Quicklink Selection for Navigational Query Results

Deepayan Chakrabarti Ravi Kumar Kunal Punera

701 First Avenue
Sunnyvale, CA 94089.
{deepay,ravikumar,kpunera}@yahoo-inc.com

ABSTRACT

Quicklinks for a website are navigational shortcuts displayed below the website homepage on a search results page, and that let the users directly jump to selected points inside the website. Since the real-estate on a search results page is constrained and valuable, picking the best set of quicklinks to maximize the benefits for a majority of the users becomes an important problem for search engines. Using user browsing trails obtained from browser toolbars, and a simple probabilistic model, we formulate the quicklink selection problem as a combinatorial optimizaton problem. We first demonstrate the hardness of the objective, and then propose an algorithm that is provably within a factor of $(1-1/e)$ of the optimal. We also propose a different algorithm that works on trees and that can find the optimal solution; unlike the previous algorithm, this algorithm can incorporate natural constraints on the set of chosen quicklinks. The efficacy of our methods is demonstrated via empirical results on both a manually labeled set of websites and a set for which quicklink click-through rates for several webpages were obtained from a real-world search engine.

Categories and Subject Descriptors

H.3.m [**Information Storage and Retrieval**]: Miscellaneous

General Terms

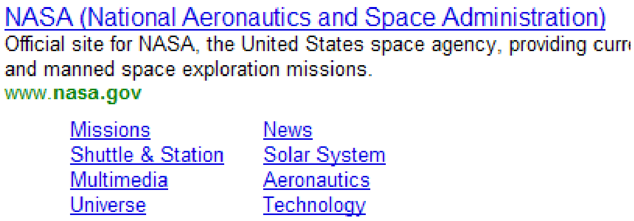
Algorithms, Experimentation, Measurements

Keywords

Quick links, Navigational queries, Toolbar data, Trails

1. INTRODUCTION

The distinction between the search box and the navigation bar in a browser is gradually vanishing. Search engines are increasingly being used as starting points for users to navigate a website: instead of typing `nasa.gov` in the navigation bar or using a bookmark, many people apparently prefer to type `nasa` in the search bar, presume the search engine returns `nasa.gov` as the first search result, and almost involuntarily click on the first result. The search engines cater to such user expectations by providing a variety of sophisticated navigational support as part of traditional web search: e.g., typing a url (intentionally or not) in the search box of some toolbars will automatically take the user to the website of the url. A particularly popular feature that is now available in all major search engines is the so-called *quicklinks* that are displayed for navigational queries.



NASA (National Aeronautics and Space Administration)
Official site for NASA, the United States space agency, providing curr
and manned space exploration missions.
www.nasa.gov

Figure 1: Quicklinks for `nasa`.

The best way to illustrate quicklinks is by an example. Figure 1 is the screenshot of the top result for the navigational query `nasa`. One can see eight hyperlinks displayed under the main result `nasa.gov`. All these links are from the NASA website and have been provided as navigational shortcuts that can directly take the user to selected points of interest within the NASA website. These links were chosen to serve the following purpose: suppose a typical user who goes to `nasa.gov` navigates to the "mission" webpage inside NASA. By providing this "mission" page as a quicklink, the user need not go through `nasa.gov` in order to get to this page; thus her navigation is made more efficient. Considering the valuable real-estate expended on showing the quicklinks, the large fraction of navigational queries, the high click-through rate noted on these quicklinks, and the associated implications on user experience, it becomes imperative for search engines to select them judiciously. This is the topic of our work.

Sources of quicklinks. Several and obvious sources of information are available to a search engine for selecting good quicklink candidates. The first is query and click logs: navigational queries along with a plenitude of reformulations and user click feedback on search results precisely pinpoint the webpages in a website that are considered interesting from a search point of view. The second is toolbar and user trail data: with the widespread adoption of browser toolbars, it has become possible to analyze navigation patterns within a website at a microscopic level and determine those webpages frequented by many users. The third is the webgraph obtained from hyperlinks: site-level link-analysis al-

Copyright is held by the author/owner(s).
WWW2009, April 20-24, 2009, Madrid, Spain.
.

gorithms can rank order the webpages within a site based on popularity measures. The fourth is bookmarking and social bookmarking websites such as `del.icio.us` and `digg.com`. Finally, sitemaps or server logs can sometimes be provided by the webmasters of the website.

None of these disparate sources of information available for selecting quicklinks is, however, perfect. The most clicked url in a newspaper website could correspond to a recent news article; clearly this is not a desirable quicklink since it is both fleeting and non-navigational. The most clicked urls in a company website could all be links discussing their most popular product, completely ignoring even slightly less popular products. Likewise, the highly visited webpages according to user trails might not be good quicklinks either; e.g., the "logout" pages or "shopping cart" pages typically have lots of trails going through them, but are not valuable as quicklinks since it does not make sense for a user to click on them from a search result page. Likewise, site-level templates might cause "privacy policy" and "copyright" pages to have high popularity; once again, a majority of users are indifferent about these pages. Thus, being a good quicklink is a combination of various attributes: how noticeable would be this webpage to the user's navigational goal when displayed as quicklink, how much traffic passes through it, and what is the tangible benefit to the user (say, in terms of fewer clicks or lower latency). Examples of good quicklinks include "store locator" in online shopping sites and "login" pages in Facebook/MySpace (since any user has to go through them before being able to do anything non-trivial).

Our contributions. In this paper we formalize the quicklink selection problem. Our formal framework is based on an objective defined over a set of user trails (i.e., the toolbar data). In our model, each webpage has a score of noticeability as a quicklink. Noticeability distills the probability that a user will recognize this quicklink as a shortcut to her eventual navigation goal. When a webpage is displayed as a quicklink, all trails that go through this webpage will benefit with probability proportional to the noticeability of this quicklink. With the remaining probability, the trails will benefit from the remaining quicklinks. Under this model, we formulate a simple cost function and a benefit-maximization objective. Even though this objective is NP-hard, under mild and plausible assumptions, it turns out to exhibit nice properties. In particular, we show that it is non-negative and submodular. Therefore, a greedy algorithm can obtain a provably approximate solution. We use this algorithm for quicklink selection.

While the greedy algorithm is applicable in the most general case, it is not amenable to certain natural constraints we may want to impose on the quicklinks. Motivated by this, we consider the problem of quicklink selection when the union of trails forms a tree. In this case, we show a dynamic programming algorithm to exactly solve the quicklink selection problem. Furthermore, this algorithm is robust enough to incorporate some simple constraints on the set of quicklinks. To be able to apply this algorithm in the general case, we propose a heuristic to extract, from a given set of trails, the best subset that will form a tree.

Empirical results using both manually labeled data and click-through data obtained from a real-world search engine demonstrate the efficacy of our approach. Our greedy algorithm for quicklink selection beats three strong competing baseline algorithms by a significant margin. We show that combining the trails obtained from toolbar logs with search logs can yield results that are better than those achieved by either individually. In fact, we obtain 22% improvement in terms of picking good quicklinks, and a 100% improvement in terms of not picking bad quicklinks, when compared against the best baseline algorithm.

Organization. The rest of the paper is organized as follows. Section 2 discusses the related work. Section 3 contains the formulation of the quicklinks selection problem. Section 4 contains our main algorithms for the problem — an approximation algorithm that works in the general case and an exact algorithm that works on trees. Section 5 contains our extensive experimental results on both manually labeled and post-hoc user-response data from real search engine traffic. Section 6 contains the concluding remarks.

2. RELATED WORK

The related work falls into three main categories. The first is the generic problem of organizing websites better for efficient user browsability. The second is the emerging area of data mining and analysis of website usage patterns, user trails, and the toolbar data. The third is the related work in the broad area of greedy algorithms and submodular maximization.

Website organization. Organizing websites based on user traffic has been studied for a long time. One of the earliest work in this area is that of Srikant and Yang [20]. They propose a simple algorithm to automatically find pages in a website whose location is different from where users normally expect to find them, using the assumption that users backtrack if they do not find the information where they expect it. Perkowitz et al. [18, 17] consider the problem of synthesizing an index page to facilitate user navigation of a website; one can think of their work as a method for automatic sitemap creation. Doerr et al. describe an extensible system to analyze weblogs and find patterns to improve the navigability of websites [8]. In particular, they consider the problem of providing shortcuts to popular targets. However, their approach is quite ad hoc.

A different line of research deals with carefully placing links in a website so as to optimize either the number of steps or the number of bytes transferred in order to reach the desired pages of a Web site. Many of the algorithms are based on recursive balanced partitioning of the tree [12, 11, 3]. Czyzowicz et al. consider the problem of enhancing the hyperlink structure in order to improve web performance [7].

Our work is different in two ways. First, the website organization work assumes that the algorithms have access to the entire webserver traffic. On the other hand, our work relies only on a partial and perhaps biased subset of the entire traffic, available through the toolbar. In addition, we also have access to the search traffic that hits pages within the website. Second, the website organization cares more about how to organize the website for maximizing user efficiency. We are more concerned about displaying the right set of quicklinks along with the website's homepage, in the context of search results. As we argued earlier, a high-traffic webpage in the site is not necessarily a good quicklink.

Website and toolbar analysis. There is a large body of work on visualizing and analyzing usage patterns in website. The focus can range from clustering [9, 10], to pattern discovery [4], and visualization of navigation patterns [5]. Mayr [14] developed a quantitative measure called the Web Entry Factor to aggregate common usage frequencies for webpages, where an entry means a website visit with an identifiable entry pattern (navigation type) from a logfile perspective.

Recently, Liu et al. proposed BrowseRank, a method that uses the toolbar data to create a user browsing graph and a continuous-time Markov chain based on the time spent along the edges [13]. The stationary probability of this chain gives a query-independent ranking of webpages. Bilenko and White considered the problem of using toolbar data to analyze post-search browsing behavior [1]. They show that post-search browsing behavior yields better relevance signal than mere search log clickthrough; see also [2]. White et al. used toolbar data to provide links to websites frequently visited by other users with similar information needs [22].

The problem of generating succinct titles for quicklinks and similar entry points was recently considered in [6]. There, the authors proposed a probabilistic model for title generation and used this model to generate short yet informative titles for quicklinks, in the context of the title of the root page.

While our work also analyzes toolbar data, the emphasis is on finding webpages that are best suited for display as quicklinks, and not on the analysis or visualization of user visit patterns. Also, our analysis combines search logs with toolbar data, yielding better results than if only the toolbar logs had been used.

Greedy algorithms. Submodular functions have been studied extensively in recent years due to their ability to encapsulate the "diminishing returns" effect that is so common in combinatorial covering problems. The greedy algorithm is a well-known way for maximizing submodular functions. Nemhauser and Wolsey [15, 16] showed that the greedy approach gives an $(1 - 1/e)$-approximation for maximizing a non-decreasing submodular function with a minimum value of zero. Our problem also has some connections to facility location [21]. However, unlike the general facility location problem, our problem has more structure and hence is amenable to greedy algorithms.

3. FORMULATION

In this section we present a formal model for selecting quicklinks.

3.1 Background

Fix a website W. Let V be the set of webpages from the website, where each webpage is associated with a url. We assume that the webpages in V have been de-duplicated, i.e., the urls corresponding to the webpages in V are canonical and any url on the website can be mapped to exactly one of these canonical urls. From here on, u will denote both the webpage and the canonical url corresponding to it. Let $r \in V$ be a distinguished node in the website, called the *root*; r corresponds to the website's homepage. Let E be the set of directed edges among V, given by hyperlinks: $(u, v) \in E$ if and only if there is a hyperlink from webpage u to webpage v. The nodes V and the edges E together form the graph for the website W.

A *trail* is a directed path in this graph. Formally, a trail $p = p_1, p_2, \ldots,$ is such that $p_1 = r$ and for each $i \geq 1$, we have $(p_i, p_{i+1}) \in E$. Note that a trail need not be simple, i.e., nodes on a trail can repeat and create loops. Let P be the (multi)set of all trails. Given a set $Q \subseteq V$ and a trail p, let the *projection* of Q with respect to p, denoted $Q|_p$ be the set of nodes in Q restricted to only the nodes in p.

Quicklinks. Let $k > 0$ be a given quicklink *budget*; typically $k \leq 8$. In the *quicklink selection problem*, the following is the situation of interest. The user issues some query for which the search engine's ranking function determines r, from website W, to be the top result for this query. The goal of the search engine is then to show k additional webpages in V, called *quicklinks*, under r. The question now is to select the best k webpages in V to show on the search result page, when r is the top search result[1].

User behavior in trails. We now state a simple model of user behavior with respect to trails. This model leads to a principled definition of our objective function. For a given website, we assume that each user has an information need that is expressed in the trail and served by the webpages in the trail; perhaps the end of the trail webpage delivers the information need. The webpages in the trail may or may not be known to the user a priori, but we assume that the user will quickly notice any webpage (perhaps by the page title or its incoming anchortext) that could lead to her desired information via some trail. We capture this notion probabilistically below.

Each node $u \in V$ has a value $\alpha(u) \in [0, 1]$ associated with it. We call this the *noticeability* of the node u. Intuitively, $\alpha(u)$ corresponds to the probability that, if u was displayed as a quicklink under r, then a user would notice u and judge it as belonging to a trail from r that would serve her information need. While ideally the noticeability is a user-specific value (e.g., depending on the user's familiarity with the website), for tractability reasons, we assume it is a user-independent global value.

Therefore, if u (with noticeability $\alpha(u)$) is shown as a quicklink under r, then with probability $\alpha(u)$ it will benefit all the trails that pass through u. The exact amount of benefit will depend on the position of u inside the trail, and is precisely captured by the ability to shortcut the trail by going directly from r to u, when u is displayed as a quicklink for r. If u is not recognized by the user of the trail (which happens with probability $1 - \alpha(u)$), then the benefits come from the remaining quicklinks.

Cost model and the objective. For every trail p, let $B'_p : V \to \mathbb{R}^+$ be a function that gives the benefit $B'_p(u)$ of having u as a quicklink with respect to the trail p, provided u is noticed by the user. The benefit $B'_p(u)$ can capture natural things such as the total expected time to reach u from r or the total number of clicks needed to reach u from r. We make two mild assumptions about this function.

(A1) If u is not on the trail p, then $B'_p(u) = 0$. In other words, nodes not part of the trail do not contribute to the benefit.

[1] Quicklinks could also be shown only when the original query was navigational; such variations do not change our formulation.

(A2) For u, v on the trail p, $B'_p(u) \geq B'_p(v)$ if and only if v is closer to r than u. In other words, as a quicklink, u benefits p more than v if it occurs further down the trail from the root r.

Now, given a set Q, the benefit of the set on trail p is calculated as follows. Let $q_\ell \in Q$ be the node that occurs last in p. Then, the *effective benefit* is defined to be

$$\begin{aligned} B_p(Q) &= \alpha(q_\ell) \cdot B'_p(q_\ell) + (1 - \alpha(q_\ell)) \cdot B_p(Q \setminus \{q_\ell\}) \quad (1) \\ B_p(\phi) &= 0 \end{aligned}$$

It is easy to see from (A1) that it suffices to consider the projection of Q with respect to p.

Observation 1. *For all p and Q, $B_p(Q) = B_p(Q|_p)$.*

Using (A2) and the fact that the range of $\alpha(\cdot)$ is $[0, 1]$, we have

Observation 2. *For all p and Q, $B_p(Q) \leq B'_p(q_\ell)$.*

Now we can formally state the quicklink selection problem.

Problem 3 (Quicklink selection). *Given a budget k, find a set of nodes $Q \subseteq V$ with $|Q| \leq k$ to maximize*

$$B(P, Q) = \sum_{p \in P} B_p(Q). \quad (2)$$

3.2 Hardness

In this section we show the hardness of the quicklink selection problem.

Lemma 4. *The quicklink selection problem is NP-hard.*

Proof. This is a reduction from the hitting set problem. In the hitting set problem, we are given a collection $\mathcal{C}$ of subsets of a universe S and an integer k and asked: is there a subset $S' \subseteq S$ with $|S'| \leq k$ such that S' contains at least one element from each subset in $\mathcal{C}$? Given an instance of the hitting set problem, we create an instance of the quicklink selection problem as follows. First, we let $\alpha(s) = 1$ for each $s \in S$. We then assume an arbitrary ordering on the elements in S; this naturally orders the elements in each subset $C \in \mathcal{C}$, giving rise to a trail p_C. For each trail p_C, we set $B_{p_C}(\cdot) \equiv 1$. The crucial point to note is that for a given trail p_C, (1) takes value 1 if and only if at least one node from p_C is included in the quicklink solution. Given this, it is easy to see that the hitting set instance has a solution if and only if the objective of the quicklink selection problem has value exactly $|\mathcal{C}|$. □

4. ALGORITHMS

4.1 A greedy algorithm

In this section we obtain a greedy approximation algorithm for the quicklink selection problem. We do this by showing that the quicklink objective is non-decreasing and submodular.

First, we state the notion of submodularity.

Definition 5 (Non-decreasing submodularity). *Let U be a finite set. A function $f : 2^U \to \mathbb{R}$ is* non-decreasing *and* submodular *if (i) non-negativity: $f(\phi) = 0$ and $f(\cdot) \geq 0$, (ii) monotonicity: $f(X) \leq f(Y)$ when $X \subseteq Y \subseteq U$, and (iii) submodularity: $f(X) + f(Y) \geq f(X \cap Y) + f(X \cup Y)$, $\forall X, Y \subseteq U$, or equivalently, (iii') $f(X \cup \{u\}) - f(X) \geq f(Y \cup \{u\}) - f(Y)$, $\forall X \subseteq Y \subseteq U$.*

Submodularity, a combinatorial analog of convexity, captures the diminishing returns property. Next, we show that the effective benefit function is non-negative and non-decreasing.

Lemma 6. *For a given set P, the function $B(P, Q)$ is non-negative and non-decreasing in Q.*

Proof. It is clear from (1) and (2) that the function is non-negative. We now show it is non-decreasing by showing it point-wise, i.e., for each p, the function $B_p(Q)$ is non-decreasing.

Let $Q \subseteq V$ and let $u \in V \setminus Q$. Let $q_1, \ldots, q_\ell \in Q$ be such that the trail p is of the form

$$p = \ldots, q_1, \ldots, q_2, \ldots, q_i, \ldots, u, \ldots, q_{i+1}, \ldots, q_\ell, \ldots.$$

Notice that u partitions Q into two sequences; call them $Q_u^- = \{q_1, \ldots, q_i\}$ and $Q_u^+ = \{q_{i+1}, \ldots, q_\ell\}$. Also, notice that $B_p(Q)$ depends only on $q_1, \ldots, q_\ell$ and

$$B_p(Q) = \alpha(q_\ell) B'_p(q_\ell) + (1 - \alpha(q_\ell)) B_p(Q_{q_\ell}^-).$$

A similar expression can be written for $B_p(Q \cup \{u\})$.

Let $A(Q_u^+) = \prod_{q \in Q_u^+} (1 - \alpha(q)) \geq 0$. Now,

$$\begin{aligned} & B_q(Q \cup \{u\}) - B_p(Q) \\ &= A(Q_u^+) \cdot \big(\alpha(u) B'_p(u) + (1 - \alpha(u)) B_p(Q_u^-) - B_p(Q_u^-)\big) \\ &= A(Q_u^+) \cdot \big(\alpha(u) B'_p(u) - \alpha(u) B_p(Q_u^-)\big) \\ &= A(Q_u^+) \cdot \alpha(u) \cdot \big(B'_p(u) - B_p(Q_u^-)\big) \quad (3) \\ &\geq A(Q_u^+) \cdot \alpha(u) \cdot \big(B'_p(u) - B'_p(q_i)\big) \\ &\geq 0, \end{aligned}$$

where the first inequality follows from Observation 2 and the second inequality follows from (A2) since $B'_p(u) \geq B'_p(q_i)$. □

Lemma 7. *For a given set P, the function $B(P, Q)$ is submodular in Q.*

Proof. Let $Q \subseteq R \subseteq V$. We need to show (iii') in Definition 5, i.e., for any $u \in V \setminus Q$, $B(P, Q \cup \{u\}) - B(P, Q) \geq B(P, R \cup \{u\}) - B(P, R)$. As before, we will show this point-wise, for each $p \in P$. We will use the same notation as in the proof of Lemma 6.

If $u \in R$, the proof follows Lemma 6. So, we assume $u \notin R$. Now, as in (3),

$$\begin{aligned} & B_p(Q \cup \{u\}) - B_p(Q) \\ &= A(Q_u^+) \cdot \alpha(u) \cdot \big(B'_p(u) - B_p(Q_u^-)\big) \\ &\geq A(R_u^+) \cdot \alpha(u) \cdot \big(B'_p(u) - B_p(Q_u^-)\big) \\ &\geq A(R_u^+) \cdot \alpha(u) \cdot \big(B'_p(u) - B_p(R_u^-)\big) \\ &= B_p(R \cup \{u\}) - B_p(R). \end{aligned}$$

Here, the first inequality follows from the definition of $A(\cdot)$ and the fact that $Q_u^+ \subseteq R_u^+$. The second inequality follows since $Q_u^- \subseteq R_u^-$ and $B_p(R_u^-) \geq B_p(Q_u^-)$ using Lemma 6. □

Thus we have established that the effective benefit objective for quicklink selection is non-decreasing and submodular. A natural way to optimize covering problems, where the function is non-decreasing and submodular, is the *greedy* approach. Start with an empty set and iteratively build the solution. At each iteration, select the element with the highest incremental benefit to the current solution and add it to the current solution. We now present the algorithm QL-Alg.

```
Algorithm QL-Alg
  Set $Q \leftarrow \emptyset$.
  While $|Q| \leq k$ do,
    Find $u \in V \setminus Q$ that maximizes $B(P, Q \cup \{u\}) - B(P, Q)$.
    Set $Q \leftarrow Q \cup \{u\}$.
  Return $Q$.
```

It is easy to see from the results of Nemhauser, Wolsey, and Fischer [15, 16] that since the effective benefit function is non-decreasing and submodular, the above greedy algorithm is a provably good solution.

Theorem 8. *Algorithm* QL-Alg *produces a* $(1 - 1/e)$*-approximation to the quicklink selection problem.*

4.2 Algorithms for trees

In the previous section we described QL-Alg, a greedy algorithm that incrementally picks quicklinks for a website so as to maximize the benefit gained over all the trails for that website. While the optimization function has nice theoretical properties that allow us to bound the performance of QL-Alg, there are several desiderata that are not handled. These typically take the form of constraints on the solution, which, unfortunately, are either not easily expressible under the previous framework or destroy the submodularity property that is essential for bounding the performance of the greedy algorithm. Next, we discuss two such constraints and propose methods to find quicklinks under these constraints.

The first constraint is the presence of parent-child webpages as quicklinks. It can be confusing to the user to show two webpages, one of which is conceptually subsumed by the other, as two separate quicklinks. E.g., for the website `chase.com`, it may not be desirable to display both `Find-us` and `ATM-locator` as quicklinks, where the latter is an in-neighbor of the former in the graph. The second constraint is homogeneity: it is aesthetically unattractive to display some quicklinks that are very broad high-level webpages while some others are deep down in the website. For example, the former could be pages like `News` or `Weather` in the `bbc.co.uk` (yielding small benefits for a large segment of users), while the latter might be very commonly accessed webpages like `Sports/Football/English-Premier-League` (yielding large benefits for a few users). Displaying them together as quicklinks can confuse the user, who might pay just a fraction of second attention to each quicklink.

Constrained quicklink selection. The above examples motivate the need for *constrained* quicklink selection. We consider pairwise constraints on the set of quicklinks that can be obtained by an algorithm, e.g., it is possible to specify a set of all pairs of webpages in the website that cannot both occur together as quicklinks. Note that pairwise constraints can capture both parent-child and homogeneity constraints. Unfortunately, at this level of generality, the problem becomes hopelessly intractable.

Lemma 9. *The quicklink selection problem, with pairwise constraints on the solution, is as hard as independent set.*

On the other hand, parent-child and homogeneity constraints are nicely handled if the graph induced by the trails in P form a tree, rooted at r. We show that in this case, we can actually solve the quicklinks selection problem exactly.

Dynamic programming on trees. In the following, we assume that the set of trails induce a tree T, rooted at r. The main idea behind the dynamic program is that selecting k quicklinks at any node in the tree has exactly one of the two options: either select the node itself as a quicklink and select $k - 1$ quicklinks from its children or select all the k quicklinks from its children. At this point, it is easy to introduce parent-child and homogeneity constraints on the solution. For simplicity, we will present the algorithm without the constraints.

First, we will convert the tree T to a binary tree. This is done by replacing any internal node u of degree $d > 2$ with children $u_1, \ldots, u_d$ by a binary tree of depth $\lg d$ with leaves $u_1, \ldots, u_d$. All the internal nodes of this binary tree will have noticeability score of zero; hence, they will never get selected in any quicklink solution.

Let P_u be the set of all trails that end at a node u. Let $C(u, Q, k)$ be the best effective benefit when Q is the current set of quicklinks and there can be at most k quicklinks in the subtree rooted at u (including u itself). Let u_1, u_2 be the children of u. The recurrence of the dynamic program, QL-Alg-Tree, is given by

$$C(u, Q, k) = \max \begin{cases} B(P_u, Q) + \max_{\ell=1}^{k}(C(u_1, Q, \ell) \\ \quad + C(u_2, Q, k - \ell)) \\ B(P_u, Q \cup \{u\}) \\ \quad + \max_{\ell=1}^{k-1}(C(u_1, Q \cup \{u\}, \ell) \\ \quad\quad + C(u_2, Q \cup \{u\}, k - \ell - 1)), \end{cases}$$

and the base cases for any leaf node u are given by $C(u, Q, 0) = B(P_u, Q)$ and $C(u, Q, k) = B(P_u, Q \cup \{u\})$, for $k \geq 1$. The dynamic program is invoked as $C(r, \emptyset, k)$. The dynamic program can be implemented in time $O(k^2 \cdot n \cdot d)$, where d is the depth of the tree T and n is the number of nodes in T.

Lemma 10. *The algorithm* QL-Alg-Tree *solves the quicklink selection problem for trees.*

Extracting a tree from trails. While the above algorithm is very efficient and can accommodate homogeneity and parent-child constraints, it assumes that the set of trails form a tree. This is a strong assumption and is general may not be satisfied, especially for websites with lots of traffic. Next, we consider the problem of extracting the best subset of trails from P so that they form a tree. Unfortunately, this problem is once again hopeless.

Lemma 11. *Given a set of trails, finding the maximum subset that induces a tree is as hard as independent set.*

Proof. Consider an instance (V, E) of the independent set problem. Let the nodes in V be ordered. We construct a trail p_u for each $u \in V$. The trail consists of the directed path $r, v_1, \ldots, v_d$ where r is a special node and $v_1, \ldots, v_d$ are the neighbors of u in E. It is easy to check that finding the maximum subset of trails in $\{p_u\}_{u \in V}$ that form a tree is equivalent to finding the maximum independent set in (V, E). □

Given this intractability, we resort to the following simple heuristic. Following [19], let the *value* $v(p)$ of a trail p be given by $v(p) = \ell(p)/(1 + n(p))$, where $\ell(p)$ is the length of p and $n(p)$ the number of other trails that intersect with p (i.e., there is at least one webpage w such that both p and the intersecting trail arrive at w, but by following links from different pages). Note that $\ell(p)$ denotes the importance of p

and $n(p)$ the number of trails that would have to be dropped if p is retained in the tree; the formula for $v(p)$ captures both these factors. Thus, a high $v(p)$ implies that either the trail is long and hence important, or that We then order the trails by decreasing order of $v(p)$ and greedily pick the next trail, which is added to the current set of trails as long as it maintains a tree.

5. EXPERIMENTS

In this section we compare the performance of our approach, QL-ALG, against strong baselines. First, we describe our experimental setup. We describe in detail the different datasets and measures we use to evaluate our approach, and give the rationale for making these choices. Then we break down the performance comparison along lines that highlight the various aspects of the quicklinks selection problem. We show in this section that across a wide array of ground truths and performance measures our approach, QL-ALG, significantly outperforms several competitive baselines.

5.1 Methodology

We first discuss the data and the implementation of the recognizability function that we used in our experiments. Then, we discuss the various datasets and measures used by our evaluation setup. Finally, we describe the baselines that QL-ALG is compared against.

5.1.1 Data and Implementation Details

We extracted data on user trails from Yahoo! toolbar logs collected over a period of three months. Each trail consists of a series of clicks by the user such that (1) any two successive clicks are at most 10 minutes apart[2], and (2) none of the clicks was on the *Back* button of the browser. The intuition is that the entire trail, from the entry point into the website up to the click on the *Back* button, corresponds to a *focussed* browse for some information on the website, and these are exactly the trails that users might follow after being led to the website from the search results page. Pressing a *Back* button or waiting on a page for more than 10 minutes are both indications that focussed browsing might have ended, and further clicks by that user might not be as relevant for quicklink selection. Note that some trails might begin at pages other than the website homepage; in such cases, we prefix the trail with the homepage.

In Section 3 we introduced the *noticeability* function that models the propensity of a user to notice and click on a quicklink that is of interest to her. In our experiments we estimate the noticeability $\alpha(u)$ of URL u in website W from the number of clicks $c(u)$ that u receives on the search engine's results page. The intuition is that a webpage that gets searched for and clicked often is likely to possess whatever features make URLs attractive to users. In addition, we have a parameter β that controls the amount of influence that noticeability exerts on the objective function of QL-ALG. In particular,

$$\alpha(u) = \left(\frac{c(u)}{\sum_{i \in V} c(i)}\right)^{\beta},$$

where V is the set of all URLs in website W.

5.1.2 Ground Truth and Performance Measures

In order to evaluate QL-ALG we use human editors to label the true quicklinks within websites. For these experiments, we randomly selected a set of 1257 websites from the set of websites that are searched for most often on the Web. This bias is essential for our evaluation since these are exactly the type of websites for which our system will be required to display quicklinks. From these websites we created the following datasets.

BEST SET OF QUICKLINKS. We tasked three human editors with picking the best set of quicklinks for 100 websites selected from the set mentioned above. The editorial guidelines called for *picking a set of 8 or fewer quicklinks that would be useful for a large fraction of the users coming to the website*. For this purpose the editors were allowed to browse the website, scan the web-master provided sitemaps etc. Though the editors had access to other sources of information, like the queries which commonly result in clicks on the website, these cues were seldom used while labeling. Because of this, the editors were often unable to determine the intention of a typical user while visiting the website, and were significantly biased by the website structure. The editors were allowed to pick less than 8 quicklinks when they found few good candidates.

MEASURE: PRECISION/RECALL. Since the ground truth described above finds the best set of quicklinks, we can evaluate the ability of QL-ALG to find exactly the same set of quicklinks. For this we employ the PRECISION and RECALL metrics, which are standard in information retrieval research. The PRECISION of a solution is the fraction of quicklinks predicted by it that also belong to the BEST SET OF QUICKLINKS ground truth. RECALL of a solution is the fraction of quicklinks in the BEST SET OF QUICKLINKS ground truth that are also predicted in the solution. The PRECISION and RECALL measures are commonly reported in a combined fashion via F-MEASURE, which is the harmonic mean of the two quantities.

As described above, obtaining the BEST SET OF QUICKLINKS for websites is a time consuming process. In order to evaluate our approach more extensively we created the following dataset.

INDIVIDUAL QUICKLINK JUDGMENTS. We provided a set of 300 websites and 15 "prospective" quicklinks candidates per website to a set of 22 editors. The editors were then asked to consider each link independently, and rate its fitness as a a quicklink. The guidelines for determining fitness of a candidate URL as a quicklink were the same as described above. Notice that since the editors were not trying to find the best 8 quicklinks for a website, the editorial process was much faster and a total of 4500 URLs were judged this way. On the flip side, since the editors were making decisions on the level of individual URLs, there was more variability in the judgments. To counteract this effect, each URL was labeled by 3 editors and the majority label was used. Also, issues like parent-child and homogeneity constraints in the positive set of quicklinks were not evaluated.

MEASURE: FRACNEGATIVES. Given the local/incomplete nature of the INDIVIDUAL QUICKLINK JUDGMENTS, we cannot punish an approach for not selecting URLs that have been rated as good quicklinks by editors. This is because

[2] The time limit of 10 minutes was set arbitrarily, but results using other time limits were similar

the approach could be finding quicklinks that were not evaluated by the editors (they were given a set of 15 quicklink candidates to evaluate). Moreover, these predicted quicklinks might be better overall than those rated as positive by the editors. Hence, the only aspect of performance we can evaluate is the ability of an approach to avoid selecting quicklinks that are rated as negative by the editors. This motivates the fracNegatives measure, which is the fraction of websites on which a URL rated as negative by the editors was picked as a quicklink by the algorithm. A lower value of fracNegatives indicates better performance.

Live Traffic Judgments. Here we use the click-through rates on the quicklinks that are currently displayed by the Yahoo search engine as the ground truth. The click-through rates were obtained for 1043 websites, while ensuring that there are enough number of views so that the click-through rates were robust. We ignored biases resulting from quicklink position: we believe these were negligible owing to the quicklink presentation (see Figure 1). Note that unlike the two ground truth datasets described earlier that scored quicklinks as a set and as individuals, Live Traffic Judgments result in a ranking of the quicklinks displayed for a website. We use this ranking for the evaluation measure described next.

Measure: fracCorrectSubsets. We use this measure to determine how well our objective function ranks sets of quicklinks. Note that using click-through rates from the Live Traffic Judgments we can obtain an ordering on subsets of the 8 quicklinks shown. For instance, given two subsets of the displayed quicklinks, each of equal size (say, 4), we can tell which of two subsets is better in terms of the sum of their click-through rates. Similarly, for these subsets of quicklinks we can also measure how they score using the objective function of QL-Alg (see Equation 2) or other baseline approaches. Using the fracCorrectSubsets measure we compute the fraction of such pairs of subsets that are ranked in the same way by Live Traffic Judgments and the quicklink selection objective function. One thing to note is that we obtain one measure of fracCorrectSubsets for each subset size: we use subsets of sizes 4,5,6,7. In each case a higher values indicate better performance.

5.1.3 Baselines

Here we describe the various baselines we compare QL-Alg against.

TopClicked. While finding the quicklinks for a website our approach heavily uses information about URLs within the site that are returned as results by a search engine and are then clicked by users. We use this information to construct the TopClicked baseline. Hence the TopClicked baseline predicts the top-8 clicked URLs within the site as quicklinks. While computing the number of times a URL in the site is clicked on the search results page, we restrict ourselves to queries that have a navigational intent. We define these queries as those that contain the website name in them. For example, clicks on URL `www.nasa.gov/topics/solarsystem` will only be counted for queries which contain the word "nasa". This helps the baseline identify URLs that are specifically sought in the context of the website and not just in general on the Web. Since the quicklinks problem relates to identifying URLs that have to shown on the search results page, as we shall see later, this simple baseline is very competitive in terms of performance.

Approach	F-measure	Precision	Recall
QL-Alg	**0.22**	**0.18**	**0.28**
PageRank	0.18	0.17	0.19
TopVisited	0.17	0.16	0.18
TopClicked	0.14	0.13	0.15

Table 1: Performance of various approaches evaluated on the Best Set of Quicklinks ground truth.

TopVisited. One of the sources of information used by our approach in order to select quicklinks for a website is the paths that users take while browsing the website. If lots of users' browsing paths involve a certain web page then chances are that it is an important one. The TopVisited baseline uses just this information to select quicklinks: it predicts the top-8 visited URLs in a website as its quicklinks. As a part of developing this baseline, we experimented with including the "time spent" on a particular page as a feature in selecting quicklinks. However, our initial experiments indicated that time spent was not a very accurate indicator of page importance: it is not clear that people are reading a web page the whole time it is open in the browser. Hence, for the TopVisited baseline we regard each visit to a web page as a single unit. These tweaks as well as the original signal in the data make this baseline extremely competitive.

PageRank. The PageRank baseline was constructed in the following manner. We take all the user trails and construct a (weighted) graph on the set of web pages as nodes along with a supernode s. Each trail $p = p_1, \ldots, p_\ell$ defines a directed path in the graph, starting at the supernode s, going through the nodes $p_1, \ldots, p_\ell$ in succession, and ending at s. Now, we compute the stationary distribution, viewing the weighted graph as the transition matrix of a random walk, and output the top ranked nodes (other than s and the root r of the website) as quicklinks. This baseline was constructed in order to see how well natural PageRank-like mechanisms work for quicklinks, if the transition probabilities were not uniform but computed from actual trails.

5.2 Comparison on Editor Labeled Data

In this section we evaluate our algorithm and baselines on the Best Set of Quicklinks ground truth. The averaged results of running QL-Alg and other competing algorithms on the websites in the ground truth are shown in table 1. The QL-Alg approach was run with the β parameter set to 2 (Section 5.4) and all approaches were required to fetch 8 quicklinks. As mentioned earlier we use the Precision and Recall measures to report results, and these numbers are summarized as the F-measure.

There are several observations to be made about the results. First, the QL-Alg algorithm outperforms all the other baselines in terms of F-measure; it scores 22% higher than the second-best baseline, PageRank. The baseline approaches are competitive, especially in terms of Precision, however, QL-Alg outperforms the baselines by a wide margin in terms of Recall. Note that the Precision and Re-

QL-Alg	PageRank	TopVisited	TopClicked
0.22	0.39	0.41	0.42

Table 2: Performance of various approaches in terms of fracNegatives on the Individual Quicklink Judgments ground truth.

call values differ because the number of quicklinks in the ground truth are not always 8, while the algorithms always predict 8.

Second, the approaches that employ user browsing patterns within the website in selecting quicklinks outperform the TopClicked baseline. The TopClicked baseline only knows the URLs where users enter the website via a search engine and not where the users navigate afterwards. This indicates that using the information present in the navigational patterns is useful in discovering the important pages within a website.

Finally, even though QL-Alg outperforms the baseline approaches, it scores overall low values in terms of Precision and Recall measures. The principal reason for this is the inability of the editors to determine the intention of an average visitor to the website. The editors did not access traffic related features of a website that the automated approaches had access to. Hence, the editors were mostly biased by the web site structure in selecting the quicklinks. In fact, the quicklinks found by QL-Alg were occasionally superior to the ones selected by the editors, in terms of helping with the average website user's browsing. Another reason for low scores in Table 1 is that there are often more than 8 good quicklinks for any particular website. Hence, quicklinks predicted by QL-Alg might be good even though they do not match those identified by the editors. A more important evaluation is to check that QL-Alg avoids bad quicklinks, and this leads to the next experiment.

5.2.1 Avoiding Bad Quicklinks

In this experiment we evaluate the output of QL-Alg and the various baselines in terms of the fracNegatives measure described in Section 5.1.2. Recall that the fracNegatives measure attempts to count the fraction of websites for which an approach outputs at least one quicklink that is rated negative by editors as part of the Individual Quicklink Judgments ground truth. The results for this experiment are presented in Table 2.

The results show that for 22% percent of the websites, QL-Alg places at least one quicklink that is considered bad by editors. The corresponding numbers for the baselines are almost 100% higher. Note that the fracNegatives measure can be made arbitrarily small if the URLs selected as quicklinks by an approach have no intersection with the ones that have been labeled by editors. Hence, the results in Table 2 have to considered in relation to the numbers in Table 1. The fact that QL-Alg outperforms all competing algorithms in both tables indicates it shows valid quicklinks for the vast majority of websites it is run on.

5.3 Comparison on Data from Live Traffic

In this section we compare the performance of QL-Alg with the baselines in terms of the fracCorrectSubsets measure described in Section 5.1.2. The fracCorrectSubsets measure is useful to compare how well the underlying objective ranks subsets of quicklink candidates for which we know click-through rates on live search traffic. The underlying objective for the QL-Alg algorithm is given in Equation 2; for the TopClicked, TopVisited, and PageRank baselines the underlying objective value for a subset is the sum of goodness values for individual items in the subset. We compute separate fracCorrectSubsets measures for subsets of sizes 4,5,6, and 7. These measures for the various quicklink selection approaches are plotted in Figure 2.

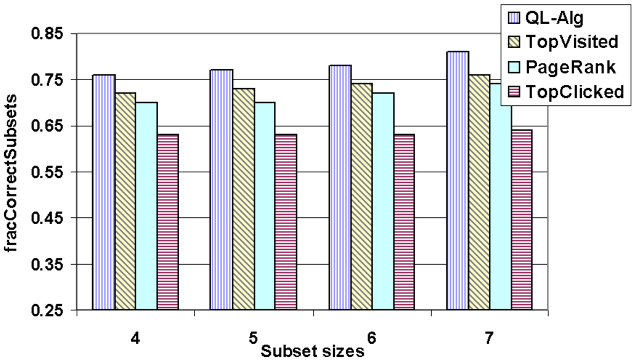


Figure 2: Performance of QL-Alg and competing algorithms on the fracCorrectSubsets measure plotted for various subset sizes.

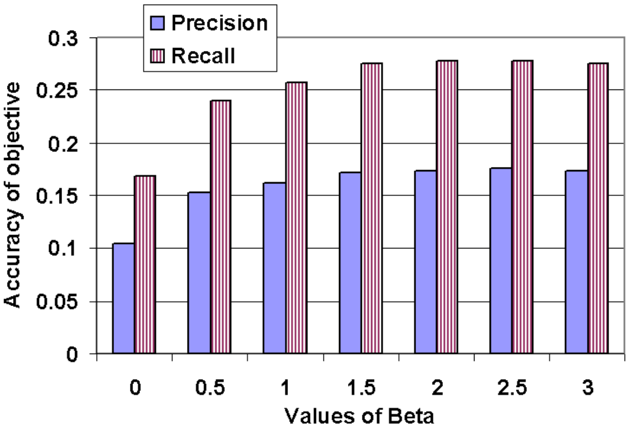


Figure 3: Performance of QL-Alg with varying values of β in terms of Precision and Recall measured over the Best Set of Quicklinks ground truth.

As we can see from the figure QL-Alg outperforms all competing approaches in terms of the fracCorrectSubsets measure for all subset sizes. The increase in accuracy of QL-Alg over the most the competitive baseline, TopVisited, is on average 5% for all the subset sizes. Moreover, the order of the baseline performance amongst themselves mimics those obtained in experiments performed in the previous section. This further corroborates our conclusion that information latent in the navigation patterns of users on a web site is particularly useful for the task of finding quicklinks, or important web pages, on the site.

5.4 Effect of Noticeability

In this section we evaluate the effect of the noticeability function (specifically, the β parameter) on the accuracy of the objective function of QL-Alg.

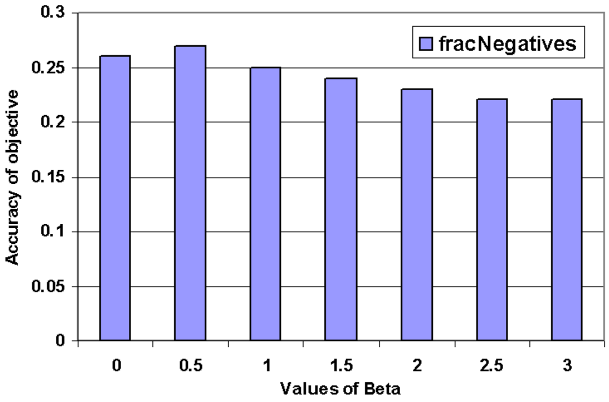


Figure 4: Performance of QL-ALG with varying values of β in terms of FRACNEGATIVES measured over the INDIVIDUAL QUICKLINK JUDGMENTS ground truth. Note that a lower value of FRACNEGATIVES is better.

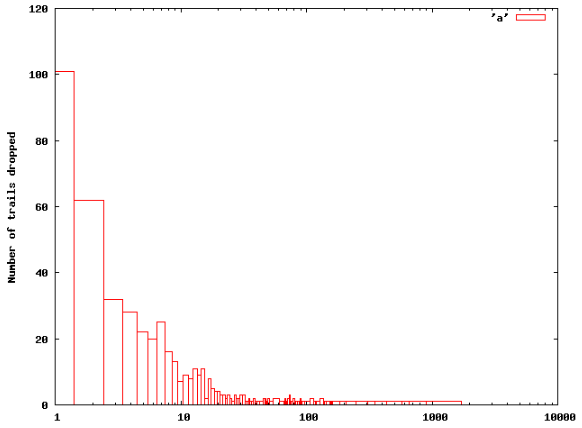


Figure 5: Distribution of trails lost while selecting trails that conform to a tree structure.

In Figures 3 and 4 we plot the performance of QL-ALG when the β parameter is varied over a wide range of values. As we can see from the plots increased emphasis on *Noticeability* increases the performance of our approach in terms of all the measures: note that lower values of FRACNEGATIVES indicate better performance. Moreover, once *Noticeability* achieves a significant amount of influence, increase the β parameter further does not result in a further increase in performance.

5.5 Experiments with Tree-Structured Trails

In Section 4 we described a process by which we select a subset of trails such that they conform to a tree structure. We also designed exact algorithms that could further be used to enforce parent-child and homogeneity constraints on the set of quicklinks selected. In this section we evaluate the performance of our tree-based quicklink selection approach.

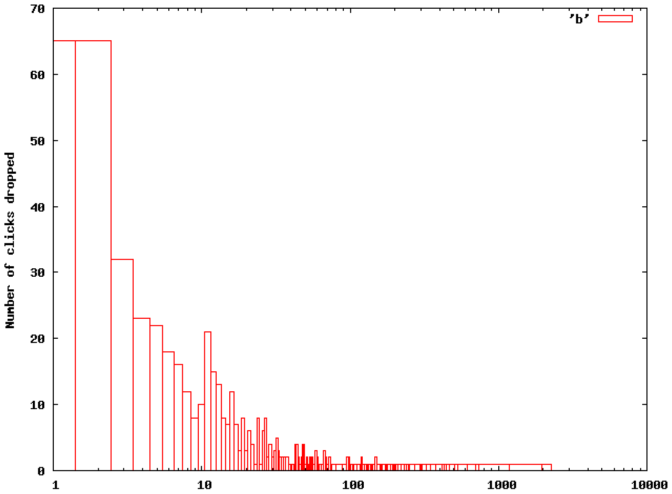


Figure 6: Distribution of clicks lost while selecting trails that conform to a tree structure.

Subset size	4	5	6	7
Full Trails	0.76	0.77	0.78	0.81
Tree Trails	0.78	0.79	0.80	0.83

Table 3: Performance of the QL-ALG objective function (Equation 2) in ranking subsets of quicklink candidates as evaluated by the FRACCORRECTSUBSETS measure.

In general, user navigation trails form a graph over the set of URLs in the website. In order to ensure that trails conform to tree-based navigation on the website we have to exclude some trails, and consequently lose some user clicks. In Section 4 we presented algorithms and heuristics to minimize the amount of loss of trails and clicks. In Figures 5 and 6 we plot the distribution of the loss in terms of excluded user trails and clicks when our algorithms were run on the set of 1257 ground truth websites. As we can see the distribution is heavily biased towards very few trails and clicks dropped, pointing to the efficacy of our heuristic.

Another measure of loss of information is the resulting loss in the ability of the objective function in ranking subsets of quicklink candidates. Hence, we repeat the experiment performed previously in Section 5.3 that plots the FRACCORRECTSUBSETS measure over the LIVE TRAFFIC JUDGMENTS ground truth set. In Table 3 we present the accuracy of the objective function on the tree-structured trails as computed by the FRACCORRECTSUBSETS measure for different sizes of subsets. For comparison purposes, we also present the corresponding values of the QL-ALG obtained in experiments in Section 5.3. For both sets of values the parameter β is set to 2. As we can see the loss of trails and clicks has no adverse effect on the power of the objective function in ranking sets of quicklinks candidates. In fact, the accuracy of the objective function seems to have increased once we streamlined the trails into a tree-structure. This indicates that most significant user navigation through a website is tree-structured and major deviations from this structure are often noise. Therefore, forcing the trails to conform to the tree structure can improve the ranking power of our objective function.

We also show the usefulness of the tree-based quicklink selection algorithm via several anecdotal examples. Consider the website for the `Electronic Arts` game publisher `ea.com`. Quicklinks chosen from the tree-based approach are high-level web pages for seven popular games, such as `thesims2.ea.com`, `battlefield.ea.com`, `fifa08.ea.com` etc.

On the other hand the greedy approach picks as quicklinks six web pages from deep within the sub-site of the most popular game, namely thesim2.ea.com. The remaining two links are to high-level pages. Clearly, the enforcement of the homogeneity constraint in our tree-based approach helps find a cogent set of quicklinks in such cases.

Similar examples serve to demonstrate the usefulness of the parent-child constraint. As an instance, for the senate.gov website the following links are picked as quicklinks by the greedy approach: obama.senate.gov/, obama.senate.gov/about,obama.senate.gov/contact, and obama.senate.gov/votes. Clearly, the first of these links is the parent of all others making the rest redundant as quicklinks. The tree based algorithm on the other hand picks several different high level pages such as, obama.senate.gov/, biden.senate.gov/, kennedy.senate.gov/, mccain.senate.gov/ etc. Clearly the latter set of quicklinks are superior.

6. CONCLUSIONS AND FUTURE WORK

As search engines aim to enhance user satisfaction, they are offering increasingly sophisticated navigational aids that attempt to infer the users' intent and quickly take them to content that they wanted to reach but did not explicitly specify in their search queries. Quicklinks is an important example of this phenomenon; they let users directly access webpages within a website whose homepage was returned as a search result. As they occupy valuable real estate on the search results page, they must be picked so as to deliver significant benefits to a large fraction of users. In this paper, we framed this critical quicklink selection problem in terms of an objective function that assesses the navigational benefit of a candidate quicklink in terms of the observed browsing behaviors of users, as obtained from browser toolbar logs.

In addition to proposing the mathematically precise problem formulation, we formally proved the hardness of optimizing the objective and gave an algorithm that is provably within a factor of $(1 - 1/e)$ of the optimal. Empirical results using both manually labeled data and clickthrough data obtained from a real-world search engine demonstrated the efficacy of our approach, which beats three strong competing baseline algorithms by a significant margin. Our method combines trails obtained from toolbar logs with search logs and attains results that are better than those achieved by either individually. In fact, we obtain 22% improvement in terms of picking good quicklinks, and a 100% improvement in terms of not picking bad quicklinks, when compared against the best baseline algorithm.

There is significant scope for future work on this problem, primarily with respect to the tree-based algorithms. While our proposed algorithm picks quicklinks that obey certain desired constraints, it is unclear what the best set of constraints are, and how to best evaluate the results. Instead of enforcing hard constraints on the set of selected quicklinks, we are planning to explore an objective function that combines the current objective with a soft penalty for constraint violation. Another area of future work concerns standardising quicklinks *across sites*: e.g., all restaurant websites should have quicklinks for the menu, the restaurant location, and such. Finally, we want to explore extensions of the objective function to handle spiking interest on certain webpages/topics, and other such temporal effects.

Acknowledgments

We thank Youssef Billawala, Will Chin, Arah Cho, and Yong Gao for providing us data.

7. REFERENCES

[1] M. Bilenko and R. W. White. Mining the search trails of surfing crowds: Identifying relevant websites from user activity. In *WWW*, pages 51–60, 2008.

[2] M. Bilenko, R. W. White, M. Richardson, and G. C. Murray. Talking the talk vs. walking the walk: Salience of information needs in querying vs. browsing. In *SIGIR*, pages 705–706, 2008.

[3] P. Bose, E. Kranakis, D. Krizanc, M. V. Martin, J. Czyzowicz, A. Pelc, and L. Gasieniec. Strategies for hotlink assignments. In *Algorithms and Computation*, pages 23–34, 2000.

[4] A. G. Büchner, M. Baumgarten, S. S. Anand, M. D. Mulvenna, and J. G. Highes. User-driven navigation pattern discovery from internet data. In *WebKDD Workshop*, pages 74–91, 1999.

[5] I. V. Cadez, D. Heckerman, C. Meek, P. Smyth, and S. White. Model-based clustering and visualization of navigation patterns on a web site. *Data Mining and Knowledge Discovery*, 7(4):399–424, 2003.

[6] D. Chakrabarti, R. Kumar, and K. Punera. Generating succinct titles for web urls. In *KDD*, pages 79–87, 2008.

[7] J. Czyzowicz, E. Kranakis, D. Krizanc, A. Pelc, and M. M. Vargas. Enhancing hyperlink structure for improving web performance. *J. Web Engineering*, 1(2):93–127, 2003.

[8] C. Doerr, D. von Dincklage, and A. Diwan. Simplifying web traversals by recognizing behavior patterns. In *Hypertext and Hypermedia*, pages 105–114, 2007.

[9] Y. Fu, K. Sandu, and M.-Y. Shih. Clustering of web users based on access patterns. In *WebKDD Workshop*, pages 21–38, 1999.

[10] B. Hay, G. Wets, and K. Vanhoof. Mining navigation patterns using a sequence alignment method. *Knowl. Inf. Syst.*, 6(2):150–163, 2004.

[11] E. Kranakis, D. Krizanc, and M. V. Martin. Optimizing web server's data transfer with hotlinks. In *IADIS Intl. Conf. on WWW/Internet*, pages 341–346, 2003.

[12] E. Kranakis, D. Krizanc, and S. M. Shende. Approximate hotlink assignment. *Information Processing Letters*, 90(3):121–128, 2004.

[13] Y. Liu, B. Gao, T.-Y. Liu, Y. Zhang, Z. Ma, S. He, and H. Li. Browserank: Letting web users vote for page importance. In *SIGIR*, pages 451–458, 2008.

[14] P. Mayr. Website entries from a web log file perspective – a new log file measure. In *Proceedings of the AoIR-ASIST Workshop on Web Science Research Methods*, 2004.

[15] G. L. Nemhauser and L. A. Wolsey. Best algorithms for approximating the maximum of a submodular set function. *Mathematics of Operations Research*, 3(3):177–188, 1978.

[16] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. An analysis of approximations for maximizing submodular set functions. *Mathematical Programming*, 14:265–294, 1978.

[17] M. Perkowitz and O. Etzioni. Towards adaptive web sites: Conceptual framework and case study. *WWW8/Computer Networks*, 31(11-16):1245–1258, 1999.

[18] M. Perkowitz and O. Etzioni. Adaptive web sites. *CACM*, 43(8):152–158, 2000.

[19] S. Sakai, M. Togasaki, and K. Yamazaki. A note on greedy algorithms for the maximum weighted independent set problem. *Discrete Applied Math.*, 126(2-3):313–322, 2003.

[20] R. Srikant and Y. Yang. Mining web logs to improve website organization. In *WWW*, pages 430–437, 2001.

[21] V. Vazirani. *Approximation Algorithms*. Cambridge University Press, 2001.

[22] R. W. White, M. Bilenko, and S. Cucerzan. Studying the use of popular destinations to enhance web search interaction. In *SIGIR*, pages 159–166, 2007.