Strathprints Institutional Repository

Azzopardi, Leif and Zuccon, Guido (2016) An analysis of the cost and benefit of search interactions. In: ICTIR '16 Proceedings of the 2016 ACM International Conference on the Theory of Information Retrieval. ACM, New York,, pp. 59-68. ISBN 9781450344975 , http://dx.doi.org/10.1145/2970398.2970412

An Analysis of the Cost and Benefit of Search Interactions

Leif Azzopardi
University of Strathclyde
Glasgow, United Kingdom
leifos@acm.org

Guido Zuccon
Queensland University of Technology
Queensland, Australia
g.zuccon@qut.edu.au

ABSTRACT

Interactive Information Retrieval (IR) systems often provide various features and functions, such as query suggestions and relevance feedback, that a user may or may not decide to use. The decision to take such an option has associated costs and may lead to some benefit. Thus, a savvy user would take decisions that maximizes their net benefit. In this paper, we formally model the costs and benefits of various decisions that users, implicitly or explicitly, make when searching. We consider and analyse the following scenarios: *(i)* how long a user's query should be? *(ii)* should the user pose a specific or vague query? *(iii)* should the user take a suggestion or re-formulate? *(iv)* when should a user employ relevance feedback? and *(v)* when would the "find similar" functionality be worthwhile to the user? To this end, we build a series of cost-benefit models exploring a variety of parameters that affect the decisions at play. Through the analyses, we are able to draw a number of insights into different decisions, provide explanations for observed behaviours and generate numerous testable hypotheses. This work not only serves as a basis for future empirical work, but also as a template for developing other cost-benefit models involving human-computer interaction.

Keywords

Search Behaviour; User Models; Retrieval Strategies; Evaluation, Measures.

1. INTRODUCTION

Information Retrieval and Seeking activities take place in the context of a task (typically a work task) [29]. The evaluation of Information Retrieval has however largely focused on measuring and modeling the performance of a system based on an abstraction of the search process (e.g. the TREC/Cranfield paradigm [28, 56]). While it has been long recognised that such paradigms ignore many factors [28, 49], there has only recently been a drive to create new evaluation measures that go beyond precision and recall, incorporating: *(i)* more sophisticated user models that encode stopping and interaction behaviour [39, 42], *(ii)* the gain and usefulness of the documents encountered [13, 20, 30] *(iii)* the cost and effort of performing different interactions [2, 50, 62, 63] and *(iv)* the sequence of interactions within and across the sessions (as opposed to considering ranked lists in isolation) [4, 5, 31, 46]. Underpinning most measures is a user model [59]; so there has been a concerted effort to improve the current user models to develop more realistic and accurate measures [40]. While much of the research has focused on how users interact with a ranked list (resulting in numerous measures [17, 40]), less attention has been paid to modeling other interactions. Now with the drive towards task based evaluations and task completion engines [11, 13, 58], it is timely to consider modeling interactions outwith the ranked list, and to understand how the different choices users can make affect their overall session performance. Two common threads run through this recent work.

(i) The **cost** of interaction, through the inclusion of the effort or time involved in the search process within the evaluation measure. For example, measures like Time Biased Gain (TBG) [50] specifically incorporate the time to assess documents. While the effort involved in processing a document in terms of readability and understandability has been explicitly included in other measures (e.g., [2, 62, 63]).

(ii) The **benefit** of interaction, through the inclusion of a gain, graded relevance or the value of the information found. For example, the U-measure values shorter documents more than longer document [46]; while in RBP, DCG and variants of [18, 30, 39, 42], documents seen at a later point in the session are valued less.

A key assumption underpinning the modeling and measuring of user-system performance is that a system or search strategy results in greater net benefit (or utility)[1] is preferable than one that yields lower net benefit . Indeed, it has been posited that people will modify their search strategies in order to maximize their net benefit, and that systems and interfaces will evolve so as to maximize the net benefit for the user [43]. For example, an instantiation of the card playing model [61] shows how the interface can be adapted to maximize the information gain at each round of interaction. On the other hand, numerous attempts have been made to augment the standard search interface but have been met with limited success [33]. Taken together, this body of work

[1] Utility (or expected utility), expressed as the gain or gain over time.

motivates the development of user models that consider the costs and benefits of the decisions involved in searching and seeking so that we can better understand the process and support it appropriately through the interfaces and systems we develop. Consequently, in this work we begin building a series of cost-benefit models of various interactions from a user perspective. This allows us to: *(1)* reason about why users perform certain actions over others, *(2)* understand why certain features or interactions are preferable over others, and *(3)* develop core underlying user models for the development of task and session based measures. To this end, we consider different decisions the user faces when querying and interacting with the system/interface, where we consider questions such as:

- why is it so hard to get users to type longer queries?
- why are users reluctant to give relevance feedback?
- and why is the standard search interface preferable to "novel" search interfaces?

Through our analyses, we draw various insights regarding user behaviour and generate numerous testable hypotheses which can be explored in future work.

2. RELATED WORK

Early on in the history of IR, the notion of utility featured heavily and gave rise to the strong evaluation based traditions in the field [56]. In [22], Cooper put forward the proposition that utility forms the basis of measurement claiming that it would be the *ideal measure*. However the difficulty in obtaining judgements of utility meant that the field has resorted to relying upon the simple demarcations of non-relevant and relevant. Nonetheless, the idea of using the utility of a document has arisen in various guises (i.e. graded relevance, gain, benefit, usefulness and negative cost) and has formed the basis of much research. For example, in [44], Robertson examined the problem of ranking in terms of the costs and benefits. This led to the formulation of the Probability Ranking Principle (PRP) which essentially applies decision theory to the ranking problem [44]. The PRP makes a number of assumptions which are also implicit within most measures used, reflecting the user model, i.e. that documents are judged/valued independently, the costs/benefits of (non) relevant documents are the same, and that documents are judged in a linear fashion. Furthermore, most measures and models have focused on evaluating a ranked list. However, there is now impetus to go beyond the ranked list, and consider the whole session in the context of the task [4, 5, 11, 31, 46]. Thus, we need to revisit the idea of modeling the utility of the information found and the actions in the sessions that lead to ascertaining that utility. Here, we consider how much utility (or usefulness) one obtains in terms of the *costs* and *benefits* as suggested by Bates [12]. In [12], Bates describes one of the monitoring tactics people employ when searching is to *weigh* the costs and benefits of their decisions/interactions. While Bates did not elaborate on this tactic, a line of research has evolved from this notion which formally models the costs and benefits of interaction using different, but related, frameworks [45, 43, 6, 25]. For instance, in [45], they examine the cost structures associated with sense-making, and in [43] Pirolli adapts Foraging Theory to explore how searchers attempt to maximise the gain (benefit) over time.

A Note on Relevance, Cost & Benefit

In the literature, the *cost* and *benefit* of documents, information and interactions have been discussed and introduced in a variety ways. *Benefit*, or the value associated with a document, has been referred to as: *relevance*, *benefit*, *gain*, *utility*, *expected utility* and *usefulness* [7, 17, 20, 22, 25, 30, 44, 60]. For example, early on relevance was associated with utility [22] in the sense that one receives benefit or gain from a document that is relevant. Of course, what constitutes relevance is subject to much interpretation [38]. However, the point is that most researchers acknowledge that users are after some useful information to help them facilitate the completion of a task. How this usefulness, benefit, gain, etc. is measured is an open challenge. While in terms of *cost*, researchers have considered mental/cognitive, physical, financial and temporal costs [6, 26, 50]. Often time is considered as a proxy for cost, or as an independent variable to contextualize the rate of gain [43, 50]. Recently the cognitive costs and the effort in processing, reading and understanding a document have also been considered [2, 50, 62, 63]. However costs are difficult to define, quantify and measure, and this is also an open challenge in the field.

In this work we will develop and discuss the costs and benefits as abstract, but common, units. However, one can think about many of the models with respect to time or money as the cost and benefit [25], i.e., how much time do I have to spend, and how much time do I save? Or how much money do I have to spend, and how much money do I save or earn? In this paper we mainly use the terms cost and benefit, but may use some of the terms above interchangeably, where appropriate, and depending on the context.

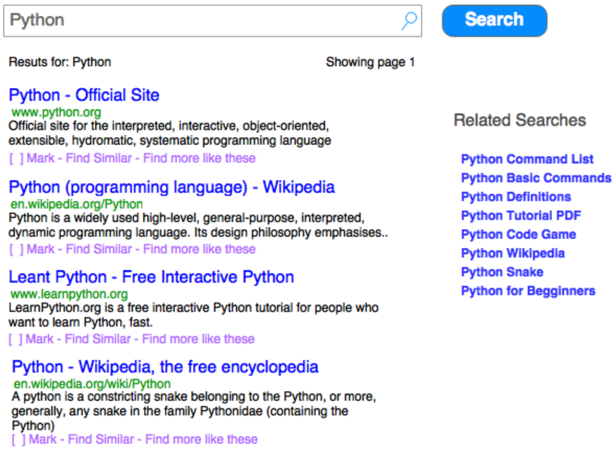
Related Models

As previously mentioned, most measures are underpinned by a user model pertaining to how users interact with a ranked list of documents. For example, precision at $\boldsymbol{k}$ assumes that the user examines documents up until rank $\boldsymbol{k}$ and then stops. Furthermore, the measure implicitly assumes that the cost of processing each document is the same, and the value obtained from each document is also the same regardless. However, there has been a succession of innovations in modeling how the user interacts with a ranked list which has led to new measures (e.g. RBP [42], DCG [30], INST [39], etc.) and analyses of the relationship between measures and user stopping models [17, 40, 41]. Recently, other advances are being introduced into measures. For example in [48], Smucker empirically explores how changes in the cost of reading snippets and the quality of snippets affects the gain in terms of the number of documents. Later, Smucker and Clarke [50] extended this work by incorporating the time to process snippets and documents into TBG as a way to evaluate the gain over time. In [2], Arvola et al. also consider the effort required to read, creating measures of the expected reading effort. Other work shows that the effort (measured in time) to judge documents varies depending on the relevance [55, 60], again demonstrating that both cost and gain need to be considered. An excellent summary of such user models and how different measures employ them is provided in [41], while in [17] Carterette demonstrates that these IR evaluation measures can be interpreted as utility (or expected utility) when these user stopping models are used, thus bringing together cost, benefit and expectation resulting from the search interaction. Other researchers have focused on modeling the interaction with the ranked list by developing click models [19] such as the cascade model [23]. However, to model (and thus measure) the whole session and the task performance, it is necessary to consider the range of search interactions as well as other components of the search interface.

The focus of this paper is on other less considered aspects of the search process; rather than trying to measure, we focus on modeling the interactions to draw insights and reason about the behaviours and interactions of users. In this direction, previous research has modeled various aspects of the information seeking and search process. For example, in [21], Cooper models the costs and benefits of various searchers (the user and the librarian/system) to understand the trade-off between how much time each party should spend searching. In [54], Varian outlined how we could apply Stigler's theory of Optimal Search Behaviour to IR [53] and how to examine the economic value of information using consumer theory, "where a consumer is making a choice to maximize expected utility or minimise expected cost" [54]. This lead to various insights such as a document has little or no value if the information it contains is redundant. In [16], Birchler and Butler also explain how Stigler's theory can be applied to search in order to predict when a user should stop examining results in a ranked list, i.e., when the marginal benefit equals the marginal cost. While in [25], Fuhr generalizes the Probability Ranking Principle such that documents have different costs and benefits and that there is some uncertainty over accepting a decision.

Information Foraging Theory [43] (IFT) examines the profitability of items found by considering the net gain obtained given time spent (which is what TBG [50] attempts to measure). Under IFT, it is possible to model various situations, such as when the user should stop examining results and move to another patch. Interestingly, in [8] this was shown to make the same predictions as the iPRP and Search Economic Theory [5]. In [5], Azzopardi considers the economics (in terms of gain and cost) to examine the tradeoff and interplay between querying and assessing. This session-based model ascribes a cost and gain functions given the two different interactions. In follow up work this was extended to include inspecting snippets and paging [7]. The model enables various hypotheses to be generated depending on the changes in the performance (gain) from the interactions with the system and the cost of the various interactions [7].

In [32], Kashyap et al. define a cost model for browsing facets to minimize cost of interaction and thereby increasing the usefulness of the interface. In [9], Azzopardi and Zuccon model the browsing costs on the Search Engine Result Page (SERP), where they consider the size of the screen, the results viewable and the number of documents to browse through. While simple, their model is instructive in understanding the relationship between scrolling and paging. In [51], Smucker and Clarke model the switching behaviour of users engaging with ranked lists which provide different levels of gain. They show at what point it is optimal to switch. We continue in this line of research and develop a number of simple models of various search interactions to add to the growing catalogue of user models for search interaction.



Resuts for: Python

Showing page 1

Python - Official Site
www.python.org
Official site for the interpreted, interactive, object-oriented, extensible, hydromatic, systematic programming language
[ ] Mark - Find Similar - Find more like these

Python (programming language) - Wikipedia
en.wikipedia.org/Python
Python is a widely used high-level, general-purpose, interpreted, dynamic programming language. Its design philosophy emphasises..
[ ] Mark - Find Similar - Find more like these

Leant Python - Free Interactive Python
www.learnpython.org
LearnPython.org is a free interactive Python tutorial for people who want to learn Python, fast.
[ ] Mark - Find Similar - Find more like these

Python - Wikipedia, the free encyclopedia
en.wikipedia.org/wiki/Python
A python is a constricting snake belonging to the Python, or more, generally, any snake in the family Pythonidae (containing the Python)
[ ] Mark - Find Similar - Find more like these

Related Searches

Python Command List
Python Basic Commands
Python Definitions
Python Tutorial PDF
Python Code Game
Python Wikipedia
Python Snake
Python for Begginners

Figure 1: A search interface with related searches and the option to mark relevant documents, find similar and find more like the ones marked. For the query, "*Python*" the page displays results mostly about the Python programming language.

3. AIMS AND METHODOLOGY

The main aim of this paper is to build a number of models regarding different decisions, i.e. actions, interactions, choices that users can make when using a search system and interface under a common framework, a cost-benefit analysis. Under the cost-benefit analysis, costs and benefits are considered to be in the same units. A more general approach is a cost-utility analysis, where the cost and benefit are expressed in different units (i.e. benefit might be expressed as the number of relevant documents found and the cost expressed as time). However, as a starting point, for modeling these interactions, we will assume that the costs and benefits are in the same (albeit abstract) units, and thus use a cost-benefit framework. During the course of this paper we will consider the following scenarios: query length, query diversity, query suggestions, relevance feedback, and find similar. For each scenario, we will model salient aspects of the interaction in order to gain various insights.

Before we begin creating models of these different scenarios, we first ground our models based upon the search interface depicted in Figure 1. We then enumerate the different variables that we will use, followed by our approach.

3.1 Search Interface

For the purposes of modeling various interactions we base our analyses on the standard search interface which includes a query box, result snippets and navigational buttons (previous and next). Each snippet has a title (a blue link), a snippet from the document (black text), and url/domain (green text). However, we have also included a few extra features: *(i)* related searches / query suggestions, *(ii)* an option to find similar, and *(iii)* an option to mark documents as useful and then to find more like those marked. The later two features are different versions of explicit relevance feedback which we shall model[2]. In various works, researchers have

[2] We note that most interfaces don't provide such options or provide

$c(.)$	cost function
c_w	cost to enter a word
c_i	total cost of interaction
c_q	cost of issuing a query
c_s	cost of examining a snippet
c_a	cost of assessing a document
c_c	cost of a click
c_d	cost of deciding to mark and undertake relevance feedback
c_m	cost of judging and then marking a document as relevant (for relevance feedback)
c_{rq}	cost of remembering the issued query
c_{es}	cost of examining query suggestions

Table 1: Notation used for cost functions and associated variables.

$b(.)$	benefit function
b_{fp}	benefit provided by the first SERP
b_{np}	benefit provided by the next SERP
b_{nq}	benefit provided by the next query
b_{rf}	benefit provided by performing relevance feedback

Table 2: Notation used for benefit functions and associated variables.

experimented with different interfaces and techniques to increase the net benefit to the user. It is therefore of interest to explore the cost-benefit relationships formally. During the course of interaction, a user poses a query, waits for the page to load, examines the page, and a number of snippets, along with examining a number of documents. The user will decide at some point either to stop browsing the current page of results, and move onto the next page, reformulate their query, take a query suggestion, provide feedback or stop searching.

3.2 Notation and Preliminaries

Costs and benefits are associated to interactions. For example, users pay a cost when formulating a query (c_q) or assessing a document (c_a), but they may also extract some benefit when exposed to the information contained in a document. Table 1 provides the different costs for various interactions, while Table 2 provides an overview of the benefits from different interactions. Note that we have defined interactions at an action level (as opposed to a keystroke level) because different interactions can be implemented differently and so depend on the instantiation of the feature. In our analyses, we will loosely base the cost on the time it takes to perform such actions, where we ground the costs based on the values in [6, 50, 47]. Though the costs (and benefits) used in the examples (see Figures 2-5) are purely to illuminate the equations - estimating the costs and benefits is left for future work. During the discussion of the models, we will explain how the cost could change for different interactions depending on how it is implemented, and how this would affect the decision users take.

3.3 Methods of Analysis

To analyse each of the different scenarios, we will first define and enumerate the different costs and benefits associated with the different actions. Given these cost and benefit functions, we will then employ a number of related techniques to determine: *(a)* how the net benefit π (or profit, i.e., benefit minus cost) changes as another variable of interest changes, or *(b)* whether one course of action leads to greater profit over another course of action. In our analysis we assume, like much of the previous work [5, 25, 43], that users will be rational in the sense that they will try to maximize their net benefit from interacting with the system.

π	profit (or net benefit)
W	number of query words
Z	number of aspects associated with a query
N	number of documents the user seeks
M	number of documents to be marked for relevance feedback
A	number of assessed documents
R	number of result in a SERP
S	number of snippets examined
Q_s	number of query suggestions

Table 3: Notation for other variables common across models.

The first method of analysis will be to examine the change in profit as the variable of interest (e.g., number of query words) changes. Here, we can determine when the profit is maximized by taking the derivative of the profit function, setting it to zero, and then solving.

The second method we employ is when we have two alternatives/decisions and we need to determine under what conditions one decision is preferable to the other. In this case, the profit resulting from each decision is compared using an inequality, to determine which decision is more profitable.

The third method we employ is similar to the second approach, except that we only consider the costs involved in the decision and assume the benefit is the same for both decisions.

4. QUERY INTERACTIONS

As argued in [4, 40], performance is determined by both the user and the system. How well a user can use the system will depend on what decisions they make. The initial decisions a user makes when starting a search session is what query to pose, how long, and how specific.

4.1 Querying Length

First, we consider the questions, why do user queries tend to be short, and how can we motivate users to type longer queries? In order to provide insights into these questions, we need to consider what factors affect the benefit of a query. While many factors are at play [29], one of the main drivers is query length. It has been shown on numerous occasions that longer queries tend to yield better performance [3, 14, 24]. Consequently, this has led to various attempts to try and elicit longer queries from users (e.g. [1, 34, 35]). However, these attempts have largely been ineffectual: we posit this is due to the costs and benefits at play. We further posit that inline autocomplete techniques are instead more effective in eliciting longer queries because they substantially lower the cost of entering a query while increasing the benefit.

We model the decision to issue a query of a particular length W to denote the number of words as follows: the benefit that a user receives is given by the benefit function $b(W)$ and the cost (or effort in querying) defined by the cost function $c(W)$. Now let's consider a benefit function

either the find similar or the find more like these options. We show them on the same interface for the purposes of illustration.

which denotes the situation where the user experiences diminishing returns such that as the query length increases they receive less and less benefit (as shown in [3, 14]). This can be modeled with the function:

$$b(W) = k.\log_a(W+1) \tag{1}$$

where k represents a scaling factor, and a influences how quickly the user experiences diminishing returns. Let's assume then that the cost of entering a query is a linear function based on the number of words such that:

$$c(W) = W.c_w \tag{2}$$

where c_w represents how much effort must be spent to enter each word. This is, of course, a simple cost model and it is easy to imagine more complex cost functions. However the point is to provide a simple, but insightful, abstraction. Now given these two functions, we can compute the profit (net benefit) π that the user receives for a query of length W:

$$\pi = b(W) - c(W) = k.\log_a(W+1) - W.c_w \tag{3}$$

To find the query length that maximizes the user's net benefit, we can differentiate and solve the equation:

$$\frac{\partial \pi}{\partial W} = \frac{k}{\log a} \times \frac{1}{W+1} - c_w = 0 \tag{4}$$

This results in:

$$W^{\star} = \frac{k}{c_w.\log a} - 1 \tag{5}$$

Figure 2 illustrates the benefit (top) and profit (bottom) as query length increases. For the left plots $k = 10$, and for the right plots $k = 15$. Within each plot we show various levels of a. These plots show that as k increases (i.e. overall the performance of the system increases), then the model suggests that query length, on average, would increase, and if a increases (i.e. the performance of the system increases but at a faster rate), then queries decrease in length. Furthermore the model suggests that as the cost of entering a word, c_w, decreases users will tend to pose longer queries.

Summary

This simple model shows that to motivate longer queries either the cost of querying needs to decrease or the performance of the system needs to increase (via k and a). Ergo, reducing the cost of querying by providing inline autocomplete suggestions reduces the cost and increases the performance sufficiently that it motivates longer queries on average being issued. It further suggests that techniques to encourage the user to issue longer queries are insufficient on their own. They instead need to be backed up by increases in system performance or reductions in user interaction cost.

4.2 Specificity and Diversity

The next question we consider is whether it is better to pose a vague query or more specific query?[3]. This, of course, depends on how the system responds to such requests by returning diversified or non-diversified results.

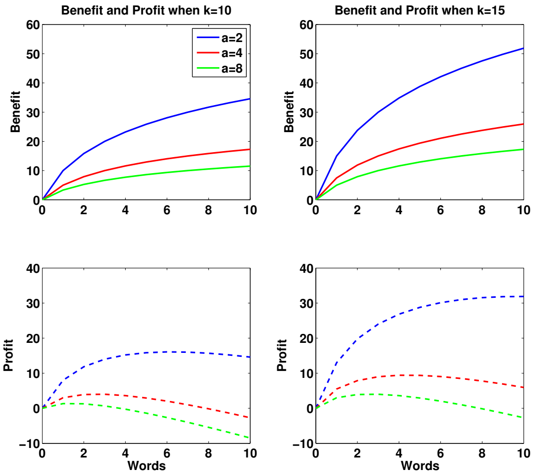


Figure 2: The top plots show the benefit while the bottom plots show the profit as the length of the query increases. Plots on the right show when the queries yield greater benefit. Each plot shows three levels of α which denotes how quickly diminishing returns sets in.

To provide the context, let's consider the following scenario. For simplicity we assume that the user is seeking N documents and their initial query is underspecified as it may refer to Z possible aspects/intents/interpretations. For example, if the user poses the query "*Python*" as shown in Figure 1, then there are several possible interpretations such as: the language, the snake, the movie, etc. We shall refer to these as *aspects* in the remainder of this paper.

Let's consider now how a system might respond. The system could take a diversified approach and provide some coverage of the Z aspects. Or it could take a non-diversified approach returning documents that correspond to the most popular or dominant aspect and thus return documents only from one aspect. Which approach would a user prefer? Or stated another way, under what circumstances would interacting with diversified results be less costly than interacting with non-diversified results? To answer this questions we consider how the user might interact with these different result lists, and then compare the costs and benefits.

When interacting with the diversified result list, the user pays a cost for issuing the query, c_{q_1}. For simplicity, we assume the diversified response blends the different aspects together uniformly so that approximately 1 in every Z results refers to a relevant aspect. This means the user would have to process $N.Z$ snippets to find the N results about that aspect in the best case. On average however, it is likely that only a proportion would be relevant, say p_r. Thus, on average they would have to inspect $\frac{N.Z}{p_r}$ snippets, but for simplicity we assume that $p_r = 1$.

With the cost of examining a snippet being c_s, the cost of interacting with the diversified results (c_i^D) is:

$$c_i^D = c_{q_1} + c_s.N.Z \tag{6}$$

On the other hand, the cost of interacting with the non-diversified results is slightly different. The user again pays the cost for issuing the first query, c_{q_1}. If the query retrieved documents for the correct aspect, the user only needs to examine N snippets, resulting in a cost of $c_s.N$ for examining

[3] While specificity is related to length we wish to consider this separately for the time being, and leave considering the relationship between specificity and length for future work.

all the snippets. For simplicity we further assume that the system returns the aspect the user had in mind with a uniform probability $1/Z$ [4].

If instead the query did not retrieve documents for the correct aspect, the user needs to issue a more specific query, paying an additional querying cost c_{q_2}. Since q_2 would typically mean adding one or more terms to the original query, then c_{q_2} is likely to be less than c_{q_1}. However, before they issue a new query, the user first examines k snippets, before realising that the response is inadequate. They then decide to re-formulate. For simplicity, we will assume that the more specific query means that the system will respond with the correct aspect (i.e., "python snake"). Of course, within this aspect, there could be sub-aspects, but again we will assume for simplicity that the re-formulated query is sufficient to retrieve documents that correspond to the user's intent. In this case, they reformulate with probability $1 - 1/Z$. Thus, the total cost of interaction with a non-diversified system ($c_{i,ND}$) is:

$$c_i^{ND} = c_{q_1} + \frac{1}{Z}c_s N + \Big(1 - \frac{1}{Z}\Big)\Big(c_{q_2} + c_s(N+k)\Big) \quad (7)$$

Given the two cost models, we can compare the two user-system interactions to determine if it is better to *(a)* pose an underspecified query and interact with the diversified result list, or *(b)* pose an underspecified query and interact with the non-diversified result list, then reformulate the query to retrieve the aspect of interest:

$$\begin{aligned} c_i^D &< c_i^{ND} \\ c_{q_1} + c_s N.Z &< c_{q_1} + \frac{1}{Z}c_s N \\ &\quad + \Big(1 - \frac{1}{Z}\Big)\Big(c_{q_2} + c_s(N+k)\Big) \\ c_{q_2} &> c_s(N.Z - k) \qquad (8) \end{aligned}$$

Following Inequality 8 and as shown in Figure 3, in the presence of many aspects or when the user is seeking many documents for the intended aspect, it is best not to have to interact with a diversified list. Conversely, if the user has a navigational intent ($N = 1$) or is after a small number of documents, then it is better to interact with a diversified list. Alternatively if the ambiguity of the initial query (measured by the number of aspects Z) is low, then the user is also better off interacting with diversified results.

Summary

To derive the costs of interactions we made a number of simplifying assumptions; we revisit them next in light of Inequality 8. We assumed the diversified system would uniformly blend results from different aspects. However, more sophisticated strategies for blending results, e.g., by sampling with higher rate from popular aspects, may on average reduce the cost of interaction with the diversified system. When considering the non-diversified system, we also assumed that a user could issue a more specific query with cost c_{q_2} and this query would retrieve documents relevant to the desired aspect of interest. Often this may not be the case and it may take the user more than one query reformulation to obtain a more specific query to find the correct results. For example, consider the case where two query refinements are required to formulate a query that returns results for the correct aspect. In these circumstances, the cost of interaction becomes:

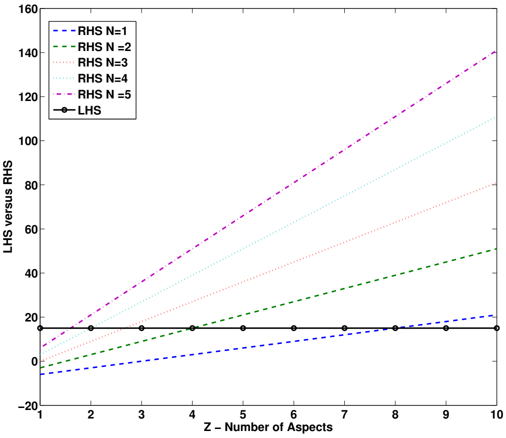


Figure 3: Plots of the left-hand side (LHS) and right-hand side (RHS) of inequality 8 where $k = 3$, showing that it is better to pose a more specific query as N and Z increase (i.e., when above the black dotted line denoting the LHS).

$$c_i^{ND} = c_{q_1} + \frac{1}{Z}c_s N + (1 - \frac{1}{Z})\Big[c_{q_2} + (\frac{1}{Z-1})c_s(N+k) + (1 - \frac{1}{Z-1})\big(c_{q_3} + c_s(N+2k)\big)\Big]$$

When this is considered in Inequality 8, we find that diversifying search results is more attractive than before as, all other values being the same, the cost of interaction with the diversified system attracts one less cost than using the non-diversified system.

Admittedly, the above model is rather naive and there are numerous ways in which to improve it. For example, including the probability distribution over aspects, including a more sophisticated interaction with the ranked list and adding in further query refinements. Another refinement would be to link the length of the query (in terms of specificity) with the number of aspects. For example, consider the queries "*python*", "*python tutorial*", and "*python 3 tutorial*". Each are progressively longer and more specific - which also impact on how many aspects are returned. It would be interesting to consider how the user can control and manipulate the system and how worthwhile it is for them to do so.

4.3 Query Suggestions

Next, we consider whether a user should take a query suggestion or not. Observe the SERP in Figure 1 where the query "*python*" has been issued. Various query suggestions have been presented to the user that expose various aspects associated with "*python*", i.e., "*python commands*", "*python snake*", "*python tutorial*", etc.

Now let's consider the following scenario. A user enters a query to the system and the system does not retrieve any relevant document. Let's further assume that this is because the query is underspecified or impoverished in some way. The user now has the choice of reformulating the query: *(a)*

[4] More complex weighted distributions are left for future work.

by making it more specific, or *(b)* by taking a query suggestion. The cost of taking a suggestion involves examining the list of query suggestions provided, which we model as a function $c_{es}(.)$, where the cost is proportional to the number of suggestions, Q_s. If one of the suggestions is sufficiently specified, then the user will take the suggestion with some probability p_s. However, if the suggestions are not specific enough then the user resorts back to issuing a new query:

$$\begin{aligned} c_{q2} &< c_{es}(Q_s) + p_s.c_c + (1 - p_s).c_{q2} \\ p_s.c_{q2} - p_s.c_c &< c_{es}(Q_s) \\ c_{q2} - c_c &< \frac{c_{es}(Q_s)}{p_s} \end{aligned} \tag{9}$$

While the cost of clicking on a suggestion is relatively cheap, the main factors influencing the selection of a suggestion is the cost of the subsequent query versus the cost of examining the suggestions and the probability of one of the suggestions being useful. In the best case scenario, the next query to be issued is within the suggestions i.e., $p_s = 1$, in which case the selection cost is not amplified. What is interesting in this case is that if there is some uncertainty as to whether the suggestions will contain a good next query, or if the cost of processing the suggestions is high, then re-formulating is preferable. In terms of processing the suggestions, more suggestions are likely to increase the cost, but also having similar alternatives is going to increase the cost because the user has to work out if they have different meanings (or not) and if they will result in different results (or not). On the other hand, if the user is finding it hard to think of a subsequent query, i.e. has run out of terms or ideas for querying, then it might be cheaper to process the suggestions. However, if the suggestions fail to provide an adequate query then the user must revert back to reformulating (or some other action).

5. FEEDBACK

In this section we consider the question, is relevance feedback worth it? Let's consider an interface that allows users to mark which documents they think are relevant, and offers them the choice to find more documents like the ones selected (via the relevance feedback options on the SERP in Figure 1). An alternative interface would be one that provides a find similar button for each document instead. We will consider both options next.

5.1 Relevance Feedback

Let's assume that the user has issued a query to the system, and that the system has returned a set of results. The user has several options; among those we consider: *(1)* examining the results on the first page, then moving to the next page (np), *(2)* examining the results on the first page, then issuing a new query (nq), *(3)* examining the results on the first page, marking which documents are relevant, and then clicking the "find more like this" button (rf).

Next, we consider the costs and benefits in each of these scenarios. Let b_{fp}, b_{np},b_{nq},b_{rf} be the gain of the first page, the gain from moving to the next page, the gain from the next query and the benefit from performing relevance feedback, respectively. The corresponding costs can be formulated as follows:

$$c_{np} = c_q + N.c_a + c_c + N.c_a \tag{10}$$

$$c_{nq} = c_q + N.c_a + c_q + N.c_a \tag{11}$$

$$c_{rf} = c_q + N.c_a + c_d + M.c_m + c_c + N.c_a \tag{12}$$

where c_q is the cost of entering a query, c_a is the average cost to examine a document, c_c is the cost to click to the next page or click to find more, c_d is the cost of deciding to mark and undertake relevance feedback, c_m is the average cost of judging (explicitly) and then marking the document as relevant, where M documents are marked.

First, let's consider the case where the benefit from the next query is the same as the benefit from relevance feedback. In this case, the user should opt to re-querying when:

$$\begin{aligned} c_{nq} &< c_{rf} \\ 2.c_q + 2.N.c_a &< c_q + 2.N.c_a + c_d + M.c_m + c_c \\ c_q &< c_d + M.c_m + c_c \end{aligned} \tag{13}$$

Here we can see that if the cost of querying is less than the cost to decide, mark and find more, then querying is preferable. Further we can see that the number of documents to mark M is based upon the following inequality:

$$M > \frac{c_q - c_d - c_c}{c_m} \tag{14}$$

In Figure 4, we have plotted Inequality 14, where we assume the costs for clicking and deciding are $c_c = 2$ and $c_d = 2$, respectively, and the cost of querying and assessing are $c_q = 15$ and $c_a = 20$, respectively. The figure shows the LHS for M from 1 to 5 versus the RHS where the cost of marking c_m increases from 1 to 5. From this plot, we can see that if the cost of marking is low, then the LHS is less than the RHS (i.e. the inequality does not hold) and so relevance feedback is preferable. As the cost of marking increases, then the option of relevance feedback becomes less desirable. Similarly, as the number of documents to be marked increases, then relevance feedback, again, becomes less desirable.

In the case of exploratory search, where the user wants to learn more about a domain, but is unsure of how to formulate a good query, then relevance feedback may become more attractive (assuming that they can do this at a low cost i.e. c_d). On the other hand, if the system provides support for querying, say by providing query suggestions, then the cost of querying could be lowered (though of course the user would have to pay the cost of shifting through the queries suggested[5]). Furthermore, since relevance feedback will typically find more of the same, if the user wants to explore other aspects of the topic, then it is likely that they will have to pose a new query in order to explore different aspects. Before we consider the case when querying and relevance feedback yield different amounts of benefit, we next consider the special case of finding similar.

5.2 Find Similar

Let's consider the find similar interface, where each document provides the option to find other similar documents. In this case, M always equals one, and the cost to find similar and mark becomes one and the same, i.e., the action to click the "find similar" button entails mentally marking the document (c_m) and then clicking the button c_c. Now the

[5] We leave this comparison between taking a query suggestion versus relevance feedback as an exercise for the reader.

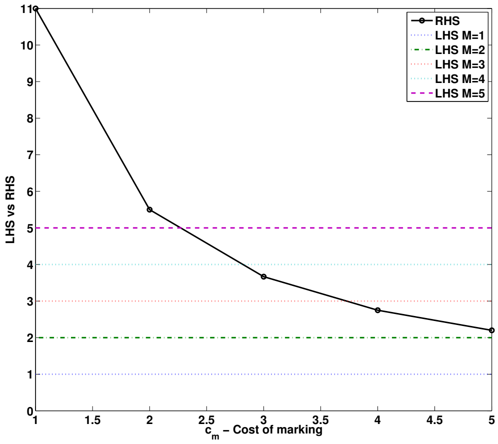


Figure 4: Inequality 14 showing that as the cost of marking increases (RHS), relevance feedback is less preferable as RHS < LHS which in turn violates the inequality.

decision to find similar is based on the inequality below:

$$c_q < c_d + c_m + c_c \tag{15}$$

such that if the cost of querying is cheaper than deciding, marking and clicking, the user will query. In a study of Excite query logs, it was shown that the find similar option was taken 1 in 20 times [52]. Nevertheless, modern search engines have more or less abandoned the feature. However, given this expression, the find similar option is probably cheaper, so why has it been abandoned? Here we posit that this is because the benefit function from each option is different (where the more expensive query yields more benefit). However, in other domains, like academic search and recommender systems, related articles or "more like this" options are still provided, though we were not able to find statistics on their usage. We posit that in academic search finding related articles is likely to be of more benefit to the less experienced researcher, who knows less about the field, and may incur higher querying costs [57]. Whereas the seasoned researcher who is more knowledgeable of the field is likely to receive less net benefit because they can formulate queries for less. On the other hand, in the context of product or movie recommendations, the cost and benefits are quite different. For example, Netflix provides users with a "more like this" option for movies and TV shows. In the context of finding a movie to watch, posing a query is more difficult because expressing how you feel or what you want from a movie is rather amorphous and anomalous [15]. This is likely to make the querying rather costly, whereas taking the "more like this" option is relatively cheaper and cognitively less taxing. So in this context "find similar" is likely to be much more useful - and as a result used more often.

5.3 Different Benefit Functions

In the models above, we have assumed that the benefit between different options is the same. However, in this context the subsequent query and the relevance feedback are likely to yield different amounts of benefit. Here we consider the implications of this. The amount of benefit a user receives from traversing the ranked list has been showing to increase but at a diminishing rate, i.e. as a user goes down the ranked list they accrue more benefit but less and less at each rank [42, 30]. This can be modeled using the following form: $b(N) = k.N^{\beta}$ [6, 8], and, when considered in the context of multiple queries, subsequent queries are assumed to be discounted. Thus, the benefit function takes the form $b(Q, N) = k.Q^{\alpha}.N^{\beta}$ and follows a similar form as the gain functions in [5, 7, 30, 31].

Using this model of benefit we can associate different amounts of benefit to the two options. The benefit from examining the initial N documents is $b_q = k.N^{\beta}$ and the benefit from the next N documents on the subsequent page is $b_{np} = k.(2.N)^{\beta} - k.N^{\beta}$. Given the expression $b(Q, N)$ from above, the benefit of the next query is based on the N documents returned, and the discount of issuing the next query. The discount for for the Qth query is $d(Q) = Q^{\alpha} - (Q-1)^{\alpha}$, i.e. for first query, $d(Q = 1) = 1$, for the second query, $d(Q = 2) = 2^{\alpha} - 1$. So the benefit from the second query would be $b_{nq} = k.d(Q = 2).N^{\beta}$. And finally the benefit from the relevance feedback would be: $b_{rf} = k.d(Q = 2).N^{\gamma}$. Since the user has already examined N and extracted the benefit b_q, the pool of relevant material has been reduced, and so we can assume that the benefit from the next query and the relevance feedback is reduced. If we consider the profit associated with these options, we arrive at:

$$\begin{aligned}
\pi_{np} &= b_q + b_{np} - (c_q + N.c_a + c_c + N.c_a) \\
&= k.(2.N)^{\beta} - (c_q + N.c_a + c_c + N.c_a)
\end{aligned}$$

$$\begin{aligned}
\pi_{nq} &= b_q + b_{nq} - (2.c_q + 2.N.c_a) \\
&= k.N^{\beta} + k.d(q).N^{\beta} - (2.c_q + 2.N.c_a)
\end{aligned}$$

$$\begin{aligned}
\pi_{rf} &= b_q + b_{rf} - (c_q + N.c_a + c_d \\
&\quad +M.c_m + c_c + N.c_a) \\
&= k.N^{\beta} + k.d(q).N^{\gamma} - (c_q + N.c_a + c_d + \\
&\quad +M.c_m + c_c + N.c_a)
\end{aligned}$$

A user would issue a subsequent query over performing relevance feedback if the following holds:

$$\begin{aligned}
\pi_{nq} &> \pi_{rf} \\
k.d(q).N^{\beta} - (c_q) &> +k.d(q).N^{\gamma} \\
&\quad -(c_d + M.c_m + c_c) \\
N^{\beta} - N^{\gamma} &> \frac{c_q - c_d - M.c_m - c_c}{k.d(q)}
\end{aligned} \tag{16}$$

This inequality shows that if the benefit that is received is greater from querying than relevance feedback, i.e. $\beta > \gamma$, then the Left Hand Side (LHS) will be positive. Unless the cost of relevance feedback is sufficiently low, such that the cost of querying is greater, then the user will re-query. However, note the impact of the discount $d(q)$, which is typically less than one because the user gets less benefit from subsequent interactions. Here it exaggerates the influence of the cost (positive or negative).

In Figure 5, we have plotted the LHS vs the RHS of Inequality 16, where we have assumed that the benefit function for the next query has β set to 0.5 (and the same costs as in the previous example shown in Figure 14). We have plotted varying levels of γ from 0.5 to 0.8. The figure shows that

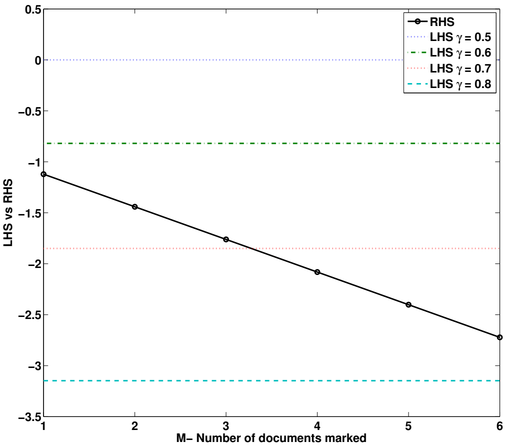


Figure 5: Inequality 16 where $c_c = 2$, $c_d = 2$, $k = 10$, $\alpha = 0.7$ and $\beta = 0.5$, showing that issuing a subsequent query is preferable when the LHS is above the black dotted line (RHS). The performance of the subsequent query would have to go up substantially before relevance feedback is worthwhile, i.e., $\gamma > 0.7$ 0.8 vs. $\beta = 0.5$.

the benefit from relevance feedback needs to be substantially greater than the benefit of the next query for relevance feedback to be useful to the user.

Summary

We have considered a number of different aspects relating to relevance feedback and how the number of documents marked and the benefit are likely to shape the user interactions. However, we have not considered other aspects such as the relationship between the amount (and type) of relevance feedback given [27, 36, 37]. For instance in [36], Keskustalo et al. show that as the number of feedback documents increases, performance increases but at a diminishing rate. Furthermore, we have approximated many of the costs and perhaps underestimated some. For example in [10] they show that the cost of marking documents as relevant is cognitively taxing. Thus, further empirical work is needed to better estimate the costs and the benefits of these interactions. If this is achieved then it will be possible to determine whether relevance feedback is worthwhile, and specifically to determine how much better the performance of the relevance feedback needs to be before it is worthwhile.

6. DISCUSSION AND FUTURE WORK

In this paper, we have developed a series of cost-benefit models for various decisions that arise when interacting with a search system/interface. Although the models we have created are rather simple, they provide interesting insights into the drivers and forces that are at play during the information seeking and retrieval process. Of course, these are not the only factors at play, and we acknowledge that these abstractions are limited. For each of the models depicted, there are obvious avenues of refinement where more detail and sophistication can be added to improve their realism. The purpose of this paper is to provide an overview of how cost-benefit analysis can be applied to assess the choices that we as designers present to users, and to hypothesize about how user behaviour will change in response to the costs and benefits. There are many more scenarios and decisions which can be explored and modeled in the future using such a framework.

Empirically estimating the parameters of these models, evaluating how well they fit, and predicting actual behaviour are obvious next steps. As such, this work provides a number of directions to explore as well as providing the basis for modeling and developing other models regarding other decisions and interactions. Creating such models is beneficial because before any experiments are conducted one can reason about how the costs and benefits are likely to influence behaviour and attempt to manipulate them accordingly. However, a number of open challenges and research questions are surfaced through this work (though not necessarily explicitly discussed): *(i)* how do we measure the costs and benefits, *(ii)* how does risk and uncertainty affect the decisions and expectations, *(iii)* why do users learn to adopt certain behaviours or others, *(iv)* at what point do certain behaviours become habitual and what are the costs of overcoming such habits *(v)* how can we encourage users to adopt better information seeking practices (and how do we define such practices as better?), and, *(vi)* how do we deal with multiple objectives and multiple cost functions.

7. REFERENCES

[1] E. Agapie, G. Golovchinsky, and P. Qvarfordt. Leading people to longer queries. In *Proc. of the SIGCHI*, pages 3019–3022, 2013.

[2] P. Arvola, J. Kekäläinen, and M. Junkkari. Expected reading effort in focused retrieval evaluation. *Inf. Retr.*, 13(5):460–484, 2010.

[3] L. Azzopardi. Query side evaluation: an empirical analysis of effectiveness and effort. In *Proc. of SIGIR*, pages 556–563, 2009.

[4] L. Azzopardi. Usage based effectiveness measures: Monitoring application performance in information retrieval. In *Proc. of CIKM*, pages 631–640, 2009.

[5] L. Azzopardi. The economics in interactive information retrieval. In *Proc. of SIGIR*, pages 15–24, 2011.

[6] L. Azzopardi. Economic models of search. In *Proc of ADCS*, ADCS '13, page 1, 2013.

[7] L. Azzopardi. Modelling interaction with economic models of search. In *Proc. of SIGIR*, pages 3–12, 2014.

[8] L. Azzopardi and G. Zuccon. An analysis of theories of search and search behavior. In *Proc. of ICTIR*, pages 81–90, 2015.

[9] L. Azzopardi and G. Zuccon. Two scrolls or one click: A cost model for browsing search results. In *Proc. of ECIR*, pages 696–702, 2016.

[10] J. Back and C. Oppenheim. A model of cognitive load for IR: implications for user relevance feedback interaction. *Information Research*, 2001.

[11] K. Balog. Task-completion engines: A vision with a plan. *CEUR-WS*, 1338, 2015.

[12] M. J. Bates. Information search tactics. *JASIS*, 30(4):205–214, 1979.

[13] N. J. Belkin. Salton award lecture: People, interacting with information. In *Proc. of SIGIR*, pages 1–2, 2015.

[14] N. J. Belkin, D. Kelly, G. Kim, J.-Y. Kim, H.-J. Lee, G. Muresan, M.-C. Tang, X.-J. Yuan, and C. Cool. Query length in interactive information retrieval. In *Proc. of SIGIR*, pages 205–212, 2003.

[15] N. J. Belkin, R. N. Oddy, and H. M. Brooks. ASK for information retrieval: Part I. Background and theory. *J. Doc.*, 38(2):61–71, 1982.

[16] U. Birchler and M. Butler. *Information Economics.* Routledge, 1st edition, 2007.

[17] B. Carterette. System effectiveness, user models, and user utility: A conceptual framework for investigation. In *Proc. of SIGIR*, pages 903–912, 2011.

[18] O. Chapelle, D. Metlzer, Y. Zhang, and P. Grinspan. Expected reciprocal rank for graded relevance. In *Proc. of CIKM*, pages 621–630, 2009.

[19] A. Chuklin, I. Markov, and M. d. Rijke. Click models for web search. *Synthesis Lectures on Information Concepts, Retrieval, and Services*, 7(3):1–115, 2015.

[20] M. Cole, J. Liu, N. Belkin, R. Bierig, J. Gwizdka, C. Liu, J. Zhang, and X. Zhang. Usefulness as the criterion for evaluation of interactive information retrieval. In *Proc. of HCIR*, pages 1–4, 2009.

[21] M. D. Cooper. A cost model for evaluating information retrieval systems. *JASIS*, 23(5):306–312, 1972.

[22] W. S. Cooper. On selecting a measure of retrieval effectiveness. *JASIS*, 24(2):87–100, 1973.

[23] N. Craswell, O. Zoeter, M. Taylor, and B. Ramsey. An experimental comparison of click position-bias models. In *Proc. of WSDM*, pages 87–94, 2008.

[24] R. Cummins, M. Lalmas, and C. O'Riordan. The limits of retrieval effectiveness. In *Proc. of ECIR*, pages 277–282, 2011.

[25] N. Fuhr. A probability ranking principle for interactive information retrieval. *Inf. Retr.*, 11(3):251–265, June 2008.

[26] J. Gwizdka. Distribution of cognitive load in web search. *JASIST*, 61(11):2167–2187, 2010.

[27] D. Harman. Relevance feedback revisited. In *Proc. of SIGIR*, pages 1–10, 1992.

[28] D. Harman. Is the cranfield paradigm outdated? In *Proc. of SIGIR*, pages 1–1, 2010.

[29] P. Ingwersen and K. Järvelin. *The Turn: Integration of Information Seeking and Retrieval in Context.* Springer-Verlag New York, Inc., 2005.

[30] K. Järvelin and J. Kekäläinen. Cumulated gain-based evaluation of ir techniques. *ACM TOIS*, 20(4):422–446, 2002.

[31] K. Järvelin, S. L. Price, L. M. L. Delcambre, and M. L. Nielsen. Discounted cumulated gain based evaluation of multiple-query ir sessions. In *Proc. of ECIR*, pages 4–15, 2008.

[32] A. Kashyap, V. Hristidis, and M. Petropoulos. Facetor: cost-driven exploration of faceted query results. In *Proc. of CIKM*, pages 719–728, 2010.

[33] D. Kelly. Contours and Convergence: Past, Present and Future (interactive) IR Research Practice - KSJ Keynote at ECIR. In *Proc. of ECIR*, 2013.

[34] D. Kelly, V. D. Dollu, and X. Fu. The loquacious user: A document-independent source of terms for query expansion. In *Proc. of SIGIR*, pages 457–464, 2005.

[35] D. Kelly and X. Fu. Eliciting better information need descriptions from users of information search systems. *IP&M*, 43(1):30–46, 2007.

[36] H. Keskustalo, K. Järvelin, and A. Pirkola. The effects of relevance feedback quality and quantity in interactive relevance feedback: A simulation based on user modeling. In *Proc. of ECIR*, pages 191–204, 2006.

[37] H. Keskustalo, K. Järvelin, and A. Pirkola. Evaluating the effectiveness of relevance feedback based on a user simulation model: effects of a user scenario on cumulated gain value. *Inf. Retr.*, 11(3):209–228, 2008.

[38] S. Mizzaro. Relevance: The whole history. *JASIS*, 48(9):810–832, 1997.

[39] A. Moffat, P. Bailey, F. Scholer, and P. Thomas. INST: An Adaptive Metric for Information Retrieval Evaluation. In *Proc. of ADCS*, 2015.

[40] A. Moffat, F. Scholer, and P. Thomas. Models and Metrics: IR Evaluation As a User Process. In *Proc. of ADCS*, pages 47–54, 2012.

[41] A. Moffat, P. Thomas, and F. Scholer. Users versus models: What observation tells us about effectiveness metrics. In *Proc. of CIKM*, pages 659–668, 2013.

[42] A. Moffat and J. Zobel. Rank-biased precision for measurement of retrieval effectiveness. *ACM TOIS*, 27(1):2, 2008.

[43] P. Pirolli. *Information Foraging Theory: Adaptive Interaction with Information.* Oxford University Press, Inc., 2009.

[44] S. E. Robertson. The probability ranking principle in ir. *J. Doc.*, 33(4):294–304, 1977.

[45] D. M. Russell, M. J. Stefik, P. Pirolli, and S. K. Card. The cost structure of sensemaking. In *Proc. of INTERACT/SIGCHI*, pages 269–276, 1993.

[46] T. Sakai and Z. Dou. Summaries, ranked retrieval and sessions: A unified framework for information access evaluation. In *Proc. of SIGIR*, pages 473–482, 2013.

[47] M. D. Smucker. A Plan for Making Information Retrieval Evaluation Synonymous with Human Performance Prediction. In *Proc. of SIGIR Workshop on Future of Evaluation*, 2009.

[48] M. D. Smucker. Towards timed predictions of human performance for interactive information retrieval evaluation. In *Proc. of HCIR*, page 3, 2009.

[49] M. D. Smucker and C. L. Clarke. The fault, dear researchers, is not in cranfield, but in our metrics, that they are unrealistic. In *Proc. of EuroHCIR*, pages 11–12, 2012.

[50] M. D. Smucker and C. L. Clarke. Time-based calibration of effectiveness measures. In *Proc. of SIGIR*, pages 95–104, 2012.

[51] M. D. Smucker and C. L. Clarke. Modeling Optimal Switching Behavior. In *Proc. of CHIIR*, pages 317–320, 2016.

[52] A. Spink, B. J. Jansen, and H. Cenk Ozmultu. Use of query reformulation and relevance feedback by excite users. *Internet Research*, 10(4):317–328, 2000.

[53] G. J. Stigler. The economics of information. *The Journal of Political Economy*, 69(3):213–225, 1961.

[54] H. R. Varian. Economics and search. *SIGIR Forum*, 33(1):1–5, 1999.

[55] R. Villa and M. Halvey. Is Relevance Hard Work? Evaluating the Effort of Making Relevant Assessments. In *Proc. of SIGIR*, pages 765–768, 2013.

[56] E. M. Voorhees, D. K. Harman, et al. *TREC: Experiment and evaluation in information retrieval.* MIT press Cambridge, 2005.

[57] R. W. White, S. T. Dumais, and J. Teevan. Characterizing the influence of domain expertise on web search behavior. In *Proc. of WSDM*, pages 132–141, 2009.

[58] E. Yilmaz, E. Kanoulas, M. Verma, B. Carterette, N. Craswell, and R. Mehrotra. Overview of the TREC 2015 Tasks Track. In *Proc. of TREC*, 2015.

[59] E. Yilmaz, M. Shokouhi, N. Craswell, and S. Robertson. Expected Browsing Utility for Web Search Evaluation. In *Proc. of CIKM*, pages 1561–1564, 2010.

[60] E. Yilmaz, M. Verma, N. Craswell, F. Radlinski, and P. Bailey. Relevance and effort: an analysis of document utility. In *Proc. of CIKM*, pages 91–100, 2014.

[61] Y. Zhang and C. Zhai. Information retrieval as card playing: A formal model for optimizing interactive retrieval interface. In *Proc. of SIGIR*, pages 685–694, 2015.

[62] Y. Zhang, J. Zhang, M. Lease, and J. Gwizdka. Multidimensional relevance modeling via psychometrics and crowdsourcing. In *Proc. of SIGIR*, pages 435–444, 2014.

[63] G. Zuccon. Understandability biased evaluation for information retrieval. In *Proc. of ECIR*, pages 280–292. Springer, 2016.