

The Last Click: Why Users Give up Information Network Navigation

Aju Thalappillil Scaria*
Google, Inc.
Mountain View, California
aju@google.com

Robert West
Stanford University
Stanford, California
west@cs.stanford.edu

Rose Marie Philip*
Facebook, Inc.
Menlo Park, California
rosephilip@fb.com

Jure Leskovec
Stanford University
Stanford, California
jure@cs.stanford.edu

ABSTRACT

An important part of finding information online involves clicking from page to page until an information need is fully satisfied. This is a complex task that can easily be frustrating and force users to give up prematurely. An empirical analysis of what makes users abandon click-based navigation tasks is hard, since most passively collected browsing logs do not specify the exact target page that a user was trying to reach. We propose to overcome this problem by using data collected via *Wikispeedia*, a Wikipedia-based human-computation game, in which users are asked to navigate from a start page to an explicitly given target page (both Wikipedia articles) by only tracing hyperlinks between Wikipedia articles. Our contributions are two-fold. First, by analyzing the differences between successful and abandoned navigation paths, we aim to understand what types of behavior are indicative of users giving up their navigation task. We also investigate how users make use of back clicks during their navigation. We find that users prefer backtracking to high-degree nodes that serve as landmarks and hubs for exploring the network of pages. Second, based on our analysis, we build statistical models for predicting whether a user will finish or abandon a navigation task, and if the next action will be a back click. Being able to predict these events is important as it can potentially help us design more human-friendly browsing interfaces and retain users who would otherwise have given up navigating a website.

Categories and Subject Descriptors

H.5.4 [Information Interfaces and Presentation]: Hypertext/Hypermedia—*navigation*; H.1.2 [Models and Principles]: User/Machine Systems—*human information processing*

General Terms

Experimentation, Human Factors

*Research done as graduate students at Stanford University.

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. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
WSDM '14, February 24–28, 2014, New York, New York, USA.
Copyright 2014 ACM 978-1-4503-2351-2/14/02...\$15.00.
<http://dx.doi.org/10.1145/2556195.2556232>.

Keywords

Navigation, browsing, information networks, abandonment, Wikipedia, Wikispeedia

1. INTRODUCTION

Finding information on the Web is a complex task, and even expert users often have a hard time finding precisely what they are looking for [19]. Frequently, information needs cannot be fully satisfied simply by issuing a keyword query to a search engine. Rather, the process of locating information on the Web is typically incremental, requiring people to follow hyperlinks by clicking from page to page, backtracking, changing plans, and so forth. Importantly, click-based navigation is not simply a fall-back option to which users resort after query-based search fails. Indeed, it has been shown that users frequently prefer navigating as opposed to search-engine querying even when the precise search target is known [12], one of the reasons being that useful additional information can be gathered along the way [18].

The complexity of such incremental navigation tasks can be frustrating to users and often leads them to give up before their information need is met [5]. Understanding when this happens is important for designing more user-friendly information structures and better tools for making users more successful at finding information. However, an empirical analysis of what makes users give up click-based navigation tasks is complicated by the fact that most browsing logs, as collected by many search-engine providers and website owners, contain no ground-truth information about the exact target pages users were trying to reach. As a result, in most previous analyses, targets had to be inferred from the queries and from post-query click behavior. For these reasons researchers have typically made heuristic assumptions, e.g., that the information need is captured by the last search result clicked, or the last URL visited, in a session [6]. Other researchers have avoided such assumptions by classifying information needs in broader categories, such as informational versus navigational [20], while ignoring the exact information need.

Furthermore, since passively collected browsing data contain no ground-truth information about the target the user is trying to reach, it is also very hard to determine why the user stopped browsing at any particular point. On one extreme, she could have found the information she was looking for and abandoned the session simply because the information need was satisfied. Or, on the other extreme, it could be that the user abandoned the search process precisely because she could *not* locate the desired information and

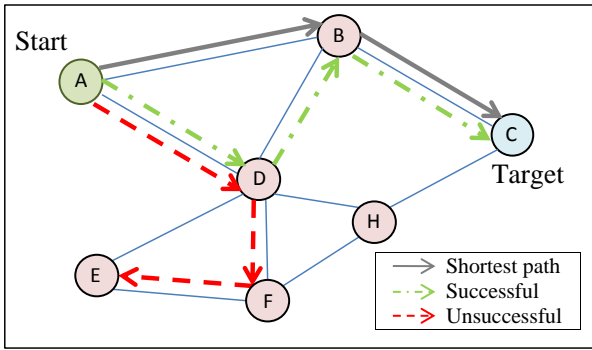


Figure 1: An example navigation mission from start A to target C. The gray arrows represent an optimal 2-click navigation, the green arrows illustrate a successful 3-click navigation, and the red arrows show an unsuccessful, abandoned 3-click navigation, where the user did not reach the target C.

gave up. Thus, abandonment in Web search and navigation is not always a sign of giving up [9].

In this paper, we address these two issues by leveraging data collected from *Wikipedia* [17], a Wikipedia-based online human-computation game. In this game, users are given explicit navigation tasks, which we also refer to as *missions* (Fig. 1). Each mission is defined by a *start* and a *target* article, selected from the set of Wikipedia articles, and users are asked to navigate from the start article to the known target in as few clicks as possible, using exclusively hyperlinks encountered in the Wikipedia articles along the way.

Using this dataset alleviates both of the problems we have discussed earlier: First, the goal of the navigation mission (i.e., the target page) is explicitly known and need not be inferred. Second, since the task can by definition only be solved by reaching the target, abandonment can always be interpreted as giving up (although we acknowledge that, next to failure, boredom could also be a reason for giving up; we address this point later, when introducing the dataset). An additional advantage of our setup is that each page is a Wikipedia article and therefore has a well-defined content, comparable to other pages by standard text-mining measures such as TF-IDF cosine.

We believe that these factors enable a more controlled analysis of the reasons causing users to give up click-based navigation tasks than most previous log-based studies. In particular, our work makes two main contributions:

1. We empirically analyze differences and commonalities between finished and abandoned navigation tasks. We also analyze back clicks, a frequently employed means to back off from unsuccessful strategies.
2. We apply our findings by building statistical models for predicting whether users are about to give up on a navigation task, and whether they will hit the back button next.

Regarding the first point, our analysis shows that some missions are intrinsically harder than others, but also that, even when controlling for intrinsic hardness, some users manage to find better strategies and are thus also less likely to give up. We observe that it is typically easy to make progress at the beginning of navigation paths, whereas there is much more variability later on, when users are trying to home in on the target page. Sometimes users make bad moves, taking them further away (in a shortest-path sense) from the

target than they already were, and in these cases, we note an increased likelihood of back clicks—a sign that users typically have a sense for whether they have made progress towards the target article or not. However, users who are to eventually give up are worse at doing so. They click back despite having made progress towards the target much more frequently than successful users do. In other words, not only are unsuccessful users worse at making progress, they are also worse at realizing it once they do make progress.

Concerning the second contribution, our predictive models could potentially be the basis for mechanisms serving to improve websites in an interactive manner. For instance, if we can reliably predict whether a user is about to give up navigating, i.e., to leave the site, we could take actions to retain that user, e.g., by displaying a support phone number or a chat window. The features we use in our models are based on the insights drawn from the analysis of successful and unsuccessful missions, and predict well if users are going to quit their navigation.

2. RELATED WORK

Related work may be divided into three broad areas: query abandonment, click-trail analysis, and decentralized search.

Query abandonment, as studied in the information retrieval community, analyzes queries that are not followed by a search-result click. Li et al. [9] distinguish ‘bad’ and ‘good’ abandonment. The former occurs when the information need could not be met (e.g., because no relevant results were returned), the latter, when the information need could be satisfied even without clicking a search result. Good abandonment occurs surprisingly often (17% of abandoned queries on PCs, 38% on mobile devices), which confirms the need for controlling for abandonment type, as we attempt to do. Diriyee et al. [5] strive to understand more deeply and on a larger scale what causes users to abandon queries, thus eliciting useful indicators which they subsequently use to build a predictive model of query abandonment. Das Sarma et al. [4] focus on bad abandonment and offer a method to reduce it by learning to re-rank results. In these works, abandonment is usually defined as ‘zero clicks’. We, on the contrary, adopt a broader definition, by considering as abandoned also those searches in which the user has already made some clicks but gives up before reaching the final target. Note that this is possible only because the exact target is explicitly known in our setup. Also, our setup involves only click-based navigation and no search queries, which allows us to analyze human navigation behavior in isolation, thus avoiding conflating users’ navigation skills with their query formulation skills.

Click-trail analysis primarily studies the click paths on which users embark, starting from search-engine result pages. In early fundamental work, Chi et al. [3] invent the notion of ‘information scent’, operationalized by Olston and Chi [11] in a system for supporting users by combining query- and click-based navigation strategies. We have already mentioned the work of White and Huang [18], who establish that click trails add value to the information contained in the ultimate target page, and of Teevan et al. [12], who show that users frequently prefer click-based navigation to querying. Downey et al. [6] investigate the benefits of navigating versus querying further, finding that navigating is particularly useful when the information need is rare. Finally, White and Singla’s [20] work is relevant in that it explores how different trail topologies (such as stars, trees, and linear chains) are observed in different search scenarios (informational versus navigational). Our work extends these ideas. For instance, while White and Singla [20] manually assign topologies to paths, we devise an automatic method for doing so. Further, while they must heuristically assume that the

last observed page is the target page, we have ground-truth target information, which allows for a much refined analysis.

Some very relevant work was done by Ageev et al. [1]. They, too, have developed a human-computation game for collecting data, in which users are asked to find the answers to as many factual questions (e.g., ‘What is the highest peak in the Western Hemisphere?’) as possible within a given amount of time, using Web search queries that may optionally be followed by click-based navigation. There, too, the goal is explicitly known, but not in the form of a specific target page but rather in the form of a specific answer string. Ageev et al.’s setup has the advantages of a more realistic search scenario, since both querying and navigating are allowed, and of a more realistic browsing environment (the Web rather than Wikipedia only). On the flip side, our approach allows for several types of analysis that are impossible in their setup. For instance, each Wikipedia page has a clearly defined topic, with the target being exactly one page (rather than an answer string that might be gleaned from several pages), such that it is straightforward to quantify how related a page along the click path is to the target. Moreover, the network topology is known entirely in our case.

Decentralized search is rooted in Milgram’s seminal small-world experiment [10], which asked participants to forward a letter to a friend such that it could finally reach a predestined target person. The game through which our data was collected is similar to this task in that a target must be reached in the absence of prior information about the underlying network. It is different in that our setup has the same user staying in control from the start all the way to the target, whereas, in the small-world experiment, every step is executed by a new, autonomous participant. Kleinberg [8] investigated the algorithmic aspects of the small-world problem, showing that efficient search is only possible when the probability of long-range link decays with a specific exponent. Much research has followed in Kleinberg’s wake, so we focus on the most directly related pieces of work: datasets such as ours were previously analyzed by West and Leskovec, who characterize human strategies in successful navigation tasks [16] and train machine-learning models capable of navigating automatically [15], and by Trattner et al. [13], who explore heuristic navigation algorithms based on hierarchical knowledge representations. The present paper extends previous work by considering abandoned navigation paths in addition to successful ones.

3. DATASET

In this section, we introduce the data collection procedure and the dataset we used. We also make the data available online.¹

Data collection. We analyze navigation paths collected through the Wikipedia-based human-computation game *Wikispeedia*² [17] (see Fig. 2 for screenshots). The rules are simple: A user arriving at the site is given a mission (Fig. 1), consisting of a start and a target article sampled randomly from the set of all articles in the strongly connected core component of Wikipedia’s hyperlink graph, to ensure that there is a path between the two. Alternatively, the user can specify a mission herself, or choose from a list of 5 randomly sampled missions solved previously by other users. Given a mission, the task is to navigate from the start to the target article, exclusively by following hyperlinks, in as few clicks as possible. Backtracking via the browser’s back button is allowed, and forward clicks that were later undone this way, as well as the back clicks themselves, are not counted when ranking users in high-score tables. Many Wikipedia articles are about obscure topics few people know about,

but having a clear idea of the target and its context is crucial for solving navigation tasks. Therefore, users have the option to read the contents of the target article before starting to navigate.

A condensed and curated version of Wikipedia³ is used, containing 4,600 articles and around 120,000 links. We consider a set of around 32,000 navigation paths. The data has been collected since August 2009, when the Wikispeedia website was launched. Participation has been completely voluntary and open to all Web users, with no reward offered, other than the fun inherent in the game.

Of the 32,000 recorded paths, only 49% were completed successfully. Indeed, it is this surprisingly low number that triggered our investigation into what causes users to give up. It is, however, important to note that, while Wikispeedia produces data that is well suited for studying user abandonment, the site was not specifically designed for this purpose.

In addition to the navigation paths, the site also logs if a task ended (1) because the session expired (after 30 minutes of inactivity; 38% of abandoned missions), or (2) because the user decided to give up on the current task and start a new one (62%). Upon restarting, the user is presented with a different start–target pair.

Data preprocessing. Several factors might play a role in giving up: (1) Users might be unable to solve the navigation task, or (2) they might have become bored, disinterested, or distracted, switching their attention to something else. We are interested in giving up due to failure (reason 1) rather than disinterest or distraction. As the Wikispeedia interface does not ask abandoning users why they gave up, we cannot know which missions were given up because of reason 1. Instead, we filter our data the following way. First, we think it is reasonable to assume that the tasks abandoned after a very small number of clicks (in the extreme case without a single click) were given up not because the users were unable to solve the task, but rather because they got bored, disinterested, or distracted. Hence, we consider only missions in which at least 3 clicks were made. Second, we argue that timeouts tend to be caused by boredom, disinterest, or distraction, whereas quitting the current mission and starting a new one is much more likely to be a sign of failure to find a path to the target article. Therefore, we restrict the set of abandoned missions to those that the user gave up in order to start a new one.

Not all start–target pairs can be solved in the same optimal number of steps, so another potential confounding factor becomes immediately clear: missions with longer optimal solutions are intrinsically harder, which is likely to increase the probability of giving up, independent of any other factors. To control for this effect, we further constrain our analysis to missions with an optimal solution length of exactly 3 clicks. This does not decrease our sample size by too much, due to the fact that the Wikipedia hyperlink graph constitutes a tightly connected ‘small world’, with a mean/median/maximum shortest-path length of 3.18/3/9, across all node pairs. After these filtering steps, the final dataset we study consists of about 16,000 paths (10,000 finished, 6,000 unfinished).

4. DYNAMICS OF GIVING UP

Previous work showed that, at every single step in our setting, there is roughly a 10% chance of the user dropping out [16]. In this section, we investigate the reasons *why* users give up while navigating to find some specific piece of information. We break our analysis into two parts. First, we investigate the intrinsic hardness of navigation tasks, as hard missions are more likely to be aban-

¹<http://snap.stanford.edu/data/wikispeedia.html>

²<http://www.wikispeedia.net>

³<http://schools-wikipedia.org> (accessed 08/2008)

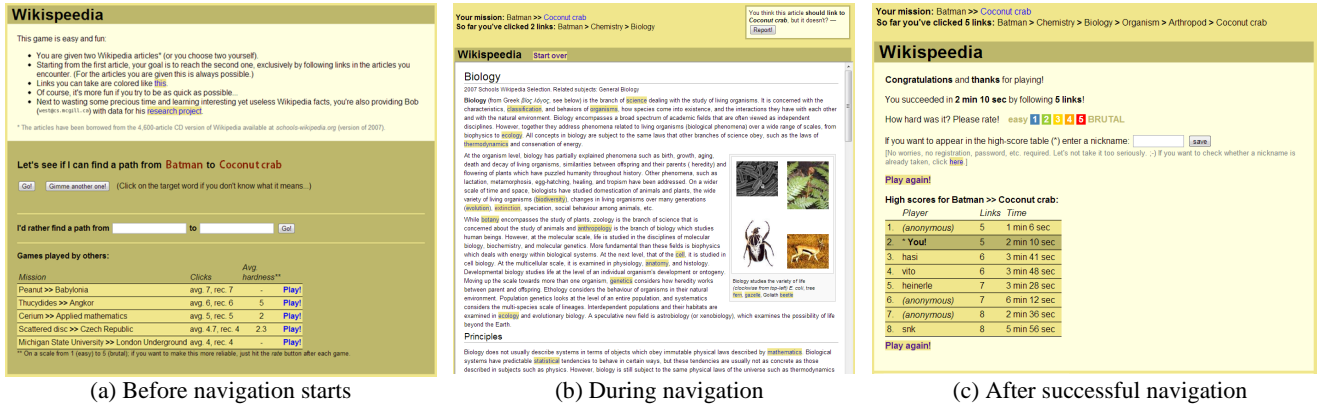


Figure 2: Screenshots of Wikipedia, the human-computation game used for data collection (cf. Section 3).

Property	Finished	Unfinished
TF-IDF distance	0.971	0.977
Target in-degree	62.8	28.6
Target PageRank	4.77×10^{-4}	2.33×10^{-4}
Source out-degree	32.4	31.1

Table 1: Means of start and target properties for finished versus unfinished navigation paths. Two-tailed t -tests yielded highly significant differences ($p < .0005$) for all pairs.

done. Second, we analyze the user behaviors that are indicative of successful or unsuccessful⁴ navigation, respectively.

While we think the way we preprocessed our data eliminated the most obvious confounding factors, the question of intrinsic mission hardness remains: even if all missions can be optimally solved in 3 clicks, solutions may be easier to find for some missions than others. Hence, we structure the present analysis into two parts: (1) We analyze what features of the start and target pages (beyond mere shortest-path length) make a mission intrinsically hard, and (2) we look into how successfully finished navigation paths differ from prematurely abandoned ones, in terms of features of the click paths rather than features of just the start and target articles.

Intrinsic mission hardness. To elicit the factors that make a mission hard independent of any user-dependent features, we compare properties of the start and target articles for finished versus unfinished missions. Note that we only consider start–target pairs that can be reached in exactly 3 clicks. The results are summarized in Table 1. We make the following observations:

- The textual distance between the source and target articles (measured as one minus the TF-IDF cosine of the article contents) in successful and unsuccessful missions are significantly different (the numbers are so close because all TF-IDF distances tend to be close to 1). This means that it is less frustrating to find paths between start and target articles that are conceptually more related.
- The in-degree and PageRank of target articles in unsuccessful missions are only about half as large as in successful ones. Both of these measures capture how easy it is to reach the target article (in-degree in a local, and PageRank in a global sense), so we conclude that missions with hard-to-find targets are more likely to be abandoned. While at first glance this might seem trivial, it is important to note that PageRank is

based on a random-walker model [2], whereas humans do not navigate randomly; they have vast amounts of background knowledge, which might well allow them to find paths that are hard to find for an uninformed random walker. So the fact that the simplistic notion of PageRank is a good indicator of whether a mission will be abandoned is in fact nontrivial and of practical importance (e.g., later on in this paper, when we use it as a feature for predicting abandonment).

- The out-degree of start nodes is similar in finished and unfinished paths. This means that the starting phase of missions is less important than the final phase. Even if the start article offers only few outgoing links, the chance of giving up are not much affected. Thus, properties of the source are not indicative of user abandonment. It is only the properties of the target that determine the success rate of a mission. We hypothesize this is caused by the small-world structure of the network: there are some hub nodes (such as UNITED STATES) that are easily reached from all over the network (even from low-out-degree nodes) and that link to many distinct parts of it, so they often serve as transition points between an easy ‘getting-away’ and a harder ‘homing-in’ phase [16].

In summary, even when the optimal path length is held fixed, the intrinsic hardness of missions varies widely as a function of both content- and network-related properties of the start and target articles. To gain additional insights, we will later further control for mission hardness by only considering all the instances of particular start–target pairs (e.g., BRAIN to TELEPHONE).

Navigation in successful versus unsuccessful paths. Another way to understand why people give up is to analyze properties of articles along the path they took. We compare how different article properties evolve along navigation paths both for successful and unsuccessful missions. The results to be discussed next are depicted in Fig. 3. Plots in the left column were computed based on finished missions, and those on the right based on unfinished ones. Even though all missions had a shortest-path length (SPL) from start to target of 3 clicks (cf. Section 3), the actual paths can consist of any number of clicks, depending on the choices made by the user. As a result, we draw a separate curve for each path length, grouping all paths of the same length together. The x -axes show the path position (in terms of the number of clicks made from the start article), and the y -axes, the mean of the respective property. Thus, each curve may be thought of as representing the average value of the property—running from left to right—for the respective path length. Since the last node is always the target in finished paths,

⁴We use the word groups *finished/successful* and *unfinished/unsuccessful/abandoned/given up* interchangeably.

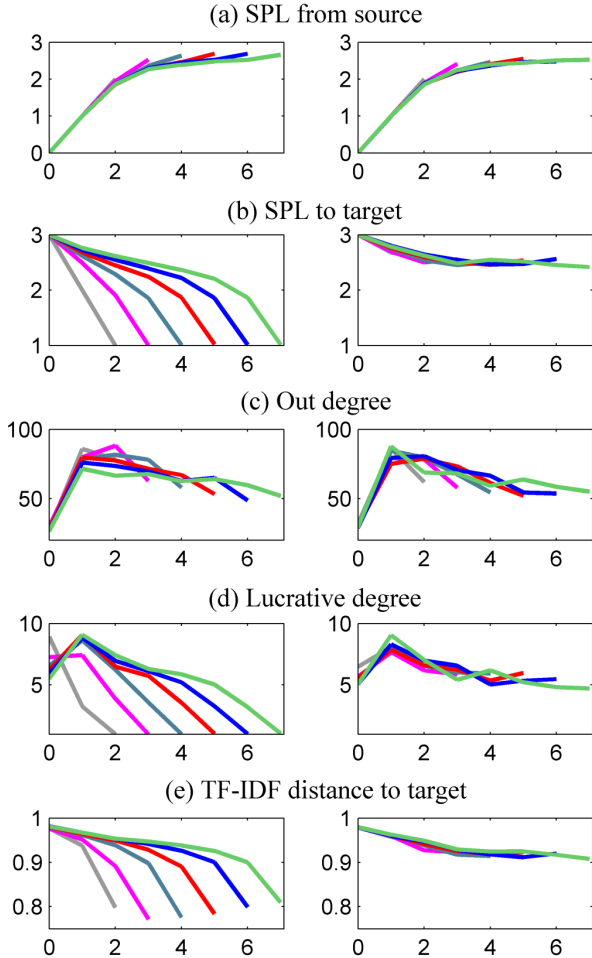


Figure 3: Comparison between finished (left) and unfinished (right) navigation missions. Paths of the same length are grouped in the same curve and have the same color. The x-axes show the path position (defined as the number of clicks from source), and the y-axes, the mean of the respective property. The last node along the path (the target in the case of finished paths) is omitted from all plots. See text for description of properties.

this would in many cases distort the y-axes, so we do not plot it. To be consistent, we also omit the last node from all unfinished paths.

One of our early hypotheses for explaining why people abandon navigating was that they might have traveled very far from both the start and the target node—in other words that they might have gotten ‘lost’ in the network. However, as the top two rows of Fig. 3 show, the average SPL (shortest-path length) with respect to both the start and the target article always remains less than 3, even for unfinished paths. We further see that the average SPL from the source (row 1) looks very much the same when compared to successful navigation tasks. Successful and unsuccessful missions also have in common that the SPL to the target (row 2) decreases with the first click. However, while in successful missions, users make on average continuous progress towards the target, this is not the case in abandoned missions, where it stays at a constant level of around 2.5. The same pattern of initial progress with ensuing stagnation is exposed by the plot of TF-IDF distance to the target (row 5).

So, while straying very far from either source or target is not a big danger—since everything tends to be close to everything else in small-world networks—unsuccessful users seem to be unable to make the crucial click that finally lets them ‘gravitate’ towards the target. Therefore, instead of ‘getting lost’, the analogy for unsuccessful navigation should rather be ‘orbiting’ at a constant distance from the target, which can be frustrating, thus causing users to give up.

This reinforces a point we have already briefly touched upon in our discussion of intrinsic hardness: making progress is easy in the beginning. SPL and TF-IDF distance to the target drop initially even for unfinished missions, but the crucial difference to finished missions is that users in finished missions continue to make progress, while those in unfinished missions do not.

As mentioned before, West and Leskovec [16] have argued that the initial progress might stem from clicking to a high-degree hub node. This is confirmed by Fig. 3, rows 3 and 4, which show that out-degree and lucrative degree (defined as the number of outgoing links that decrease the SPL to the target) spike after one click. We see that a hub-based strategy is pursued even by users who are to eventually give up, another indicator that the opening is not as crucial as the later phase in determining whether a navigation task will be finished.

The case of paths with more than 8 clicks. Note that Fig. 3 is based on missions of a maximum of 8 clicks. It is also interesting to look at longer missions, the question being: If a successful mission consists of many clicks, is it that the user made steady progress throughout the mission and just took longer to find the target, or did they rather stumble upon the target by luck after trying a lot of different options?

To answer this question, Fig. 4 repeats some of the plots of Fig. 3 for longer missions (10, 15, and 20 clicks), comparing them with missions of length 5. We find that the distinction between successful and abandoned missions is much diminished for paths of many clicks, except that there is a brief final phase of ‘gravitating into the target’ in the case of successful missions. It looks more as though in these cases the successful users found a way to the target not so much via a straightforward strategy but rather by luck, having ‘orbited’ around the target for a long time, as typical also for abandoned missions. The other properties show similar trends and we omit them for space reasons.

Controlling for intrinsic hardness. Since certain features of start and target articles can make missions intrinsically harder, a critical reader might object that mixing missions of all levels of intrinsic hardness together (as we do in Fig. 3 and 4) might induce a potential confounding factor: as harder missions are *a priori* more likely to be abandoned, the differences observed in our previous analysis might indeed be caused by the skewed levels of hardness in finished versus unfinished missions, rather than by more versus less successful strategies. To rule this out, we repeated our analysis for 4 *fixed* pairs of start and target articles for which an increased number of samples were collected, thus controlling for intrinsic hardness. The resulting figures are qualitatively very similar to those shown, and all our findings still hold. We nonetheless show the figures based on all missions, as they are based on more samples and therefore more reliable.

5. BACK-CLICKING PATTERNS

Given that navigating information networks is a complex task, it is quite natural for users to backtrack while navigating to locate a specific target. *A priori*, it is not clear if backtracking is rather a sign of successful or of unsuccessful missions: On one hand, a mission might be successful because the user extensively explored



Figure 4: Analogous to Fig. 3 (b and e), but for a wider range of path lengths (left: finished; right: unfinished).

	Finished			Unfinished		
	–	=	+	–	=	+
SPL	.242	.102	.033	.233	.153	.116
TF-IDF dist.	.189	.096	.049	.253	.161	.111

Table 2: Back-click probability conditional on progress: (+) progress, i.e., property decreased with last click; (–) regress, i.e., property increased; (=) neither progress nor regress (in the case of TF-IDF, taken as a relative change of < 10%).

the network via back clicks, but on the other hand, backtracking might be a sign of the users being at a loss.

The data favors the second alternative, with backtracking being more common in unsuccessful missions: 20.3% of abandoned paths contained at least one back click (average of 2.48 back clicks per mission), while this percentage is only 16.4% (2.24 back clicks per mission) for finished paths.

Next, to dig deeper, we consider the following two questions: First, when do users typically backtrack? Second, what are the most common backtracking patterns?

When do users typically backtrack? We conjectured that users are inclined to backtrack when they feel they might have navigated too far away from the target article. To test this hypothesis, we observe how the probability of backtracking varies depending on whether progress towards the target was made with the previous click. Here we measure progress in two ways: (1) Did the SPL to the target decrease? (2) Did the conceptual distance to the target (measured as one minus TF-IDF cosine similarity of article texts) decrease? The findings are summarized in Table 2. Note that backtracking is least likely when progress was made with the previous click, most likely when the opposite of progress was achieved (i.e., when SPL or TF-IDF distance to the target increased), and on a middle ground when neither of these two extremes is the case. Furthermore, the probability of backtracking after making progress is considerably higher in unsuccessful than in successful paths (11.6% vs. 3.3% for SPL, and 11.1% vs. 4.9% for TF-IDF distance). We draw two conclusions from these observations. First, even though the underlying network structure is unknown, both successful and unsuccessful users have an intuition whether a click choice they have just made was good or bad. Second, successful users are better at forming these intuitions, whereas unsuccessful users are three times more likely to undo a useful click, which in

fact took them closer to the target. This is similar to a finding by Ageev et al. [1], who show that successful users are better at identifying relevant documents on search engine result pages.

What are the most common backtracking patterns? Given that backtracking is important in distinguishing finished from abandoned navigation paths, we wanted to go beyond simply counting paths with at least one back click and move to a more topological analysis of how back-click patterns are arranged in paths. We started by adopting a methodology similar to White and Singla’s [20] and manually inspected graphical representations of navigation paths (such as in Fig. 5), finding that one main dimension of variability is the ‘star-likeness’ of paths. A path from start to target is more star-like when all back clicks lead back to the same node, which serves as a hub for exploring its neighborhood (cf. Fig. 5(d) for an extreme example).

Whereas White and Singla stick to hand-labeling, we strive to automatically analyze the tens of thousands of paths we have at our disposal. To this end, we introduce a novel but simple measure, which we call *star score*. Intuitively, a path is maximally star-like if all back clicks return to the same hub, and least so if every back click leads to a different previously visited node. More formally, the star score should be high if the distribution of back-click targets is highly peaked. Additionally, when this distribution is equal for two paths, we want a path with a larger number of back clicks to receive a higher star score. Formally, the *star score* of a path p is defined as

$$S(p) = (1 - G(p)) \cdot N, \quad (1)$$

where N is the total number of back clicks on the path, and $G(p) = 1 - \sum_{i=1}^n f_i^2$ is the Gini impurity of the back-click distribution. The Gini index measures how even or uniform a given distribution is (f_i is the fraction of back clicks to node i and n is the number of nodes in the path).⁵ The star score is defined for paths with at least one back click and always lies between 1 and N .

Fig. 5 shows the plot of the paths taken by some users as directed graphs, alongside the corresponding star scores. Clearly, the graphs with denser star-like patterns have higher star scores.

Initially, we hypothesized that unsuccessful missions would be more likely to show star-like patterns because users keep exploring new ways and then backtrack from them without achieving success. However, we were unable to discern any noticeable differences in the distribution of star score in successful versus unsuccessful paths. We think that the reason is that exploring in a star-like fashion is not necessarily a sign of failure—indeed, as argued in the beginning of this section, it might even contribute to a better understanding of the local graph neighborhood and thus increase the chances of success. Still, navigation paths with very high star scores (greater than 6), even though few in number, all ended up being abandoned.

Moreover, it seems that not all nodes are equally suited for backtracking. Backtracking may be the manifestation of a change of strategy, so we hypothesized that users are more likely to return to hub nodes, i.e., nodes of high out-degree, which are best suited for ‘teleporting’ to a different part of the network. To investigate this issue, we found, for each path involving a back click, the node receiving the highest number of back clicks and computed its rank in the list of all nodes on the path sorted by out-degree. The results are presented in Fig. 6. In 30% of unfinished, and around 20% of finished missions, the node receiving most back clicks is also the node of highest out-degree on the path. Also, the likelihood of a node re-

⁵A user might visit the same article more than once even without a back click, through a series of clicks, which we consider equivalent to a back click.

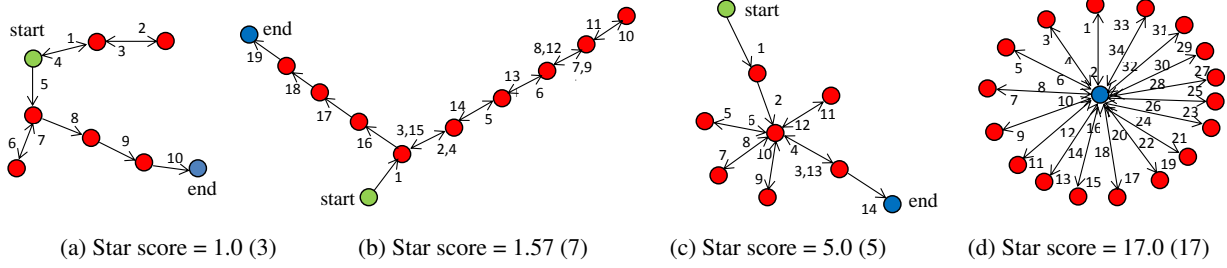
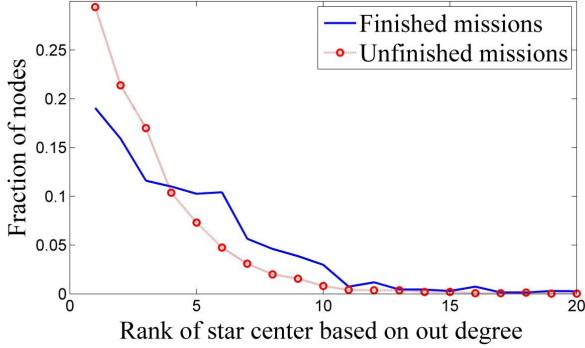


Figure 5: Examples of navigation paths, illustrating our star score metric. The numbers within brackets are the numbers of back clicks on the paths. Users navigate from the start node to the end node (i.e., the target for finished, and the last visited node for unfinished, paths). The numbers along the edges indicates the ordering of clicks. In (d), the start and end nodes are the same.



ceiving most back clicks shrinks with decreasing out-degree rank. It is also interesting to note that users in unfinished missions have a higher inclination to backtrack to hub nodes.

We have focused on star-like patterns here, but for future work, it might be interesting to explore a wider variety of topological navigational structures, using more advanced graph techniques from the data visualization community (e.g., [14, 7]).

6. PREDICTING ABANDONMENT AND BACK CLICKS

In the earlier sections, we have analyzed the reasons why some navigation paths end up abandoned, and also in what situations users decide to backtrack. Our next goal is to apply the lessons learnt, in order to develop models for predicting three navigation events, based only on the first few clicks:

1. Will the user abandon the mission before finishing it?
2. Will the user backtrack with the next click?
3. Will the user give up without making any additional clicks?

We formulate binary classification tasks and apply logistic regression with a set of features based on the observations made earlier. The model learns weights for each feature and makes a prediction of the likelihood of the event given a new example. Since the model outputs a probability value, the effectiveness of the system depends on the threshold (or confidence) above which we declare a sample to belong to the positive class. As the choice of threshold affects the accuracy of the task, we decided instead to plot ROC curves and use the area under the curve (AUC), which does not require choosing a threshold, as our performance measure. Since our models are trained on actual human navigation trajectories, they learn characteristic features of human behavior.

Applications of such models have already been alluded to in the introduction. They include tracking the progress of a website user

to understand if he is about to stop his navigation, or is lost in the network of pages, so that we could incentivize him to stay by offering phone or chat support, or by offering special bargain deals.

6.1 Description of Prediction Tasks

We first give details about each of the three prediction tasks. Then we describe the features used, and finally we present the results. For each of the tasks, we created a balanced dataset consisting of equal numbers of positive and negative examples (by randomly sampling from the class with more examples) and used 70% of the data for training and 30% for testing.

Task 1: Will the user abandon the mission before finishing it?

Here, we predict if a user will complete or abandon a mission with known start and target, based on only the first few clicks. The number of clicks given as input to the classifier, which we refer to as the *window size* W , is a parameter that can be chosen, and we build separate models for $W = 1, \dots, 4$. We restrict ourselves to missions with a SPL from start to target of at least 3. The negative examples consist of unfinished paths of at least W clicks. Regarding the positive examples, we took additional care: In Fig. 4, we saw that for very long paths, the differences between successful and unsuccessful instances vanish, so for each path, we included it only if it was finished within 5 additional clicks, compared to the optimum (i.e., compared to the SPL from start to target). For instance, if the SPL from start to target was 3, we allow up to 8 clicks; and for SPL of 4, up to 9, and so on. We define this value as the *tolerance* allowed in positive samples (in this case, the tolerance is 5). Further, Fig. 3 shows that finished paths differ a lot from unfinished ones in the last couple of clicks. We want to prevent the learning algorithm from ‘cheating’ (by simply learning these typical endgame patterns), so, for window size W , we consider only paths of at least $W + 2$ clicks.

Task 2: Will the user backtrack with the next click?

The dataset we used for this task is same as in the previous one. We again exclude the last two clicks from consideration, for the same reasons outlined above. In the previous task, each input path was represented by the first W clicks. However, since in the present task we are trying to predict a local event (Will the next click be a back click?), we extract features from a sliding window of W clicks. For example, for $W = 3$, in a path with 8 clicks, 3 samples may be constructed: one to predict click 4 (whether it will be a back click or not) from clicks 1, 2, and 3; another to predict click 5 from clicks 2, 3, and 4; and so on. Positive examples are those in which the last click in the window was followed by a back click, and the rest, negative examples.

Task 3: Will the user give up after the current click?

We extend the model used for predicting back clicks to also predict if the person is going to quit navigating immediately after the current click, without making any additional clicks. Here, too, we use a sliding window, similar to the model for predicting back clicks.

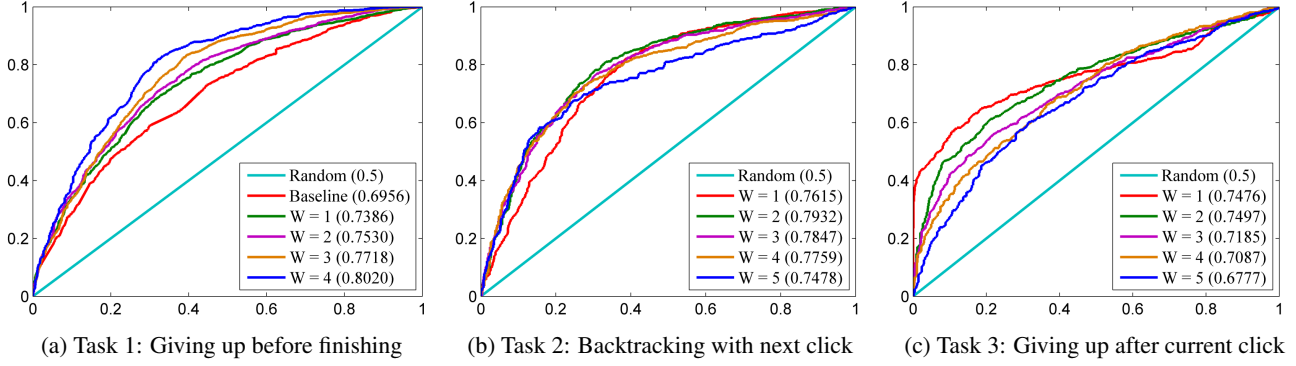


Figure 7: ROC curves for the 3 prediction tasks; AUC in parentheses; W is a window-size parameter (see text for description).

	Tolerance w.r.t. SPL		
	3	4	5
AUC	0.958	0.8304	0.802

Table 3: Variation in AUC of classifier for predicting abandonment with tolerance in path length of positive samples (for windows size $W = 4$). See text for description.

Positive examples are those in which the last node in the window is also the last node on the path, and the rest are negative examples.

6.2 Features Used for Learning

The features we use in all three tasks are based on the insights drawn by comparing the properties of finished and unfinished missions. In each of the prediction tasks, given W clicks (or $W + 1$ nodes), we use the following features:

- *Properties of start and target nodes:* SPL from start to target; lucrative degree (as defined in Section 4) and lucrative ratio (defined as the fraction of out-links of a node that decrease SPL to target) of start node; PageRank of target; in-degree of target; TF-IDF distance of start and target.
- *For each adjacent node pair:* TF-IDF distance; relative change in lucrative degree and lucrative ratio.
- *For each node:* lucrative degree; lucrative ratio; TF-IDF distance to target; SPL to target; indicator variable: Was a back click done although SPL to target decreased compared to previous node (inspired by Table 2)?

For some tasks, we use additional features, which will be introduced as we are discussing the results for the respective task.

6.3 Experimental Results

Next, we present the results of our predictive models on each of the three tasks.

Results for Task 1. For the task of predicting if a path is going to be finished successfully or not (Fig. 7(a)), we built a baseline model that uses only properties of start and target to make a prediction. This model yields an AUC of 0.70. Considering that random guessing would give an AUC of 0.5, this baseline model performs rather well, which supports our argument from the Section 4 that some navigation tasks are intrinsically harder than others. Fig. 7(a) further shows that, as expected, a larger window size W results in better predictions; specifically, for $W = 4$, the AUC increases to 0.80, a 14% relative improvement over the start–target baseline.

We also wanted to see how sensitive our results were to the choice of the tolerance of path length allowed in positive samples (set to 5, for the plots in Fig. 7(a)). So, we repeated the evaluation for tolerances of 3 and 4 (and $W = 4$), with the result that, the lesser the tolerance, the higher the AUC (cf. Table 3). This is expected: as we raise the quality of finished paths (by reducing the tolerance), the differences between finished and unfinished navigation paths becomes more obvious.

Results for Task 2. For this task, we used two additional features apart from the ones used for Task 1: the number of clicks made so far on the path, and the fraction of back clicks in the path till now. The number of pages visited is important because people could have a different rate of quitting depending on how far into the path they are, and we want to capture this. The performance for various choices of the window size W , visualized in Fig. 7(b), lie between 0.75 and 0.79. Performance goes up as we increase W from 1 to 2 and then goes down as we increase it further. We assume this is the case because backtracking is a local event that is done mainly when users feel they have made bad choices in the last one or two clicks. As a result, beyond $W = 2$, the larger context size increases the noise, while adding only little signal.

Results for Task 3. Here we use the same features as in Task 2. The results are plotted in Fig. 7(c). The performance of predicting whether the user is going to leave after the current click turns out to be harder than Task 1, in which we predicted whether the user will abandon *at any later point*. Even if someone has made up their mind to leave the mission, it could still be the case that they make one or two more clicks, simply out of curiosity or to see if they get closer to the target by luck. This kind of behavior is difficult to predict and could be a reason why our model performance is slightly lower than that on the other two tasks. Even so, we still achieve an AUC between 0.68 and 0.75, depending on W .

6.4 Analysis of Learned Predictive Models

Our models use around 30 features. In this section, we describe our method to find the features that are most important and analyze the results. We performed 10-fold cross validation on the training set, in combination with backward feature selection, starting with the set of all features and progressively discarding features that contributed least to the AUC score until the AUC score began to drop. The following list of features was found to be most powerful for Task 1, alongside explanations why these features might matter:

- *SPL from last node in input to target:* If a user has not made much progress towards the target in the first few clicks, he has more ground to cover and there is a higher chance that he might get frustrated and give up.

- *TF-IDF distance between last two nodes in input and target:* The same reasoning applies.
- *In-degree of target:* The fewer in-links the target has, the more difficult it is to find in the network, making the mission intrinsically harder (cf. Table 1).
- *Indicator whether a back click was made despite having made progress:* As we saw in Table 2, users who give up are worse at identifying that they have made progress.
- *TF-IDF distance between adjacent nodes:* Making abrupt changes in terms of article content seems to be an indicator of frustration or confusion.

The same features were prominent in Tasks 2 and 3, with the addition that the two features used specifically in those two tasks (number of clicks already made so far, and fraction of back clicks) were also found to be important.

7. DISCUSSION AND CONCLUSION

In this paper, we investigate why and when users abandon click-based navigation tasks in a given information network. Even though billions of navigation paths are given up when browsing websites or navigating Wikipedia, the navigational patterns that lead to task abandonment have largely remained underexplored. Many previous empirical analyses of navigation behavior have suffered from shortcomings imposed by the datasets studied. The target pages users attempted to reach were not explicitly known, so it was not clear whether they stopped navigating because the target was already found or because, on the contrary, they were unable to find it. In contrast, by studying navigation paths from a human-computation experiment in which users are navigating Wikipedia by clicking from page to page, we overcome this obstacle. Since users are explicitly told what they are supposed to find, the target of the navigation task is known and can be factored into the analysis.

Our analysis demonstrates that some missions are intrinsically harder than others, determined primarily by factors dependent on the navigation target and not the source. However, since the target is explicitly known, we can control for this intrinsic mission hardness and focus on which kinds of navigation strategies are more likely to result in success than others.

We find that it is easier to make progress at the beginning of paths. In both successful and unsuccessful missions, users navigate through high-degree hub nodes. However, it is homing in on the target that makes the difference. Unsuccessful users keep orbiting around the target and eventually get frustrated, leading to abandonment, whereas, successful users manage to find the crucial link that lets them gravitate into the target.

We also find that backtracking plays an important role and is employed in both successful as well as unsuccessful paths. In some cases, it could be used by people who are gathering useful information by exploring the local neighborhood, aiding their navigation, whereas, in other cases, it could indicate that the user is confused and is desperately trying to find a way to the target. In both successful and unsuccessful missions, users prefer backtracking to hubs of high out-degree, which serve as landmarks from which many different parts of the graph can be explored. Interestingly, both successful and unsuccessful users prefer backtracking after making a bad click, i.e., one that increases the shortest-path length or TF-IDF distance to the target. However, a distinctive difference between the two types of user is that successful navigators are better at realizing when they have made progress. Unsuccessful users are three times more likely to backtrack after making progress towards the target.

Building on these insights, we formulate predictive models for three tasks, showing that our analytic findings can be applied to

predict user behavior. Our models have the potential to be useful in practice, such as aiding website owners to improve navigability, or user experience in general. However, applying the models to build real-world tools also poses unique challenges. For example, some of our features depend on the target, which is explicitly known in our case, but which is normally unknown in real use cases. Thus, an advantage and at the same time also the limitation of our current approach and model is that the target of the navigation is explicitly known ahead of time. However, we argue that target prediction could be treated as an orthogonal problem and then be combined with the method we introduce here. For instance, West and Leskovec [16] propose an algorithm for target prediction, given only a few initial clicks.

Our user experiments were conducted entirely within the Wikipedia hyperlink network. This enables more powerful analyses, compared to log-based analyses of users browsing the ‘wild Web’, since in our setup, each page has a clearly defined topic, and since the full network topology is known. On the flip side, constraining the navigation environment to Wikipedia poses a *caveat* to the generalizability of our results, since the placement of hyperlinks in Wikipedia follows rather well defined rules, making it easier, and possibly qualitatively different, for users to find their way on Wikipedia than on the ‘wild Web’.

Further, real navigation settings tend to involve additional actions beyond clicking from page to page, such as search engine queries. Rather than beginning on a page that is unrelated to the navigation target (as in our data), users will typically use a search engine to be ‘teleported’ to a page that is more or less related to the final target, followed by some click-based navigation to zero in on the exact target page [1, 6, 12]. When this fails, query reformulation offers yet another recovery mechanism, in addition to the backtracking strategies we study in this work. On the one hand, it is an advantage of our experimental setup that it lets us study click-based navigation in isolation, without confounding users’ pathfinding skills with their query formulation skills. On the other hand, we anticipate opportunities for highly interesting future research in extending our analysis of abandonment to more holistic settings involving both query- and click-based navigation, in order to understand the interplay between these two modes of information search.

In summary, we have made contributions towards understanding what causes users to give up navigating towards a target in information networks. We hope that future work will draw on our insights to build tools to help users navigate websites more easily to find what they are looking for.

Acknowledgements

This research has been supported in part by NSF CAREER IIS-1149837, ARO MURI, Okawa Foundation, Allies, and Volkswagen. Robert West acknowledges support by a Stanford Graduate Fellowship and a Facebook Graduate Fellowship.

8. REFERENCES

- [1] M. Ageev, Q. Guo, D. Lagun, and E. Agichtein. Find it if you can: A game for modeling different types of Web search success using interaction data. In *SIGIR*, 2011.
- [2] S. Brin and L. Page. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 30(1), 1998.
- [3] E. H. Chi, P. Pirolli, K. Chen, and J. Pitkow. Using information scent to model user information needs and actions and the Web. In *CHI*, 2001.

- [4] A. Das Sarma, S. Gollapudi, and S. Ieong. Bypass rates: Reducing query abandonment using negative inferences. In *KDD*, 2008.
- [5] A. Diriye, R. W. White, G. Buscher, and S. Dumais. Leaving so soon? Understanding and predicting Web search abandonment rationales. In *CIKM*, 2012.
- [6] D. Downey, S. Dumais, D. Liebling, and E. Horvitz. Understanding the relationship between searchers' queries and information goals. In *CIKM*, 2008.
- [7] M. Harrigan, D. Archambault, P. Cunningham, and N. Hurley. EgoNav: Exploring networks through egocentric spatializations. In *AVI*, 2012.
- [8] J. Kleinberg. The small-world phenomenon: An algorithmic perspective. In *STOC*, 2000.
- [9] J. Li, S. Huffman, and A. Tokuda. Good abandonment in mobile and PC Internet search. In *SIGIR*, 2009.
- [10] S. Milgram. The small world problem. *Psychology Today*, 2, 1967.
- [11] C. Olston and E. H. Chi. ScentTrails: Integrating browsing and searching on the Web. *TCHI*, 10(3), 2003.
- [12] J. Teevan, C. Alvarado, M. S. Ackerman, and D. R. Karger. The perfect search engine is not enough: a study of orienteering behavior in directed search. In *CHI*, 2004.
- [13] C. Trattner, D. Helic, P. Singer, and M. Strohmaier. Exploring the differences and similarities between hierarchical decentralized search and human navigation in information networks. In *i-KNOW*, 2012.
- [14] T. von Landesberger, M. Gerner, and T. Schreck. Visual analysis of graphs with multiple connected components. In *VAST*, 2009.
- [15] R. West and J. Leskovec. Automatic versus human navigation in information networks. In *ICWSM*, 2012.
- [16] R. West and J. Leskovec. Human wayfinding in information networks. In *WWW*, 2012.
- [17] R. West, J. Pineau, and D. Precup. Wikispeedia: An online game for inferring semantic distances between concepts. In *IJCAI*, 2009.
- [18] R. W. White and J. Huang. Assessing the scenic route: Measuring the value of search trails in Web logs. In *SIGIR*, 2010.
- [19] R. W. White and D. Morris. Investigating the querying and browsing behavior of advanced search engine users. In *SIGIR*, 2007.
- [20] R. W. White and A. Singla. Finding our way on the Web: Exploring the role of waypoints in search interaction. In *WWW*, 2011.