task, represented in the task graph as w . This process is repeated until we reach the desired size.
- **ODP Hierarchy Path:** Tours are generated using the same technique described in the previous point. The only difference is that tasks are linked based on the ODP hierarchy. Two tasks are considered relevant if they share a common ancestor. The strength of the connection is the number of levels between the common ancestor and the root. When selecting the next task, ties are broken by selecting the most popular task in our dataset.

The workers were given the tours one at a time. For each task, the description of the task and a list of the most popular URLs were shown to turkers. The tour trigger was marked. The rest of the tasks were ordered according to the triggering score ($Conf(A \rightarrow B)$ from earlier) or the tour generation order. We did not tell the

workers that the tasks are ordered, neither did we ask them to judge the ordering. The turkers were asked to judge the coherence of the tours on a three-point scale:

- **Highly Coherent (rating=3):** All tasks in tour are coherent.
- **Somewhat Coherent (2):** Most tasks in tour are coherent.
- **Not Coherent (1):** Most tasks in the tour are incoherent or could not understand the information need behind the tour.

They were also asked judge coverage also on a three-point scale:

- **Excellent Coverage (rating=3):** The tour covers all the possible aspects of the complex task.
- **Good Coverage (2):** The tour covers most of the possible aspects of the complex task, but some aspects are missing.
- **Bad Coverage (1):** Most aspects not covered / tour incoherent.

The kappa value for coherence and coverage were 0.52 and 0.5 respectively. This is considered “fair to good” agreement [6].

5. FINDINGS

We now present the findings of our study, broken out by the four research questions. We begin with the validity of the task graph.

5.1 Task-Graph Validity

Figure 2 shows the percentage of highly relevant, somewhat relevant, and not relevant task associations for the proposed method and the two baselines. We notice that almost 80% of the associations identified by the proposed method are either highly relevant or somewhat relevant. This percentage drops to 32% for the ODP hierarchy baseline and 18% for the random baseline. The results show that using the ODP hierarchy is not good enough for identifying associations and that the proposed method is doing a very good job in finding relevant task. A χ^2 test has shown that the difference between proportions is significant at the $p < 10^{-6}$ level.

The reason behind bad performance of the ODP hierarchy baseline is that many related tasks have different ancestors in the ODP hierarchy. For example “searching for school districts” and “real estate search” are clearly related but have completely different ancestors in the ODP hierarchy. On the other hand some tasks that belong to the same high level ODP category are not related (e.g., “Job Search” and “Real Estate Search” are both in *Business*).

5.2 Tour Validity

We evaluated the validity of the generated tours using several metrics. The first is “coherence” which evaluates how well the tasks in the tour belong together. The second is “coverage” which evaluates how well the tour covers different aspects of the complex task. The last is the quality of the tour trigger identification. The latter is important since all users exposed to task tours in the search engine will see the tour trigger, irrespective of how the other tasks in the tour are presented at the search interface.

Figure 3 compares the coherence of the proposed method and the two baselines. The figure shows that 92% of the tours generated by the proposed method are either highly or somewhat coherent. This percentage drops to 73% for the first baseline that uses the task graph but extracts paths with strong pairwise connections rather than communities. This shows that the value of the community finding step in identifying coherent tours. The percentage further drops for the third baseline to less than 50%. The third baseline does not use the task graph but uses the ODP hierarchy instead. This confirms that the task graph models the task association in a much better way compared to the ODP hierarchy.

Next we move to coverage where we tried to evaluate how well the tours cover the different aspects related to the complex tasks. The results are shown in Figure 4. Like coherence, the coverage

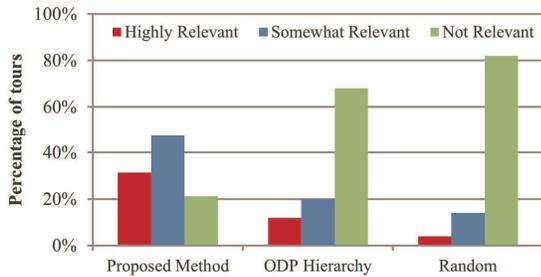


Figure 2: Task associations for our method and the baselines.

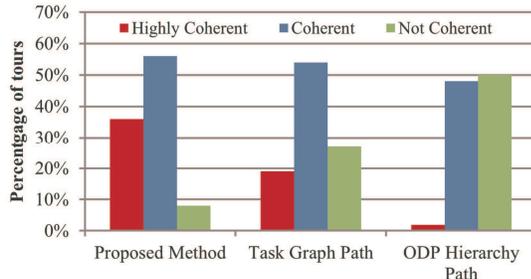


Figure 3: Tour coherence for our method and the baselines.

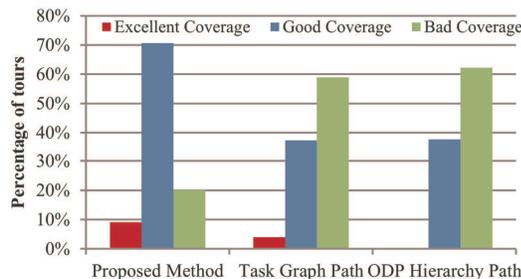


Figure 4: Tour coverage for our method and the baselines.

of the tours generated by the proposed method is much better than the coverage of the tours generated by the baselines. Unlike coherence, the percentage of tours with excellent coverage is considerably less with most of the mass falling under the “good coverage” category for the proposed method and the “bad coverage” category for the baselines. The performance with respect to coherence in general is better than the performance with respect to coverage. Better coverage may be achieved by using larger datasets that would allow less frequent tasks to appear with sufficient frequency that would allow more tasks to make it to the task graph. We also broke down the results into the long and short tours. As expected, longer tours have much better coverage than short tours. A χ^2 test has shown that the differences between proportions in Figure 3 and Figure 4 are significant at the $p < 10^{-6}$ level.

6. CONCLUSIONS

Search engines offer limited support for tackling complex search tasks. In this paper we have described and evaluated novel methods for automatically generating task tours to support the completion of complex tasks by outlining important elements. Our method automatically identifies the key tasks required, including determining the task from which the tour should start, based on task transition evidence mined from log data. We evaluated our methods using human judges and the results suggest that the tours generated by our method are of good quality. In future work we will incorporate task tours into SERPs to help users and evaluate their utility directly for different search scenarios. The indications from our study are that task tours will provide useful direction to users.

REFERENCES

- [1] Agichtein, E., Brill, E., and Dumais, S. (2006). Improving web search ranking by incorporating user behavior information. *SIGIR*, 19–26.
- [2] Bates, M.J. (1989). The design of browsing and berrypicking techniques for the online search interface. *Online Review*, 13, 5, 407–424.
- [3] Bilenko, M. and White, R.W. (2008). Mining the search trails of surfing crowds: identifying relevant websites from user activity. *WWW*, 51–60.
- [4] Bush, V. (1945). As we may think. *Atlantic Monthly*, 3(2): 37–46.
- [5] Chalmers, M., Rodden, K., and Brodbeck, D. (1998). The order of things: activity-centered information access. *WWW*.
- [6] Fleiss, J.L. (1981). Statistical methods for rates and proportions (2nd ed.). New York: John Wiley.
- [7] Guinan, C. and Smeaton, A.F. (1993). Information retrieval from hypertext using dynamically planned guided tours. *ECHT*, 122–130.
- [8] Hammond, N. and Allison, L. (1988). Travels around a learning support environment: rambling, orienteering, or touring? *SIGCHI*, 269–273.
- [9] Joachims, T. (2002). Optimizing search engines using click-through data. *SIGKDD*, 133–142.
- [10] Joachims, T., Freitag, D. and Mitchell, T. (1997). Web-Watcher: a tour guide for the world wide web. *IJCAI*, 770–775.
- [11] Kellar, K., Watters, C., and Shephard, M. (2007). A field study characterizing Web-based information-seeking tasks. *J. Am. Soc. Inf. Sci. Technol.*, 58(7), 999–1018.
- [12] O’Day, V. and Jeffries, R. (1993). Orienteering in an information landscape: how information seekers get from here to there. *INTERCHI*, 438–445.
- [13] Palla, G. et al. (2005). Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435(7043): 814–818.
- [14] Pandit, S. and Olston, C. (2007). Navigation-aided retrieval. *WWW*, 391–400.
- [15] Pirolli, P. and Card, S.K. (1999) Information foraging. *Psychological Review*, 106(4): 643–675.
- [16] Radlinski, F. and Joachims, T. (2005). Query chains: learning to rank from implicit feedback. *SIGKDD*, 239–248.
- [17] Reich, S. et al. (1999). Where have you been from here? trails in hypertext systems. *ACM Computing Surveys*, 31(4).
- [18] Shen, X., Dumais, S., and Horvitz, E. (2005). Analysis of topic dynamics in web search. *WWW*, 1102–1103.
- [19] Singhal, A. (2001). Modern information retrieval: a brief over-view. *Bulletin of the IEEE Computer Society Technical Communication on Data Engineer.*, 24(4): 35–43.
- [20] Trigg, R.H. (1988). Guided tours and tabletops: tools for communicating in a hypertext environment. *TOIS*, 6(4).
- [21] Wexelblat, A. and Maes, P. (1999). Footprints: history-rich tools for information foraging. *SIGCHI*, 270–277.
- [22] Wheeldon, R. and Levene, M. (2003). The best trail algorithm for assisted navigation of web sites. *LA-WEB*.
- [23] White, R.W. and Drucker, S.M. (2007). Investigating behavioral variability in web search. *WWW*, 21–30.
- [24] White, R.W. and Huang, J. (2010). Assessing the scenic route: measuring the value of search trails in web logs. *SIGIR*, 587–594.