



Information Retrieval as Card Playing: A Formal Model for Optimizing Interactive Retrieval Interface

Yinan Zhang

Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, IL 61801
yzhng103@illinois.edu

Chengxiang Zhai

Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, IL 61801
czhai@illinois.edu

ABSTRACT

We propose a novel formal model for optimizing interactive information retrieval interfaces. To model interactive retrieval in a general way, we frame the task of an interactive retrieval system as to choose a sequence of interface cards to present to the user. At each interaction lap, the system's goal is to choose an interface card that can maximize the expected gain of relevant information for the user while minimizing the effort of the user with consideration of the user's action model and any desired constraints on the interface card. We show that such a formal interface card model can not only cover the Probability Ranking Principle for Interactive Information Retrieval as a special case by making multiple simplification assumptions, but also be used to derive a novel formal interface model for adaptively optimizing navigational interfaces in a retrieval system. Experimental results show that the proposed model is effective in automatically generating adaptive navigational interfaces, which outperform the baseline pre-designed static interfaces.

1. INTRODUCTION

Developing formal models for information retrieval (IR) has always been an important fundamental challenge. For example, the Probability Ranking Principle (PRP) [18] proposed more than three decades ago has laid out a solid foundation and provided a theoretical justification for framing the retrieval task as a ranking problem, leading to the development of many effective retrieval functions for ranking documents that are used in current search engines (e.g., [20, 19, 17, 10, 1, 21, 6]). Despite the great success of PRP, however, it is also known that it is based on two problematic assumptions, i.e., sequential browsing and independent relevance (utility) of documents, which are generally not true in practice. As a result, e.g., the traditional retrieval models developed based on PRP cannot handle redundancy among documents directly (and must rely on post-processing of search results), an immediate consequence of the independence assumption of document relevance. The sequential browsing

assumption implies limitations on the interface also, and in particular ignores the actions that a user can take when interacting with an interface displaying search results (e.g., faceted browsing).

Recognizing these limitations and attempting to generalize the PRP for interactive IR, Fuhr[8] has recently proposed a novel formal framework for optimizing interactive retrieval and derived a PRP for interactive IR (IIR-PRP) where a user's effort and benefit are captured when optimizing the ranking of documents. This work effectively addressed the independence assumption made in the PRP and provides a theoretical foundation for optimizing document ranking when a user is assumed to interactively browse a list of search results. Unfortunately, it has not addressed the sequential browsing assumption, which remains an assumption made for optimizing ranking in interactive retrieval. In this work, we relax this assumption and propose a more general formal model than IIR-PRP for optimizing interactive retrieval.

The sequential browsing assumption touches a much larger problem of how to model a user's reactions to a retrieval result interface, which further depends on what the interface looks like, raising the interesting question "how can we formally model the problem of interface design for an interactive IR system?" Interestingly, in contrast with a large body of work on formal methods for optimizing ranking, there has been little work on formal methods for optimizing the interface of a system, despite that the dynamic and interactive nature of information seeking process has long been recognized and studied from information science perspective (see, e.g., [4, 11]). While optimizing ranking is clearly very important for optimizing a retrieval system, when we consider optimizing an *interactive* retrieval system, we must also optimize the interface part of the system so as to optimize the *whole* system, which is the goal of our study.

The study of interface optimization is especially important in the current era of ever faster technology advancement, leading to the emergence of smart phones and various kinds of wearable devices with very small screens, which generally require a different interface than the popular interface designed for desktops. For example, while showing a document list on a relatively large screen is popular and appropriate, it may not be appropriate to do so on a very small screen where an interface with navigational tags might be more useful as it enables a user to more efficiently navigate into the relevant information. Even if we consider the current interface of a Web search engine such as Google or Bing on a large screen, which typically shows a list of fixed number of snippets on each page, there are still many interesting questions

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.

SIGIR'15, August 09 - 13, 2015, Santiago, Chile.

© 2015 ACM. ISBN 978-1-4503-3621-5/15/08 ...\$15.00.

DOI: <http://dx.doi.org/10.1145/2766462.2767761>.

related to the optimization of the interface. For example, how many snippets should we display on each page? The commonly used number, 10, is not necessarily always the best choice. Also, what about shortening some snippets to make room for more results or vice versa? These questions have been tackled by Human-Computer Interaction (HCI) researchers with many empirical findings. Unfortunately, it is still unclear how we can leverage these findings to build an intelligent IR system that can automatically optimize its interaction interface adaptively both to the screen size and to the user's information need.

In this paper, we study the novel problem of *automatic* interface optimization formally, and propose a new general formal model, called the *Interface Card* model, for optimizing interactive retrieval interface. The basic idea behind this model is to view an interactive retrieval process as a process of the retrieval system playing a cooperative card game¹ with the user in the following way: at each interaction lap, facing a current retrieval *context*, the system would choose an optimal "interface card" to present to the user. The user can then perform any action from a set of possible actions associated with the interface card presented. Depending on the user's action on the interface card (e.g., selecting a particular facet value), the system would then transition to a new context, and have to choose another (generally new) interface card to show to the user. The game would continue in such a way until the user decides to stop (either due to satisfaction of the information need or abandoning the search). At each interaction lap, the system's goal is to choose an interface card that can maximize the expected gain of relevant information for the user while minimizing the effort of the user with consideration of a user action model and any desired constraints on the interface card.

We show that such a general formal interface card model can not only cover IIR-PRP as a special case by making multiple simplification assumptions, including the sequential browsing assumption, but also be used to derive a novel formal interface model for adaptively optimizing navigational interfaces in a retrieval system by assuming that an interface card is composed of one or more information blocks to support interactive navigation and a user's action is mainly to select one of the presented blocks. The derived model enables, for the first time, automatic generation of optimal navigational interfaces that can be adaptive to screen sizes and user interactions. Experimental results show that the proposed model is effective in automatically generating adaptive navigational interfaces, which outperform the baseline pre-designed static interfaces.

2. RELATED WORK

The majority work on formal models for IR has been based on the Probability Ranking Principle (PRP) [18] and all attempt to optimize a ranking function defined on a query-document pair; they include all kinds of traditional retrieval models such as vector space models [20], classic probabilistic models [19], language models [17, 10, 22], divergence from randomness [1], inference networks [21], axiomatic approaches [6], and recent extensions in the direction of learning to rank [7, 14]. These works generally do not model user interactions, thus providing no guidance for interface design.

The PRP for interactive IR (IIR-PRP) [8] generalizes the PRP to optimize ranking in an interactive IR setting, where a number of important concepts for modeling user interaction from the perspective of decision making were introduced, including situation, effort and benefit, and an optimal ordering principle is derived for ranking items when a user is assumed to sequentially browse the list. Our model shares a similar high-level goal with the IIR-PRP in that both attempt to establish a formal model for interactive retrieval, but it is more general than the IIR-PRP, which can be shown as a special case of our model under a set of simplification assumptions. Due to its generality, our model does not make the sequential browsing assumption and can be directly used to optimize the interaction interface; as a result, our model can suggest interfaces that would dynamically adapt to the assumed screen sizes, which cannot be achieved in any existing work on formal models.

The dynamic and interactive nature of information seeking process has long been recognized and studied from information science perspective (see, e.g., [4, 11]). Our work can be regarded as an attempt to formalize some of the theoretical arguments in these literature with an operational mathematical model that can be used for building an intelligent IR system with an adaptive interface for navigation.

Our model of dynamic information need is related to the ostensive model (OM) [5], which provides a framework for modeling the evolution of information needs over time. Our model is sufficiently general to allow us to refine it with the ostensive model or any other model of evolving information needs. Our proposed framework enables any such model to be adopted for optimizing navigational interface.

The model we derived for optimizing navigational interface uses an objective function to maximize the difference between a user's benefit and cost for finding a relevant information item. Such a decision criterion is related to some recent works that have explored economic models for IR (e.g., [2]). Furthermore, optimizing the ranking of documents with consideration of user actions has also been studied in the context of feedback to optimize the session-level utility [12] and using a POMDP framework in [15], where a dual-agent POMDP was proposed to model both user actions and system actions. However, none of these studies has proposed a model to optimize navigational interface, a primary goal of this paper.

Optimizing search engine interface has been extensively studied in the Human-Computer Interaction community (see, e.g., the survey in the book [9]), including designing and evaluating faceted browsing systems and coming up with various ad hoc ways to optimize such systems (e.g. [3] and [13]). However, no existing work can optimize a navigational interface with an explicitly defined objective function, which our model attempts to achieve.

3. INTERFACE CARD MODEL

In general, any interaction between a user and an interactive information retrieval system can be partitioned into a series of interaction laps, in each of which the user issues an action and the system then reacts to the user's action by selecting an optimized interface instance to show to the user. For example, in a traditional search engine, the first interaction lap consists of the user issuing a query and the search engine responding with 10 most relevant items as the first result page. After this interaction lap, the user may issue a

¹http://en.wikipedia.org/wiki/Card_game

second action by either clicking an item or “next page,” and the interface reacts by displaying a second interface instance optimized for the perceived user action.

The interaction laps may be defined in various levels of granularity and the set of user actions would change accordingly. The previous example can be regarded as modeling the interaction at the page level. If, however, the 10 search results could not simultaneously fit into the screen of the interface as in the case of searching with a smart phone, then the interaction can be modeled at a finer granularity - the current screen shown to the user, and the user actions would additionally include scrolling up/down, to which the interface reacts by “sliding” the screen up/down by one position. In this scenario, when the user scrolls down, the interface in theory could have a chance to decide again according to the user’s action about the item to be shown next, which may be different from the one originally ranked at this position. Such “drilling down” of the interaction granularity could continue, if we consider a user’s every eye movement as an action and the interface may dynamically change the displayed content accordingly, assuming the availability of an eye-tracker device².

How do we model an arbitrary interactive retrieval system formally at any given interaction granularity level? To address this question, we propose to view any user-interface interaction as a card game, in which the “interface player” determines the optimal card to play in each lap, and we present a novel interface card model to formally model the interactive retrieval task. Unlike a real card game where players maximize their own benefits, however, the interface card model assumes a cooperative game in which the “interface player” always maximizes the user’s benefit by taking into consideration the user’s current action, the interaction history, the reward and cost of the user’s next possible actions and any constraints posed onto the card the “interface player” plays at the current lap. We now formally introduce the model by first defining all these components and then the core mathematical optimization problem.

Definition 3.1 (Lap). A *lap* is the interaction unit between the user and the interface in which the user issues an action and the interface then reacts by generating an optimized interface instance: $t = 1, 2, \dots$

The laps serve as the timestamps for the user-system interactions and will always be shown as superscripts.

Definition 3.2 (Card). A *card* is an interface instance generated by the interface system in reaction to the user action in each lap: q^t .

The notion of card generalizes a wide range of interface instances including a result page or a screen of a partial result page in a search engine, a question in a conversational retrieval interface the system uses to clarify the user’s information need, etc.

Definition 3.3 (Constraint). The *constraint* is a possible set of restrictions a card needs to satisfy in a lap, and for simplicity is assumed to have the form of a single constraint function: $f_c^t(q^t) \leq 0$.

² In theory, we could go even deeper: imagine that we might some day have sensors installed for everyone to track every neuron excitement in their brain and consider that as an interaction unit.

The constraint is typically associated with the design and restrictions of the interface. For instance, a result page of a traditional search engine may display at most 10 items at a time. If screens are considered as a finer unit for the interaction, when the user scrolls down, all the items on the next screen except the bottom one are restricted to be the ones sliding from the previous screen. In more complicated interface designs like faceted browsing interfaces, there might be panels of facet values and items regulating how much space they could respectively occupy.

In many cases, the constraint could not be captured within a single constraint function. The notion of the single constraint function is only meant for the purpose of notational simplicity, and more complicated forms of constraint do not change the model in any fundamental way.

Definition 3.4 (Action). An *action* is a move the user chooses to take next from a set of possible moves that may depend on the current card: $a^{t+1} \in \mathcal{A}(q^t)$.

The interaction is initiated either by the user or the system, and we allow both situations in this model. Most of the time, the user is the initiator, and the interaction starts with a^1 (followed by the interface playing the first card q^1 , then the user issuing the second action a^2 , etc.). For example, when a user queries a search engine, a^1 would be the very first query the user enters. Alternatively, if the search engine attempts to display a possibly personalized search homepage to each user and at each time, then we could define a^1 to be the user’s action of entering the website. In both cases, the first card the interface plays is designated to be the first interface instance that needs to be optimized according to the user’s action. Typically, we are not interested in the set of possible actions for the very first user action a^1 because a^1 is always regarded as given to the model and there isn’t any uncertainty in it.

There are occasionally situations where the interface system is the initiator of the interaction, e.g. if a smart phone is set to alert the user whenever some interesting news events happen by popping up a screen of news event snippets. In such a scenario, we set q^1 to be this first screen and set a^1 to be a “null” action.

For the sake of simplicity, from now on we always assume that the action set $\mathcal{A}(q^t)$ is either finite or countably infinite, so that we could use the summation sign “ \sum ” for summing over all possible actions in a particular lap. In cases where the action set is not countable (e.g. if a touch-screen smart phone measures how much force the user uses when touching the screen), we could replace the “ \sum ” signs with the “ \int ” sign and the core model is not affected in any fundamental way.

Definition 3.5 (Context). The *context* is all the information the interface system has accumulated till a particular lap about the user for estimating the user’s choice on the next card: c^t . Such information includes (a) a priori information about the user, i , if any, (b) interactions in all previous laps, if any, between the user and the interface (i.e. the sequence of previous actions issued by the user and previous cards played by the interface), and (c) the action the user just issued in the current lap. The context is expressed as a vector starting with $c^1 = (i, a^1)$ and iteratively updated by $c^{t+1} = (c^t, q^t, a^{t+1})$.

Typically, the a priori information about the user may capture any prior belief about the user, e.g. any available personalization information.

Definition 3.6 (Action Model). The *action model* specifies the system’s estimated probability distribution of the user’s actions in the next lap given the current card and under the current context: $p(a^{t+1}|c^t, q^t)$.

Here we are assuming a probabilistic model for user actions, which provides a general solid framework for formally modeling most real world scenarios.

Definition 3.7 (Reward). The *reward* is the system’s estimated expected benefit to the user for issuing an action given the current card and under the current context: $r(a^{t+1}|c^t, q^t)$.

The reward may capture the short-term benefit to the user from a relevant item, as well as any long-term benefit, e.g., if the action serves to navigate the user to a new information subspace (as in the case of answering a clarification question in a conversational retrieval system or clicking a facet value in a faceted browsing system). The reward may depend on future laps if the system decides to perform the estimation computation in such a way, but here for notational simplicity, we only put c^t and q^t into $r(a^{t+1}|c^t, q^t)$ and hide any possible dependency of the reward on future laps in the reward function r .

Definition 3.8 (Cost). The *cost* is the system’s estimated expected effort the user spends for issuing an action given the current card and under the current context: $s(a^{t+1}|c^t, q^t)$.

For example, the cost function typically captures any possible effort the user needs to take for scanning through a result page of a search engine, for the decision-making process to determine whether to click or skip a particular item, etc.

Definition 3.9 (Surplus). The *surplus* is the difference between the reward and the cost to the user for issuing an action given the current card and under the current context: $u(a^{t+1}|c^t, q^t) = r(a^{t+1}|c^t, q^t) - s(a^{t+1}|c^t, q^t)$.

Here we borrow the concept of surplus from economics studies to designate the net benefit to the user for issuing an action. From the user’s perspective, they would typically tend to choose actions leading to higher surplus, and there have been well established economics theories, e.g. the discrete choice model [16], for modeling such behaviours. In this study, however, we do not go deeper in such directions and stop at the level of the action model in formalizing the user’s behavior from the interface system’s perspective.

With all the necessary components defined, we formally introduce the core mathematical optimization problem:

Definition 3.10 (Interface Card Optimization). In each lap t , the interface system should play a card q^t that maximizes the expected surplus u^t given the current context and under the current constraint, where the expectation is taken with respect to the user action model:

$$\begin{aligned} \underset{q^t}{\text{maximize}} \quad & E(u^t|c^t, q^t) \\ &= \sum_{a^{t+1} \in \mathcal{A}(q^t)} p(a^{t+1}|c^t, q^t) u(a^{t+1}|c^t, q^t) \quad (1) \\ \text{subject to} \quad & f_c^t(q^t) \leq 0 \end{aligned}$$

A possible source of confusion is that there are in total two levels of expectations in this formalism: the inner one being encapsulated within the notion of surplus that deals

with the uncertainty in the reward and cost for individual actions (recall that the reward and cost are both defined as expectations), and the outer one that deals with the uncertainty in the user’s decision on which action to issue. Such an encapsulation generally holds in reality and lays down a convenient formalism framework for multiple ways of instantiating the interface card model as we will further discuss in the following sections.

The proposed interface card model is very general and does not make the “sequential browsing” assumption that is underlying all other existing theoretical IR models. Rather, we only adopts a “sequential interaction” scheme where the interaction laps between the user and the interface system take place sequentially, which is a much broader notion than “sequential browsing”. The reason is that “sequential browsing” is *positional* - it assumes that the user follows a strict sequential order when scanning the positions on a list. In contrast, “sequential interaction” is *temporal* - it only assumes that the interaction flows in a sequential manner, and if we think more deeply, as interaction always flows in the same direction as time passes, the “sequential interaction” notion is essentially stating that time passes uni-directionally, which is universally and always true. In such a sense, we are effectively adopting the broadest possible “assumption” underlying any human-computer interaction process. We will show in the following section that we can make simplification assumptions to reduce our “sequential interaction” scheme to the traditional “sequential browsing” scheme and derive the IIR-PRP model proposed in [8].

4. PLAIN CARD

As our first instantiation of the interface card model, we will show that the interface card model can cover the IIR-PRP model as a special case under a set of simplification assumptions including particularly the sequential browsing assumption. Consider the problem setting of a generalized interactive information retrieval (IIR) system as introduced in [8], where the system’s task is to present a sequential list of binary choices to the user, and the system needs to determine the optimal order of the list so as to maximize the user’s expected benefit. Such a problem formulation generalizes a wide range of IIR tasks. (Please refer to [8] for more in-depth explanations.)

In order to instantiate the IIR-PRP model, we first need to incorporate the “sequential browsing” assumption into our model, and we do so by reducing “sequential interaction” to “sequential browsing” using the following pair of assumptions, which is essentially stating the “sequential browsing” assumption in the language of our model:

Assumption 4.1 (Plain Card). Each card is defined to be a choice in the ranked list. The choices are sequentially denoted by $e^t, t = 1, 2, \dots$, and we define $q^t = e^t$.

Assumption 4.2 (Binary Action). There are two possible user actions in each lap: $\mathcal{A}(e^t) = \{a_0^{t+1}, a_1^{t+1}\}$, where a_0^{t+1} and a_1^{t+1} respectively represent the actions of accepting e^t and rejecting e^t (to examine the next choice e^{t+1}).

The interface card optimization problem is now equivalent to determining which choice to place on each position of the list. We are implicitly modeling the interaction at the level of the user’s eye movement: imagine that the interface system is accompanied with an eye-tracking device that could

sense the user's eye movement and automatically scroll the page whenever it detects that the user intends to skip to the next choice; in such a way we get rid of the necessity of the scrolling action and the action of clicking "next page."

Since a card is simply assumed to be a choice on the list, there is no interesting constraint defined on the interface. Further, we adopt the independence assumption in [8] and assume that the probability of the user accepting a choice is independent of the choices they have rejected, so that the action model does not depend on any previous cards or user actions until an accept action takes place. We also follow [8] to focus on the optimization problem before the user's first accept action (the optimization problem afterwards is regarded as a new round of optimization), so the context is simply collapsed to the a priori information about the user (if any). We thus omit the notion of context in the following writing, and define the shorthand notation $p(e^t)$ for specifying the action model: $p(e^t) = p(a_0^{t+1}|e^t) = 1 - p(a_1^{t+1}|e^t)$. (Please refer to [8] for the notion of "situations" to understand the details and rationales of these assumptions.)

Assumption 4.3 (Rejection reward). The reward for a reject action is the expected surplus in the next lap:

$$r(a_1^{t+1}|e^t) = E(u^{t+1}|e^{t+1}) \quad (2)$$

Here we choose to explicitly model the dependency of the reward in the current lap on the future laps, so in order to optimize the first card, we would need to simultaneously optimize all the following cards (i.e. all choices in the list).

We define another shorthand notation $r(e^t)$ for the reward of accepting choice e^t : $r(e^t) = r(a_0^{t+1}|e^t)$. In [8], this reward is further decomposed into the expectation of two cases - (a) the accept action is right and (b) the accept action is wrong. We do not go further along this direction; the main line of the derivation would not be affected in any fundamental way.

Assumption 4.4 (Decision cost). The costs for the accept and reject actions are the same in each lap, which equal the amount of effort the user spends to examine the current choice for deciding whether they should accept or reject it:

$$s(a_0^{t+1}|e^t) = s(a_1^{t+1}|e^t) = s(e^t) \quad (3)$$

Now, with all the necessary assumptions introduced, we plug Equation (2) and (3) into Equation (1), extract the common decision cost out of the summation, and come to:

$$E(u^t|e^t) = -s(e^t) + p(e^t)r(e^t) + (1-p(e^t))E(u^{t+1}|e^{t+1}) \quad (4)$$

We recursively apply Equation (4) starting from the first lap and obtain:

$$E(u^1|e) = \sum_{t=1}^{\infty} \left(\prod_{j=1}^{t-1} (1-p(e^j)) \right) (-s(e^t) + p(e^t)r(e^t)) \quad (5)$$

where e denotes the vector of all choices on the list. (The summation could alternatively be defined as a finite one if we assume a finite list but the derivation stays the same.)

Since the surplus captures all long-term benefits (via its reward part), u^1 in Equation (5) captures the surplus of the entire list. We explicitly wrote out the dependency of u^1 on all future cards (i.e. all following choices) by expanding " $E(u^1|e^1)$ " to " $E(u^1|e)$ " for the purpose of clarity.

Finally, from Equation (5), we follow the approach used in [8] by considering optimizing the order of each consecutive choice pair and obtain the IIR-PRP model: assuming that

all $p(e^t)$ are greater than 0, $E(u^1|e)$ is maximized when the choices are ranked in decreasing order of:

$$\rho(e^t) \stackrel{\text{def}}{=} r(e^t) - \frac{s(e^t)}{p(e^t)} \quad (6)$$

We have thus mathematically demonstrated that the interface card model is a generalization of the IIR-PRP model. As a remark, although we stated earlier that we would typically need an eye-tracking device to model user interactions at the granularity level of eye movement, it turns out that there's no need for the eye-tracking device under the set of assumptions in this example: the ranking could be pre-generated from the " ρ " values of the choices and never needs to dynamically change according to the user's eye movement.

5. NAVIGATIONAL CARD

We now come to the second instantiation example of the interface card model, where we demonstrate that without assuming "sequential browsing" and given the availability of a richer set of navigational elements, the interface card model can lead to very powerful optimization results that could not be achieved by any existing formal frameworks.

We go back to the classic IR setting where the user is looking for some items using the search engine, but we consider a new popular set of real world scenarios where we have some navigational elements to show on the interface in addition to the items themselves, which we collectively refer to as *tags*. For example, when we are searching for books in an online library catalog, we may use subject headings to quickly narrow down the set of books we need to examine. In a news browsing website, as another example, the news keywords could serve as navigational tags following which the user is able to identify an interesting news article much faster than they could if they are only given an article list, even a well optimized one. In general, these navigational tags themselves are not what the user is looking for, but they are linked to (possibly overlapping) subsets of items into which the user could quickly zoom by selecting the tags.

One key challenge in this setting is that, since the user is now faced with both a list of tags and a list of items, there is no longer a single list of choices which is assumed by [8], and the sequential browsing assumption no longer holds. As a result, many interesting questions regarding how to optimally generate a navigational interface in such cases cannot be answered in a theoretically rigorous way: e.g., (a) how many tags and items should we present in each interface instance, and how do we optimally partition the interface into the tag panel and the item panel? (b) should we allocate a larger proportion of the screen space to tags in smaller screens, and if so, how do we make such adjustment optimally? (c) along the interaction process, do we start by showing tags and then switch to the items when the system becomes more certain about the user's information need, and if so, what would be the optimal time for the switch?

We address all these questions in a novel principled approach by establishing another instantiation of the interface card model that only assumes a "sequential interaction" scheme without going further towards "sequential browsing". We first present a set of assumptions and notations and then demonstrate the effectiveness of our approach.

Definition 5.1 (Block). A *block* b is a display unit on a card representing either a tag or an item that could be selected

by the user. A block representing a tag / item is referred to as a *tag / item block*.

Assumption 5.1 (Navigational Card). Each card is a subset of blocks along with their presentation strategy.

The presentation strategy of the blocks on the card is a generalized notion that typically incorporates any ordering and/or panel layout of the blocks. Note that the user may or may not follow any order in examining the blocks (as is assumed in the traditional sequential browsing scheme).

Assumption 5.2 (Selection Action). In each lap t , the user could either select a block on the card q^t or select “next card”: $\mathcal{A}(q^t) = q^t \cup \{a_N^{t+1}\}$, where a_N^{t+1} denotes the “next card” action.

From now on, we will directly use q^t to designate the set of blocks on q^t , and we use e to represent items (which we used to represent choices in the previous section). The “next card” action is a generalization of many real world user actions to skip everything shown in the current interaction lap and see more options, e.g. clicking “next page”, scrolling down, shifting eye focus one position down, etc.

Definition 5.2 (Preference). The *preference* is the system’s estimated probability distribution characterizing the user’s interest in each item e . The system relies on the context to progressively update the preference along the interaction process and we designate the preference at lap t by $p(e|c^t)$.

Definition 5.3 (Item Action Model). The *item action model* for item e is the user’s action model on the current card given their interest in item e : $p(a^{t+1}|e, q^t)$.

In practice, the item action model serves as the main linkage between our interface model and the item-tag relations. The intuition is that a user interested in a particular item would generally be more likely to select a tag related to the item. Of course, if the item block corresponding to the item itself is displayed on the card, the user would almost always select the item block rather than any tag block. But if neither the item block nor any related tag block is displayed, the user would most likely issue the “next card” action. We will come back to this in more detail later.

The original action model could now be written as the expected item action model, where the expectation is taken with respect to the preference:

$$\begin{aligned} p(a^{t+1}|c^t, q^t) &= \sum_e p(e|c^t, q^t) p(a^{t+1}|e, c^t, q^t) \\ &= \sum_e p(e|c^t) p(a^{t+1}|e, q^t) \end{aligned} \quad (7)$$

where we assume that (a) the preference is independent of the next card given the context and (b) the item action model is independent of the context given the item of interest to the user, both of which are very reasonable in general.

The rationale underlying such a decomposition of the original action model into two probabilistic models, the preference and the item action model, is two folds. Firstly, by dividing the action model at the item level, we allow for more flexibility in user modeling efforts in practice. Secondly, the decomposition naturally leads to a principled way of updating the preference via Bayes’ theorem:

$$\begin{aligned} p(e|c^{t+1}) &= p(e|c^t, q^t, a^{t+1}) \\ &= \frac{p(e|c^t, q^t) p(a^{t+1}|e, c^t, q^t)}{p(a^{t+1}|c^t, q^t)} = \frac{p(e|c^t) p(a^{t+1}|e, q^t)}{p(a^{t+1}|c^t, q^t)} \end{aligned} \quad (8)$$

where $p(a^{t+1}|c^t, q^t)$ comes from Equation (7) and we adopted the same two assumptions we made in deriving Equation (7).

To make the optimization problem more tractable, we make the following assumption about the reward function to prevent the optimization from depending on future laps:

Assumption 5.3 (Information Gain Reward). The reward of an action is the information gain in the preference distribution estimated in the next lap over the current lap:

$$\begin{aligned} r(a^{t+1}|c^t, q^t) &= \text{InfoGain}(\mathbf{p}(e|c^{t+1}), \mathbf{p}(e|c^t)) \\ &= H(\mathbf{p}(e|c^t)) - H(\mathbf{p}(e|c^{t+1})) \end{aligned} \quad (9)$$

where $p(e|c^{t+1})$ comes from (8) and H is the information or entropy function: $H(\mathbf{p}) = -\sum_p p \log p$.

Intuitively, at a high level, the interactive retrieval process resembles an encoding of the user preference: the lower the entropy of the preference, the more the system knows about the user’s information need, and the easier it would be for the system to help the user find some interesting items. Therefore, the amount of reduction in the entropy of the preference becomes a natural choice for approximating the reward. (We could have explicitly written out the dependency of the reward on future laps just as what we did in the previous section, but it would make the computation overly complicated and intractable.)

Now, we come to address the problem that the user may not always follow the sequential browsing scheme while examining a card due to the fact that the card may often be more complicated than a simple list. Ideally, we want a “browsing model” to characterize the browsing behavior of users, which may be obtained through user studies or estimated based on interaction logs, but as an initial step along such a generalization direction, we choose to focus on interfaces with a *relatively small capacity* with respect to humans’ attention, and we make the following two assumptions:

Assumption 5.4 (Capacity Constraint). The only constraint on the blocks shown on a card is that the total space the blocks occupy does not exceed the capacity of the card:

$$f_c^t(q^t) = \sum_{b \in q^t} w(b) - 1 \quad (10)$$

where $w(b)$ is the space block b occupies relative to the card.

Assumption 5.5 (Uniform Cost). The cost is assumed to be uniform across any action the user issues on a card:

$$s(a^{t+1}|c^t, q^t) = s, \forall a^{t+1} \in \mathcal{A}(q^t) \quad (11)$$

A key implication behind the capacity constraint is that, since it serves as the only constraint on the cards, we do not further impose any requirement regarding (a) what proportion of the card should be allocated to tag blocks and item blocks, (b) how many tag blocks and item blocks should be shown on the card, and (c) whether the card should be completely devoted to tag blocks or item blocks. Essentially, there is no presentation strategy - we are setting a completely “flexible” interface layout that our interface card model could freely optimize. (We implicitly assumed that the blocks are all of regular shapes, so that a block could always be packed into the card as long as the amount of space left on the card is no less than the space the block occupies.)

Meanwhile, the uniform cost assumption also has a key implication: we assume the user could browse the blocks

on the card in *any* order and, due to the relatively small capacity of the card, it always takes the user a constant, very small amount of attention to browse the blocks and make a decision on what action to issue next no matter what order the user follows in browsing the blocks. In such a way we are effectively relaxing the sequential browsing assumption.

With all the necessary assumptions and definitions laid down, we plug Equation (9), (10) and (11) into Equation (1). It is easily observed that two terms could be extracted out of the summation: (a) the entropy of the current preference, $H(\mathbf{p}(e|c^t))$, and (b) the constant cost, s , and since these two terms do not involve q^t , we could simply remove them from the objective function without affecting the optimization result. Eventually, the final optimization problem for our navigational card becomes:

$$\begin{aligned} & \underset{q^t}{\text{minimize}} && \sum_{a^{t+1} \in \mathcal{A}(q^t)} p(a^{t+1}|c^t, q^t) H(\mathbf{p}(e|c^{t+1})) \\ & \text{subject to} && \sum_{b \in q^t} w(b) - 1 \leq 0 \end{aligned} \quad (12)$$

where $p(a^{t+1}|c^t, q^t)$ and $p(e|c^{t+1})$ respectively come from Equation (7) and (8).

Now, we continue with analytical and real user experiments to demonstrate that Equation (12) leads to very interesting and powerful interface optimization results not achievable by any other existing method in principled ways.

Before we go into real computations, we first need to have (a) an initial preference model as the starting point for the series of context updates along the interaction, and (b) a working item action model. Since this study is not meant to be a user modeling study, from now on we simply assume a flat initial preference distribution:

Assumption 5.6 (Uniform Initial Preference). The initial user preference is uniform across a set of n items, i.e. $p(e_i|c^1) = 1/n, \forall i = 1, 2, \dots, n$.

In reality, the system usually has a better estimate of the user preference in the initial lap. For example, the a priori information may suggest to the system that the user is generally more interested in certain categories of items. Additionally, in cases of search engines, the system may have estimates of the probability of relevance for each item with respect to the user's query, so that the probability of the user's interest in each item along the ranked list returned by the system should be decreasing. The assumption of uniform initial preference we make here is for the sake of computational convenience; it is solely meant to reduce some distracting details not relevant to the core model.

Definition 5.4 (Item-Tag Map). An *item-tag map* is a weighted bipartite network composed of (a) item nodes and tag nodes respectively corresponding to the set of all items and tags, (b) weighted edges between item and tag nodes if the item and tag they represent are related, with the edge weight quantifying the strength of their relation. A *uniform item-tag map* is an item-tag map in which all edges are of uniform weights. For nomenclature purpose, we say that a tag *covers* an item if there's an edge linking their corresponding nodes in the item-tag map.

Definition 5.5 (Simple Item Action Model). Under the *simple item action model*, given the user's interest in item e and the set of blocks shown on card q^t , the user would issue an action based on the following three rules:

1. If the item block corresponding to e , b_e , is on the card, the user always selects it: if $b_e \in q^t$, then $p(b_e|e, q^t) = 1$; $p(b|e, q^t) = 0, \forall b \neq b_e$; and $p(a_N^{t+1}|e, q^t) = 0$.
2. Otherwise, if the card contains at least one tag block covering e , the user will either select one of these tag block(s) or "next card" with probabilities proportional to the corresponding edge weight(s) in the item-tag map and a predefined parameter ε , respectively:

$$\begin{cases} p(b|e, q^t) = \frac{v(e, b)}{\sum_{b' \in q^t} v(e, b') + \varepsilon} & (13) \\ p(a_N^{t+1}|e, q^t) = \frac{\varepsilon}{\sum_{b' \in q^t} v(e, b') + \varepsilon} & (14) \end{cases}$$

where $v(e, b)$ denotes the weight of the edge between the nodes in the item-tag map representing e and b .

3. Otherwise, the user will always select "next card", i.e. $p(a_N^{t+1}|e, q^t) = 1$ and $p(b|e, q^t) = 0, \forall b \in q^t$.

The *simple uniform item action model* denotes the simple item action model on top of a uniform item-tag map, and the *perfect uniform item action model* is the simple uniform item action model with ε set to 0.

We implicitly assumed in the second rule that, in cases of "competing" blocks, i.e. multiple blocks covering the same item simultaneously appearing on the card, the relative tendencies of the user selecting these blocks are kept constant, and equal the relative weights of the corresponding edges in the item-tag map. Such a simplification may not always hold in reality, since the relative tendencies of block selections might depend on the user, the lap, and other blocks on the card; however, it is in general a valid approximation and could greatly simplify the computation.

The user might sometimes accidentally miss a related tag and select "next card", which could be captured using the ε parameter, though we assume that the user would never miss the items they are interested in.

5.1 Analytical Experiments

We apply our result for optimizing navigational cards in some simple example scenarios to analytically demonstrate the effectiveness of the interface card model in generating optimal interactive interfaces. Although it might be possible to develop alternative ad hoc approaches that could result in the very same analytical solutions we derive here, our approach adopts a principled way that is solidly rooted in a theoretical IR model, which no existing approaches could achieve. In this section, we mainly focus on mathematically deriving the optimal conditions for the blocks on the card, and in particular the tag blocks (since the cases for item blocks are generally simpler); we leave the demonstration of our model's effectiveness in automatically generating optimal interface layout in reality to the user study experiment.

To make the presentation cleaner, we omit the lap and context notions in all places: we assume that all the discussion here is about the optimization in the initial lap, and we adopt the *uniform initial preference assumption*. We also adopt the *perfect uniform item action model* for the sake of mathematical convenience. Furthermore, in order to better focus on the most crucial line of the calculation without worrying about any trivial technical details, we assume a "perfect world" of tag navigation:

Assumption 5.1.1 (Complete Tag Set). There always exists some tag that precisely covers any given item subset.

As a consequence, we could entirely focus on deriving the mathematical conditions for the optimal tag(s) we should pick onto the card without worrying about whether such tag(s) actually exist or not in reality.

5.1.1 One Tag Per Card

In this example, we assume that the card only has space for a single tag block:

Assumption 5.1.1.1 (One Tag Per Card). $w(b) = 1, \forall b$.

The optimization question now becomes: what is the optimal number of items the picked tag should cover? If the user is interested in some item covered by the picked tag, then the user will select the tag; otherwise, the user will select “next card”. Based on Equation (8), in the first case, the preference is updated to narrow down towards the subset of items covered by the picked tag, and in the second case, the preference narrows down towards the subset of items not covered by the picked tag. Suppose the picked tag covers x items, $x \in \{1, 2, \dots, n\}$. We plug the entropies of the two updated preference distributions into Equation (12) and after some straightforward algebraic simplifications, the optimization problem becomes:

$$\underset{x}{\text{minimize}} \quad \frac{1}{n} (x \log x + (n - x) \log (n - x)) \quad (15)$$

We consider Equation (15) as a function of x and extend its domain to real numbers in $[1, n]$. By taking the derivative, we conclude that the minimization solution is:

$$x = \frac{n}{2} \quad (16)$$

Therefore, selecting a tag block covering around half of the items leads to an optimal card. This result shows the model tends to create a balanced partition of the item preference distribution, which coincides with our intuition.

5.1.2 Two Tags Per Card

In this example, we “expand” the card and assume it has space for two tag blocks:

Assumption 5.1.2.1 (Two Tags Per Card). $w(b) = 1/2, \forall b$.

Now, the optimization problem becomes two-folds: (a) how many items should each of the two picked tags cover? and (b) how many items should the two tags’ coverages overlap? To answer these two questions, let the number of items covered by the two tags respectively be x and y , $x, y \in \{1, 2, \dots, n\}$, and let the number of common items covered by the two tags be t , $t \in \{0, 1, \dots, n\}$, $t \leq x$, $t \leq y$, $x + y - t \leq n$. A crucial difference between this example and the last one is that if the user is interested in some item in the two tags’ overlap, the user may select either one of them with equal probabilities, which affects the calculation of the action model and the updated preferences. After some tedious algebraic simplifications, the optimization problem in Equation (12) eventually comes to:

$$\underset{x, y, t}{\text{minimize}} \quad \frac{1}{n} \left(t \log 2 + (x - \frac{t}{2}) \log (x - \frac{t}{2}) + (y - \frac{t}{2}) \log (y - \frac{t}{2}) + (n - x - y + t) \log (n - x - y + t) \right) \quad (17)$$

Again, we consider Equation (17) to be a function of x , y and t , and we relax the integer constraint. By taking the partial derivatives, without going much into the technical details, we conclude that the final minimization solution is:

$$x = y = \frac{n}{3}, t = 0 \quad (18)$$

There are two implications from this result: (a) it reassures that the model would favor a balanced partition of the preference distribution, and (b) it additionally suggests that the model would minimize the partitions’ overlaps, coinciding again with our intuition.

5.2 User Study Experiments

To further demonstrate the effectiveness of our interface card model, we built real prototype interface systems based on the navigational card model to show that our model could lead to automatic interface layout adjustment, which no existing method could achieve in a principled way, and we validate the superiority of our automatic interface layout results by comparing them with baseline pre-designed static interfaces in user studies.

The prototype interfaces were built on top of the set of most popular news articles and their associated keywords returned from the New York Times Most Popular API³, in which the articles and the keywords respectively correspond to the items and the tags in our model. We developed two interfaces with different sizes, a medium sized one⁴ and a very small one⁵. We assumed in our implementation that the user would follow the simple uniform item action model, and we heuristically set $\varepsilon = 0.5$.

Though the optimization problem in Equation (12) was shown to have closed form solutions in our two analytical experiments, it is generally difficult to solve for real world scenarios. For building the prototype interfaces, we implemented a straightforward randomized algorithm to tackle the problem. The algorithm heuristically generates multiple candidate cards at each lap, and chooses the one minimizing Equation (12). To obtain each candidate card, the algorithm picks blocks to add to the card one at a time that (a) do not violate the capacity constraint and (b) have a minimal overlap with all blocks that are already picked onto the card (as in line with what we observed in the analytical experiments). Since algorithmic designs are not the focus of this paper, we don’t go into all the technical details of our algorithm due to space limitations. However, we point out that the algorithm is very efficient - its time complexity is linear with respect to the input size, i.e. the total number of items and the total number of tags of all items.

5.2.1 Sample Cards

In Figure 1, the left and top-right images are screenshots of an initial interface layout on the medium sized screen and the very small screen, respectively, as automatically determined by the interface card model based on the popular news articles in New York Times and their keywords some time in late January 2015. We see that the algorithm intelligently decided to include only tag blocks on the small

³<http://developer.nytimes.com/>

⁴<http://timan102.cs.illinois.edu/yzhng103/fomalhaut/m-times-navigation.php>

⁵<http://timan102.cs.illinois.edu/yzhng103/fomalhaut/s-times-navigation.php>

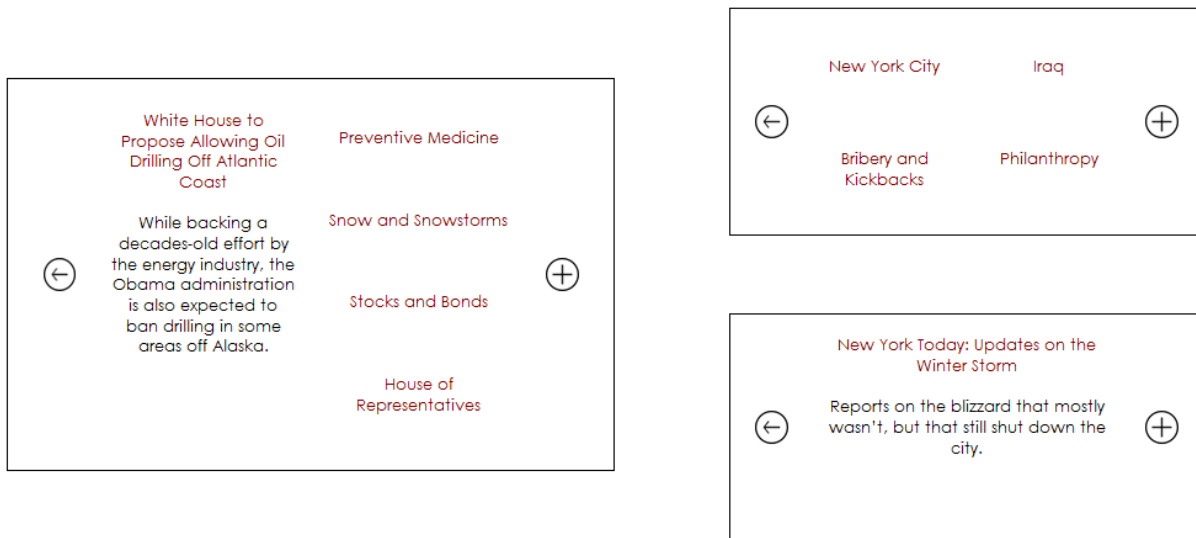


Figure 1: Screen shots of example cards.

screen, but include both tag blocks and item blocks on the medium sized screen. Such a decision makes sense since unless we are relatively sure about what item the user is looking for (which unlikely happens in the initial interaction lap), it would likely be a waste of screen space if specific items are displayed; in contrast, tags are potentially more useful. The bottom-right screenshot in Figure 1 shows an automatic layout adjustment in response to the user’s action of selecting the “New York City” tag in the top-right interface. Despite its limited capacity, the screen is entirely filled with an item block because the estimated user preference is narrowed down to only a few items and the system determined that directly showing an item is more beneficial. These results demonstrate that our model can effectively achieve automatic layout adjustment according to both the screen size and the user interaction.

5.2.2 User Studies

We built two baseline interfaces for comparison purpose: one is for the medium sized screen, where we put a separate static tag panel on the right side of the main item panel⁶; for the very small screen, we have either a tag panel or an item panel on the screen at each time, and put a switch button to allow users to switch between the two panels⁷. These two baselines represent popular layouts seen on many mobile interfaces with medium and small screen sizes⁸. We conducted real user experiments on Amazon Mechanical Turk⁹ to compare the two baselines with our interfaces (on both medium and small screens) for a task of navigating into the most interesting article that was pre-identified by the user, and we measure the number of interaction laps for the users to reach their target article and compute p-values based on a one-side Wilcoxon sign-ranked test. We also varied the size of the item set to see its impact. The results in Table 1 show

⁶<http://timan102.cs.illinois.edu/yzhng103/fomalhaut/m-times-duplex.php>

⁷<http://timan102.cs.illinois.edu/yzhng103/fomalhaut/s-times-simplex.php>

⁸<http://itunes.apple.com/app/amazon-mobile/id297606951>

⁹<http://www.mturk.com/>

that our interface outperforms the baseline interface in all the cases, though with varying significance levels (p-values less than 0.05 are highlighted). It is clearly observed that the superiority of our interface over the baseline interface is higher when the screen is smaller, and is also higher when there are more items.

Table 1: Significance levels of comparison results.

Card size	Item set size	Valid sample size	P-value
Small	20	19	0.004753
Small	50	23	0.0003546
Medium	20	18	0.09183
Medium	50	20	0.01097

We also asked the users survey questions for their opinions on the two types of interfaces to obtain some qualitative comparisons, and a majority of the users indicated that our interface was both quicker and easier to use. For example, one user wrote: “*the interface seemed to intuitively know what article I wanted from just selecting two keywords.*” Many users noted that the baseline interface felt familiar and thus was straightforward to use, but they also pointed out that it did not take much effort to learn how to use our interface: “*at first I was unsure of how I would find my target article but followed my instincts and found it right away.*” The difference in navigational efficiency between the two interfaces was more exaggerated in the very small screen, even in the search space of 20 articles. Since the baseline interface layout does not automatically switch between the keywords and the articles, a lot of users were not able to take full advantage of the keywords and simply ended up being scrolling through the entire article list: “*it seemed like I had to search longer and scroll through almost every article to find the one I wanted.*” In the medium sized screen, even though the baseline interface shows both the tag panel and the article panel, quite a few users noticed the ability of our interface to dynamically change the layout and applauded it: “*I liked that the interface gave such large amount of results when you clicked on a tag.*”

6. CONCLUSIONS AND FUTURE WORK

We proposed a novel general formal model for optimizing interactive information retrieval interfaces by viewing the interactive retrieval process as a process of a system playing a cooperative card game with a user with the goal of minimizing the user's effort and maximizing the user's gain of relevant information. At each interaction lap, the system would choose an optimal interface card (i.e., an interface instance) to present to the user based on the current context, a model of the user's possible actions on the interface, and a model of the user's gain and effort. The user can then choose an action to take on the prompted interface, which would lead to a new context for the system to choose the next optimal interface card.

We showed that this general interface card model can cover the PRP for Interactive IR as a special case under a set of simplification assumptions, particularly the sequential browsing assumption (thus also easily cover the classic PRP as a more special case). We further derived a novel model for optimizing navigational interfaces that are adaptive to both the screen size and the user's information need. Experimental results with real users show that the proposed model can effectively optimize a navigational interface and is significantly better than baseline static interfaces that are heuristically customized for different screen sizes.

The interface card model is very general and can model interactions at any meaningful granularity level as long as we can define meaningful interface cards and user actions; thus we can model both "micro" interactions at the level of actions such as scrolling up/down inside a page, and "macro" interactions at the level of page navigation. The new model opens up many interesting new directions in optimizing the whole interactive retrieval system through incorporating machine learning and HCI study results. Specifically, the proposed formal framework naturally fits a wide variety of state-of-the-art machine learning techniques, and can easily adopt learning to rank methods [7, 14] and models such as the ostensive model [5] for evolving information needs to further improve the estimate of user preferences. With abundant interaction log data that can be recorded automatically, such learning techniques would provide more accurate estimate of multiple components in the framework. Also, findings from HCI research could be directly incorporated into the constraint part in our optimization problem, providing our model guidance in certain domains that currently could not be formalized in a straightforward way, e.g. learnability concerns, error tolerance, etc. With the general trend in IR pushing researchers to focus more on the interface part and formalize interactive IR, we hope this paper can stimulate alternative and more advanced formalisms for interactive IR to be developed in the coming years (e.g., those in line of economic models for IR [2] and POMDP [15]).

7. REFERENCES

- [1] G. Amati and C. J. Van Rijsbergen. Probabilistic models of information retrieval based on measuring the divergence from randomness. *ACM Trans. Inf. Syst.*, 20(4):357–389, Oct. 2002.
- [2] L. Azzopardi. Modelling interaction with economic models of search. In *SIGIR '14*, pages 3–12, 2014.
- [3] S. Basu Roy, H. Wang, G. Das, U. Nambiar, and M. Mohania. Minimum-effort driven dynamic faceted search in structured databases. In *CIKM '08*, pages 13–22, 2008.
- [4] N. J. Belkin, R. N. Oddy, and H. M. Brooks. Ask for information retrieval: Part I: Background and theory. *Journal of Documentation*, 38(2):61–71, 1982.
- [5] I. Campbell and K. V. Rijsbergen. The ostensive model of developing information needs. In *CoLIS 2*, pages 251–268, 1996.
- [6] H. Fang and C. Zhai. An exploration of axiomatic approaches to information retrieval. In *SIGIR '05*, pages 480–487, 2005.
- [7] N. Fuhr. Optimum polynomial retrieval functions based on the probability ranking principle. *ACM Trans. Inf. Syst.*, 7(3):183–204, July 1989.
- [8] N. Fuhr. A probability ranking principle for interactive information retrieval. *Information Retrieval*, 11(3):251–265, 2008.
- [9] M. A. Hearst. *Search User Interfaces*. Cambridge University Press, 2009.
- [10] D. Hiemstra and W. Kraaij. Twenty-one at TREC-7: ad-hoc and cross-language track. In *TREC-7*, volume 500-242, pages 227–238, 1999.
- [11] P. Ingwersen. Cognitive perspectives of information retrieval interaction: elements of a cognitive IR theory. *Journal of Documentation*, 52:3–50, 1996.
- [12] X. Jin, M. Sloan, and J. Wang. Interactive exploratory search for multi page search results. In *WWW '13*, pages 655–666, 2013.
- [13] A. Kashyap, V. Hristidis, and M. Petropoulos. FACeTOR: Cost-driven exploration of faceted query results. In *CIKM '10*, pages 719–728, 2010.
- [14] T.-Y. Liu. Learning to rank for information retrieval. *Found. Trends Inf. Retr.*, 3(3):225–331, Mar. 2009.
- [15] J. Luo, S. Zhang, and H. Yang. Win-win search: Dual-agent stochastic game in session search. In *SIGIR '14*, pages 587–596, 2014.
- [16] D. McFadden and K. Train. Mixed MNL models for discrete response. *Journal of Applied Econometrics*, 15(5):447–470, 2000.
- [17] J. M. Ponte and W. B. Croft. A language modeling approach to information retrieval. In *SIGIR '98*, pages 275–281, 1998.
- [18] S. E. Robertson. The probability ranking principle in IR. *Journal of Documentation*, 33(4):294–304, 1977.
- [19] S. E. Robertson and K. S. Jones. Relevance weighting of search terms. *Journal of the American Society for Information Science*, 27(3):129–146, 1976.
- [20] G. Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. *Commun. ACM*, 18(11):613–620, Nov. 1975.
- [21] H. Turtle and W. B. Croft. Evaluation of an inference network-based retrieval model. *ACM Trans. Inf. Syst.*, 9(3):187–222, July 1991.
- [22] C. Zhai. Statistical language models for information retrieval a critical review. *Found. Trends Inf. Retr.*, 2(3):137–213, Mar. 2008.