Information Retrieval Evaluation as Search Simulation: A General Formal Framework for IR Evaluation

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

Xueqing Liu
Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, IL 61801
xliu93@illinois.edu

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

ABSTRACT

While the Cranfield evaluation methodology based on test collections has been very useful for evaluating simple IR systems that return a ranked list of documents, it has significant limitations when applied to search systems with interface features going beyond a ranked list, and sophisticated interactive IR systems in general. In this paper, we propose a general formal framework for evaluating IR systems based on search session simulation that can be used to perform reproducible experiments for evaluating any IR system, including interactive systems and systems with sophisticated interfaces. We show that the traditional Cranfield evaluation method can be regarded as a special instantiation of the proposed framework where the simulated search session is a user sequentially browsing the presented search results. By examining a number of existing evaluation metrics in the proposed framework, we reveal the exact assumptions they have made implicitly about the simulated users and discuss possible ways to improve these metrics. We further show that the proposed framework enables us to evaluate a set of tag-based search interfaces, a generalization of faceted browsing interfaces, producing results consistent with real user experiments and revealing interesting findings about effectiveness of the interfaces for different types of users.

CCS CONCEPTS

• **Information systems → Evaluation of retrieval results**;

KEYWORDS

IR evaluation; User simulation; Interface card

ACM Reference format:
Yinan Zhang, Xueqing Liu, and ChengXiang Zhai. 2017. Information Retrieval Evaluation as Search Simulation: A General Formal Framework for IR Evaluation. In *Proceedings of ICTIR '17, Amsterdam, Netherlands, October 01-04, 2017,* 8 pages.
https://doi.org/.1145/3121050.3121070

ICTIR '17, October 01-04, 2017, Amsterdam, Netherlands

ACM ISBN 978-1-4503-4490-6/17/10…$15.00
https://doi.org/.1145/3121050.3121070

1 INTRODUCTION

Information Retrieval (IR) is an empirically defined task in the sense that there is no way to mathematically prove one IR system is better than another, and the question of which IR system is the best can only be answered based on how well the system can help users finish a task. Thus, how to appropriately evaluate an information retrieval (IR) system has always been one of the most important research questions in IR [10, 14, 22]. So far, the dominant methodology for evaluating an IR system has been the Cranfield evaluation methodology proposed in 1960s [24]. The basic idea is to build a test collection that consists of a sample of queries, a sample of documents, and a set of relevance judgments (indicating which documents are relevant/non-relevant to which queries). An IR system can then be evaluated using such a test collection as follows. First, we run the system on the test collection to generate retrieval results for each of the test queries. We then quantitatively evaluate the system results for each query with various measures (such as precision and recall) based on the relevance judgments. Measures on all the queries can be aggregated to quantify the performance of a system on the whole set of queries. Such a methodology has also been widely used for evaluating many other empirical tasks, including particularly machine learning tasks.

A key benefit of using the Cranfield evaluation methodology is that the test collection, once built, would be reusable as many times as we want to, which enables repeatedly using the *same* test collection to compare different systems or examine the effectiveness of each component in a complicated system. Such reusability is key to ensure reproducibility of IR experiments. The Cranfield evaluation methodology has played a crucial role in advancing IR technologies, and the reusability of the created test collections has enabled the development of many effective retrieval algorithms that are used in many modern search engine applications today.

Unfortunately, the Cranfield evaluation methodology, in its current form, can only be used for evaluating simple IR systems that return a ranked list of documents, and would encounter significant difficulty when applied to more sophisticated IR systems which have become increasingly popular due to the advancement in technologies for human-computer interaction. In particular, it is hard to use it to evaluate an interactive IR system where we need to assess the overall performance of a system over an entire interactive search session and compare two different search interfaces that may go beyond a ranked list of documents (e.g., an interface with features such as query suggestion or faceted browsing); such sophisticated IR systems have so far been evaluated primarily through controlled user studies [14] or a proxy of such a user study experiment by

performing search log analysis [9]. However, the experiment results obtained in such a way would be hard to reproduce due to the difficulty in completely controlling the users.

In this paper, we propose a general formal framework for evaluating IR systems based on search session simulation that can be used to evaluate *any* IR system with *reproducible experiments*, including systems with sophisticated retrieval interfaces. The key idea is to build "user simulators," which are software programs that can simulate how a user would interact with a search engine (interface) when trying to finish a task. With a set of such user-task simulators, we can then test each IR system by having the system interact with the simulators. The interaction sequence of system responses and user actions can then be used to compute various quantitative measures of the system based on how effective the system has helped the (simulated) user finish a task.

We show that such a simulation-based evaluation framework is, in fact, a generalization of the traditional Cranfield evaluation method to enable reproducible experiments to evaluate or compare sophisticated IR systems. The current ranked list evaluation method can be derived quite naturally as a specific instantiation of the framework, where the simulated search session is a user sequentially browsing the presented search results.

One immediate benefit of the proposed framework is that it enables us to examine any existing evaluation metric *formally* from the perspective of user simulation, which further helps reveal the exact assumptions a metric has made (often implicitly) about the simulated users. The analysis also helps provide an interpretation of any metric from a user's perspective. We formally study several widely used measures, Precision, Recall, and Average Precision (AP), and reveal the assumptions made by these measures.

A more important benefit of the framework is that it would enable us to evaluate more complicated IR systems that are hard to evaluate with existing evaluation methods. As a case study to pursue this benefit, we build search session simulators to evaluate a set of tag-based search interfaces, a generalization of faceted browsing interfaces, with validation of our proposed framework from real user experiments and interesting findings about effectiveness of the interfaces for different types of users.

2 RELATED WORK

Evaluation has always been a central research topic in IR; the three surveys by Sanderson [22], Kelly [14], and Harman [10] have covered most progress in IR evaluation research, though many newer papers on the topic have been published since those three surveys were written, notably the axiomatic approaches to IR evaluation [4], and applications of statistical analysis techniques. Cranfield test-collection evaluation methodology proposed a long time ago [24] remains the dominant evaluation method in IR for comparing different retrieval algorithms today, and the ranking performance is often assessed using measures such as Precision, Recall, MAP and/or NDCG. It was demonstrated in [20] that MAP could be derived under certain user behavior assumptions, which was one of the initial attempts to interpret IR evaluation metrics from the perspective of user behavior models. Additional evaluation measures have been proposed and used for evaluating various IR tasks, such as α-NDCG[8], Rank-based Precision [18], Expected Reciprocal Rank [6], and time-based measures [23]. A very recent study [27] proposed a novel Bejeweled Player Model for evaluating IR systems, which could not only cover many existing metrics as special case but also provide a more principled and refined model for users' stopping behaviors when scanning along a ranked list. However, while these approaches work well for evaluating retrieval results in the form of a ranked list, it is unclear how it can be applied to evaluate an interactive retrieval system associated with more diversified interface elements and user behaviors. The proposed simulation-based evaluation framework breaks this limitation and generalizes the previous evaluation method to provide a principled way to evaluate any interactive system.

User studies are also often conducted to evaluate an IR system, including both small-scale controlled studies and larger-scale user studies using A/B test. While such an evaluation method involves real users and accurately reflects the utility of a retrieval system in application settings, it has a serious drawback (as compared with Cranfield evaluation method) in not being reproducible. A main point of our paper is that the only way to enable reproducible experiments with interactive IR systems is through user simulation. The framework can be regarded as both a generalization of the test collection approach to enable evaluation of interactive IR systems, and an "artificial" way to perform interactive user studies.

A previous work [5] has already made an attempt to evaluate session search by doing simulation; our work is a step forward to propose a more general framework. Indeed, it appears that we have no choice but to use such a simulation framework if we want to perform reproducible experiments to evaluate an interactive retrieval system with sophisticated interfaces since this appears to be the only way to control the user. Our work is also related to the recent work by the Glasgow group on user simulation (see, e.g., the simulation toolkit [16]), but our goal of doing simulation is different, i.e., it is to evaluate an arbitrary IR system.

There have been extensive studies on evaluating ranking systems' performance using simulated user [5, 15, 26]. Traditional IR studies have long been focusing on modeling users' click behaviors [7] and relevance feedback [13, 15]. Recent studies have gone beyond click models to simulate other aspects of user behavior, including simulating user queries [26] (often based on language models [2, 12, 26]), simulating a user's stopping behavior [17, 25] based on gain/cost ratio [19], and query reformulation [5]. A common weakness of these studies is that they are mostly based on random sampling instead of learning from real user behavior [5]. As a result, it remains a challenge how to fairly compare different algorithms using results generated by these simulators. However, they can be leveraged to build an accurate simulator for use in the proposed evaluation framework.

The line of work on economic models for IR [1, 3] studied user interactions with an interactive IR system from the perspective of economic factors, e.g. reward/cost. Our proposed framework also models user reward/cost factors but focuses on evaluating the IR system.

3 SEARCH SIMULATION FRAMEWORK

In this section, We formally characterize our proposed search simulation framework for interactive IR evaluation. We first explicitly

define the basic components in the framework at the level of the whole interaction.

Definition 3.1 (System, User, Task and Interaction Sequence). In any interaction involving two parties issuing actions to each other in turn, we define the *(interactive) system* S to be the party to be evaluated, the *user* U to be the other party, the *task* T to be the user's information need, and the *interaction sequence* I to be the whole process of the interaction.

A user may have different information need, or task, when using a system, and the user with a specific task may result in different interaction sequences due to the randomness of the user actions and the system responses.

Definition 3.2 (Simulator). A *simulator* is a (synthetic) user with a task, created for the purpose of evaluating a system.

In general, a system's performance over an interaction sequence can be measured in two dimensions from a user's perspective: reward and cost:

Definition 3.3 (Interaction / Simulator Reward and Cost). For an interaction sequence I between a user U with task T and an interactive system S, the *interaction reward* $R(I,T,U,S)$ and the *interaction cost* $C(I,T,U,S)$ respectively represent the overall amount of reward and cost the user gets from the whole interaction. For a simulator simulating a user U with task T and an interactive system S, the *simulator reward* $R(T,U,S)$ and the *simulator cost* $C(T,U,S)$ respectively represent the expected interaction reward and cost over all possible interaction sequences: $R(T,U,S) = E(R(I,T,U,S))$ and $C(T,U,S) = E(C(I,T,U,S))$, where the expectation is taken with respect to the distribution of all possible interaction sequences, $\boldsymbol{p}(I|T,U,S)$.

Note that $\boldsymbol{p}(I|T,U,S)$ would be entirely concentrated on a single interaction sequence if the interaction is deterministic.

The simulator reward $R(T,U,S)$ and cost $C(T,U,S)$ provide a complete and interpretable characterization of the utility of system S to user U with task T: $C(T,U,S)$ measures the effort made by a user, while $R(T,U,S)$ gives the reward that a user would receive for the effort. We chose to maintain reward and cost as two separate measures because the desired trade-off between them is inevitably application-specific, thus it should be treated as an external application of our framework. Moreover, we can easily further define the average utility and cost of a system over a group of simulators to obtain an overall reward and cost, or first combine reward and cost for each individual simulator and then compute the average over a group of simulators; these again would be better treated as applications of the framework. We will see some interesting examples in Section 4.

The formalism established above serves as a high-level framework for assessing interactive retrieval systems in general on the whole interaction level, in particular by evaluating the reward and cost of a task oriented user when interacting with the system through an interaction sequence. To assess the reward and cost at a finer level, we must define the interaction sequence in more detail. To this end, we follow the Interface Card Model [28] and partition the interaction between a user and an interactive IR system into a series of interaction laps:

Definition 3.4 (Lap, Action and Interface Card). The *lap* $t = 1, 2, \ldots$ is the time unit of the interaction between a user and a system in which the user and the system each acts once in turn. In each lap t, the user first issues an *action* a^t, and the system then reacts by generating an *interface card* q^t. The *stopping action* a^t_B is a special action the user could issue in each lap which ends the interaction.

It is often the case that there is certain level of intrinsic randomness in the user action and the system's interface cards. In this work, we focus more on the user side, and we will later adopt a user action model describing the probabilistic distribution of the user actions at each lap.

When different users interact with the same system, or even when the same user interacts with the same system at different times, the user might tend to issue different actions, depending on e.g. the user's habits, information need (task), and any past interactions between the user and the system. We characterize such user side information by user state (which we adopt from [29]):

Definition 3.5 (User State). At each lap t, the *user state* z^t denotes the collection of all the information that as a whole is sufficient to determine how likely the user issues each possible action given any interface card the system issues. The user state starts from the initial user state z^1, which depends on the user U and the task T and follows an *initial user state distribution* $\boldsymbol{p}_I(z^1)$. The user state then transitions across laps probabilistically via the *user state transition function* $\boldsymbol{p}_\mathcal{T}(z^{t+1}|z^t, a^t, q^t)$.

Intuitively, the user state in many cases could be in the form of a multi-dimensional vector where each element denotes some aspect of the status of the interaction process, e.g. the stage of the interaction process, the remaining information need, etc. Based on the user state, we formalize the action model of the user:

Definition 3.6 (User Action Model). The *user action model* specifies the probability distribution of the user issuing each possible user action in a given lap, where the probabilities are conditioned on the user state and the interface card: $p(a^{t+1}|z^t, q^t)$.

We can now define the interaction sequence on a finer level:

Definition 3.7 (Interface Card Interaction Sequence). For an interaction process between a system S and a user U with task T, the interaction sequence I is composed of the sequence of user states, the user actions and the interface cards in the whole interaction: $I = ((z^1, a^1, q^1), (z^2, a^2, q^2), \ldots, (z^n, a^n, q^n))$, where n denotes the total number of laps in the interaction. We define I^t to be the partial interaction sequence from lap 1 to lap t, $1 \le t \le n$. ($I = I^n$.)

The interaction reward and cost can now be refined as follows:

Definition 3.8 (Cumulative / Lap Reward and Cost). For user U with task T, system S and interaction sequence I, the *cumulative reward* and *cumulative cost* at lap t are respectively the total reward and cost the user obtains by the end of lap t: $R^t(I,T,U,S) = R(I^t,T,U,S)$, and $C^t(I,T,U,S) = C(I^t,T,U,S)$. The *lap reward* and *lap cost* are respectively the difference of cumulative reward and cost between consecutive laps: $r^t(I,T,U,S) = R^t(I,T,U,S) - R^{t-1}(I,T,U,S)$, and $c^t(I,T,U,S) = C^t(I,T,U,S) - C^{t-1}(I,T,U,S)$. (We define $R^0(I,T,U,S) = C^0(I,T,U,S) = 0$.)

The notion of cumulative reward and cost provides the basis for the simulator to track the reward and cost measures progressively along the interaction process. The lap reward and cost may depend on many factors related to the user's current status and past interactions. To simplify the discussion, we assume that the user state contains the information sufficient to determine the lap reward and cost (in addition to the user action model) given any interface card:

Definition 3.9 (Action Reward and Cost). The lap reward and cost are determined by the user's action, the user state, and the system's previous interface card (if any), and are also called the *action reward and* action cost: $r^t(I,T,U,S) = r(a^t|z^t,q^{t-1})$, $c^t(I,T,U,S) = c(a^t|z^t,q^{t-1})$. (There will not be the term q^{t-1} when $t = 1$.)

We expand out the cumulative interaction reward and cost as a summation over action reward and cost, forming the computational basis for our proposed search simulation evaluation framework:

$$R^t(I,T,U,S) = \sum_{i=1}^{t} r(a^i|z^i,q^{i-1}) \tag{1}$$

$$C^t(I,T,U,S) = \sum_{i=1}^{t} c(a^i|z^i,q^{i-1}) \tag{2}$$

4 ANALYSIS OF EXISTING METRICS

In this section, we formally analyze some commonly used existing evaluation metrics using the proposed framework to reveal the (implicit) assumptions made underlying each measure and understand how we should interpret them based on the reward and cost defined on the user simulation.

We first instantiate the framework to obtain a general simulator for classical IR metrics:

Definition 4.1 (Classical IR simulator). The simulator's task is to find relevant documents by going through a ranked list of documents. At each lap t, the interface card is the document ranked at position t. The user is assumed to sequentially browse the list and choose from three actions: click, skip or stop at each lap t. We assume the simulator will always click a relevant document, and when seeing a non-relevant document, the user may skip or stop depending on the specific setting. The lap reward is 1 for a relevant document and 0 otherwise, and the cumulative reward is thus the number of relevant documents the simulator scanned through. The lap cost is 1 for each document scanned by the simulator, and the cumulative cost is the total number of documents the simulator scanned through. The cumulative reward and cost are recorded in the user state.

The classical IR simulator serves as a common basis for further instantiations into specific simulators corresponding to each classical IR evaluation metric. In the following sections, we assume we have a test collection consisting of a number of queries and the relevance judgment labels of a set of documents with respect to each query, and our goal is to evaluate a ranked list of results generated by a system in response to a query. We will show that Precision, Recall, and Average Precision can all be interpreted from the perspective of our proposed reward and cost measures when specific simulators are used. These simulators can help reveal the assumptions made by these measures and also provide interpretations of them from a user's perspective.

We first examine precision and recall, two of the most fundamental metrics in IR:

Definition 4.2 (Precision). Given a list of retrieval results, the traditional measure Precision can be defined as the ratio of interaction reward and cost, i.e., $R(I,T,U,S)/C(I,T,U,S)$, of a classical IR simulator that would never stop until having scanned through the whole result list.

The Precision Simulator shows clearly that Precision is focused on measuring the reward per unit of cost, but does not take into consideration of task completion; the task is not well specified, but the implied task can be assumed to be to find as many relevant documents as possible.

Definition 4.3 (Recall). Suppose there are N relevant documents in the collection. Given a list of retrieval results, the traditional measure Recall can be defined as the task completion percentage $R(I,T,U,S)/N$, i.e. the interaction reward relative to the best possible interaction reward for perfectly completed task, for a classical IR simulator that never stops until having scanned through the whole list.

It is easy to see that the assumed task in the Recall Simulator is to find *all* relevant documents. Meanwhile, Recall is only focused on the collected reward, but does not measure the cost at all. Even if we combine Precision and Recall, there is still no direct measure of the cost, and the cost is only indirectly reflected in the Precision (relative to the reward). Interestingly, we can interpret the reciprocal of Precision as the average cost per relevant document (more generally, cost/reward ratio).

Definition 4.4 (Precision@K / Recall@K). Precision@K and Recall@K are defined similarly as how Precision and Recall are defined except that such a simulator would stop when the accumulated cost (which is equal to the number of documents examined by the simulator) reaches K.

This definition shows that Precision@K and Recall@K can be interpreted as Precision and Recall with a "cost budget," i.e., the simulated user wants to control the amount of effort. We can thus easily generalize both measures by allowing variable cost in examining each document/snippet (e.g., examining a longer document/snippet would have a higher cost) and using a cost threshold τ_c, leading to Precision@τ_c and Recall@τ_c, respectively.

We now examine one of the most important measures, Average Precision (AP). We first define the variable-recall simulator:

Definition 4.5 (Variable-Recall Simulator). A variable-recall simulator is a classical IR simulator whose task is to collect N' relevant documents, where $1 \le N' \le N$ (N is the total number of relevant documents). The simulated user never stops scanning through the list until either the task is completed or the list is exhausted.

Definition 4.6 (Average Precision). In the simulation framework, Average Precision can be defined as the average ratio of the interaction reward and cost: $R(I,T,U,S)/C(I,T,U,S)$ for a set of N variable-recall simulators, each with the task of collecting $1, 2, \ldots, N$ relevant documents, respectively.

By examining AP in the simulation framework, we see that AP should be interpreted as the average performance of a system on a set of *different* retrieval tasks or *different* simulated users. While the Precision and Recall simulators only simulate a single user/task, the AP simulator simulates a set of users with variable recall demand; this explains why AP is more discriminative than Precision/Recall, and is thus also more suitable for comparing two ranked lists. This analysis result further suggests that in general, we can systematically vary the parameter of any simulator (recall in the case of AP) to obtain more discriminative measures that can better detect even the smallest differences between two ranking methods; AP is only one of the many such possibilities and may not necessarily be the best one.

The variant of AP@K can be easily derived by setting a cost budget for all the simulated users as in the case of Precision/Recall@K.

Many other evaluation metrics such as Mean Reciprocal Rank (MRR) [6], Ranked-Based Precision (RBP) [18], Normalized Discounted Cumulative Gain (NDCG), time-based measures [23], can also be studied rigorously in the framework to reveal their assumptions about users and tasks. For example, MRR is obtained when a precision simulator has a task of only finding one relevant document (and then stop). RBP assumes, on top of the precision simulator, a constant stopping rate at each position of the ranked list. In NDCG, the discounting factors for each ranked position also correspond to the simulator's stopping rate at each position, and the overall gain calculated is the simulator's expected reward over all stopping positions. The time-based evaluation is closely related, only except that the probability of stopping depends on the time spent into the search session (i.e., time cost) instead of on the lap count. Due to the space, we cannot include details of these derivations.

We could also easily extend our instantiations to generalizations of evaluation metrics on session search. For example, Session NDCG [11] could be derived similarly as classic NDCG, only with the additional simulator action model for continuing / abandoning the search after scanning through the document list of each query in the session. The U-measure based on trail-text proposed in [21], as another example, could be derived from our proposed framework by dividing the simulator's interaction with the system into word-level laps, and the simulator may abandon the search after reading till each word (e.g. in snippet, document, etc.).

The great generality of our framework is not a coincidence; it is a natural consequence of the basis of our framework - the simulator and the reward / cost measures - which are the minimal basis that maps to real world users and what they care about in an IR system; all existing metrics tried to achieve the same goal but with additional simplification assumptions for the sake of computational convenience. In particular, for example, our analysis based on the simulator models suggest that one major class of assumption underlying the existing evaluation metrics is on when and how likely the user stops throughout the interaction, and every assumption has its own advantages as well as drawbacks when compared with real user behaviors. A very important future direction is thus to study users' stopping tendencies more rigorously and propose more realistic user stopping action models, which can then be used in the proposed framework to derive more meaningful metrics than the existing ones.

5 SIMULATED EVALUATION ON TAG-BASED SEARCH INTERFACE

In this section, we apply our proposed general framework on interactive retrieval systems that do not follow a simple ranking interface, and show that an instantiation of our proposed general framework could lead to novel evaluation method for interactive systems where no traditional evaluation methodology could be applied in a principled way.

We focus on a set of interactive retrieval interfaces where, in addition to lists of documents, tags related to the document contents are used to facilitate user navigation. A common example of such tag-based search interfaces is the faceted browsing interface, where facet filters serve as tags to help users zoom into specific subsets of the documents. The Interface Card Model (ICM) proposed in [28] led to a novel method for optimizing tag-based search interfaces via automatically adjusting the interface layout based on the screen size and the estimated user interest. To evaluate and compare these relatively more sophisticated interactive retrieval interfaces, traditional evaluation methodologies focusing mainly on assessing ranked lists of documents could not be easily applied, because the user-system interactions do not adopt a sequential scanning manner. This is also the reason why the authors in [28] could only rely on real user experiments for the comparison experiments. In this work, as an example of demonstrating the effectiveness of our proposed search simulation framework, we show that an instantiation of the framework could lead to reasonable evaluation practices of the search interfaces on different types of users (or simulators), and we also validate the simulation by comparing the simulator behaviors with real user behaviors.

To instantiate the search simulation framework into a simulator model for the tag-based search interfaces, we assume that each screen the simulator sees is an interface card; the simulator could either select a document or a tag (if shown) on the screen, or click some other control buttons (e.g. scroll down / next page) to look for new content, and then the system displays a new interface card to the user and the interaction goes on. In the traditional faceted-browsing interfaces, the interface layout is *static*: on a moderate sized screen, there is typically a tag list on the left and a document list on the right, where the user could either scan through the documents, or scan through the tags to narrow down the set of documents shown on the right; on a very small screen (e.g. of a smart phone), only one of the two lists (i.e. the tag list or the document list) could be displayed at a time, and there usually is an extra button for the user to switch between the two lists. On the contrary, the interfaces proposed in [28], which we designate by "ICM interfaces," automatically adjust their layouts (e.g. between only showing tags, only showing documents, showing half-screen tags and half-screen documents, etc.), and the user either clicks a shown document / tag or click "next page" in each interaction lap.

We define the instantiation of our proposed search simulation framework for the case of tag-based search interfaces as follows:

Definition 5.1 (Tag-based search interface simulator). A *tag-based search interface simulator* U is assumed to be interested in one or a few documents in the collection, which are designated by the simulator's *target* document(s). The simulator's task T is to find all target document(s). The simulator's action model on the interface

cards in a tag-based search interface is defined as follows (assuming τ, τ_1 and τ_2 are constants between 0 and 1):
(1) If the simulator sees a target document, they always click it, and in cases of multiple target documents, they click one of them uniformly randomly.
(2) Otherwise, if the simulator sees a tag related to a target document, they always click it, and in cases of multiple related tags, they click one of them uniformly randomly.
(3) Otherwise, they seek for the next card in a way depending on the type of the interface:

a. On an ICM interface, they always click next card;

b. On a moderately sized traditional static interface displaying both a tag list and a document list, the simulator scrolls down the document list with probability τ (designated as the *document tendency value*) and scrolls down the tag list with probability $(1-\tau)$;

c. On a very small traditional static interface displaying only a tag list *or* a document list, if the simulator faces a document list (which is usually the case for the initial interface card), they scroll it down with probability τ_1 (designated as the *document inertia value*) and switch to the tag list with probability $(1-\tau_1)$; if the simulator faces a tag list, they scroll it down with probability τ_2 (designated as the *tag inertia value*) and switch to the document list with probability $(1-\tau_2)$.

(4) The simulator only and always stops when all target documents are found.

The lap cost is 1 for each lap the simulator undergoes, and the overall evaluation metrics is the simulator's interaction cost $C(I, T, U, S)$ for completing the task.

The implicit user state of the simulator is the task, i.e. the set of target documents, plus, for interacting with the very small static interface, the additional binary status of whether the user is browsing the document list or the tag list. The parameters τ, τ_1 and τ_2 could be very different for different types of users, and could be learned from user search logs.

Such an instantiation is apparently an overly simplified model for users in the real world, and it could be easily extended in a lot of aspects to reflect more realistic settings (e.g. with consideration of information scent when the simulator decides on what link to follow). As the very first example of instantiating our proposed search simulation framework, we stick with this simplified simulator model and demonstrate that it could lead to fairly reasonable and interesting evaluation results, leaving further extensions of the simulator to future research work.

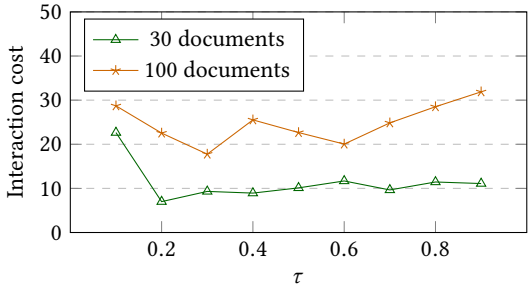
5.1 Simulated Evaluation

We implemented the tag-based search interface simulators and use them to evaluate and compare the static interfaces and the ICM interfaces on a medium screen as well as on a small screen, where we used the New York Times API[1] to obtain news articles and keywords respectively as our documents and tags. The medium screen could hold up to 2 documents or 8 tags; on the static interface, 1 document alongside 4 tags on the left are displayed at a time. The small screen could hold up to 1 document or 4 tags; on the static interface, the (simulated) user needs to switch between the document list and the tag list. We vary the number of documents in the collection as well as the parameters τ, τ_1 and τ_2. We assume the simulator is interested in only one (uniformly randomly selected) document in the collection in each search session, and we record down the average number of laps for the simulator to find the target document across multiple simulated sessions, which is an unbiased estimate of the simulator's interaction cost.

5.1.1 Medium screen. Figure 1 shows the interaction cost against different document tendency values τ on a medium screen with the static interface, and we set the number of documents in the collection to be either 30 or 100. It is firstly not surprising to find that the interaction cost is always lower on a collection of 30 documents than on a collection of 100 documents across all τ values, as it naturally takes less laps for the simulator to navigate in a smaller collection. It could also be observed that the cost tends to grow higher when τ is either too low or too high, suggesting that it is not a good idea for the simulator to stick too much to the document list (high τ), or too much to the tag list (low τ). Such an implication makes sense: sticking too much to the document list is essentially giving up the "zoom-in" functionality provided by the tags, whereas sticking too much to the tag list makes the simulator pay too little attention to the documents, which are after all what the simulator is really looking for. It is also interesting to observe that the negative effect of sticking too much to the document list (high τ) is weaker on the smaller collection, which is reasonable as keeping scrolling through a small collection is not a problem as serious as keeping scrolling through a large collection.

Note that the curves are observed to fluctuate a lot around their overall trends, since the effectiveness of the tags (news keywords) in helping the simulator narrow down to specific documents (news articles) could vary significantly depending on the specificity of the tags. Such fluctuations will also be seen in the other experiments we report.

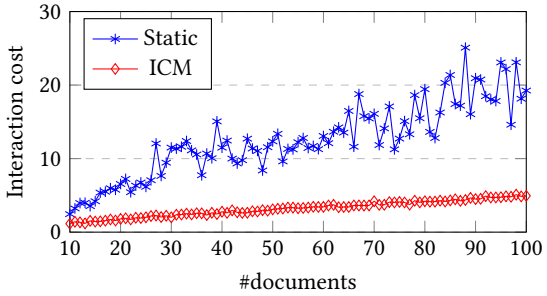
Figure 1: Cost for different document tendency values (τ) on medium screen with static interface



To use our simulators to compare the static interface with the ICM interface, we set $\tau = 0.3$ for the static interface, and Figure 2 shows the simulation result on both interfaces with various number of documents in the collection. Despite the expected fluctuations, we clearly observe that the ICM interface achieves more efficient navigation across all #documents than the static interface, and the interaction cost grows at a slower pace in the ICM interface than in the static interface as the collection size grows. We also tried setting τ to other values and obtained similar results. Such comparison outcomes coincide with the findings from real user studies in [28].

[1] https://developer.nytimes.com/

Figure 2: Cost comparison for medium screen



5.1.2 Small screen. On a small screen with static interface, there are two parameters, the document inertia τ_1 and the tag inertia τ_2, underlying the simulator's action model. Figure 3 shows the interaction cost for different combinations of τ_1 and τ_2 on top of a collection of 30 and 100 documents, with brighter color for lower cost and darker color for higher cost. In addition to what we observed on the medium screen - the cost in navigating through a smaller collection is lower than that in navigating in a larger collection - there are a couple of interesting findings unique to the small screen. Firstly, for both collection sizes, the cost is generally lower when the tag inertia is high ($\tau_2 \geq 0.7$), i.e. when the simulator tends to scan more tags before switching back to the document list. It is a reasonable strategy for the simulator to keep scanning through more tags, since discovering a good tag would eventually shrink the number of documents to look through even though it takes a few more scrolls on the tag list in the short run. Secondly, given a relatively high tag inertia τ_2, it is a good idea to keep the document inertia low in the smaller collection ($\tau_1 \leq 0.6$), while it is better to raise it higher in the larger collection (($\tau_1 \geq 0.5$). Such a finding also makes intuitive sense: when the document collection grows larger, the simulator should be more patient in scrolling through the document list rather than quickly jumping back to the tag list.

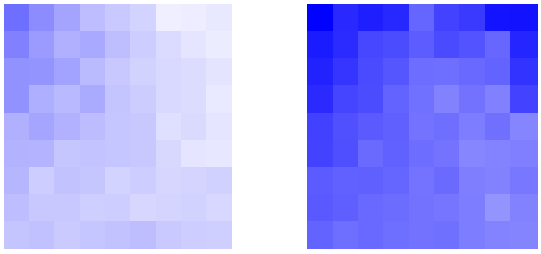
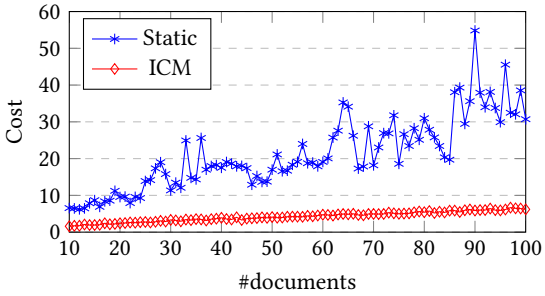


Figure 3: Heat maps of interaction cost (in log scale) for different document inertia (τ_1) and tag inertia (τ_2) values on small screen with static interface. Left: #documents = 30; right: #documents = 100. Top to bottom: τ_1 = 0.1 to 0.9; left to right (in each heat map): τ_2 = 0.1 to 0.9.

To compare the static interface with the ICM interface on the small screen, we set $\tau_1 = 0.5$ and $\tau_2 = 0.8$ for the simulator, and Figure 4 shows the interaction cost for the simulator on the two interfaces across different collection sizes. The comparison result is analogous to the one for the medium screen: the ICM interface achieves lower cost than the static interface, and the cost also grows slower on the ICM interface as the collection grows. The finding again coincides with those found in the real user studies in [28].

Figure 4: Cost comparison for small screen



5.2 Validation from real user experiment

We conducted real user experiments on Amazon Mechanical Turk[2] following the scheme described in [28], and compare real user behaviors with the behaviors of our simulators. We gave users the task of finding a target news article of their choice and asked them to navigate through the static interface and the ICM interface, on both medium screens and small screens, and we record down the users' clicks throughout the interaction. On the medium sized static interface, we compute the users' average rate of choosing to scroll the document list across all laps as $\hat{\tau}$; on the small static interface, we compute the users' average rate of choosing to scroll the list across all document screens and all tag screens as $\hat{\tau_1}$ and $\hat{\tau_2}$, respectively. Table 1 displays the result.

Screen size	Sample size	Workers' average
Small	42	$\hat{\tau_1} = 0.845$, $\hat{\tau_2} = 0.370$
Medium	38	$\hat{\tau} = 0.211$

Table 1: Real user action averages

It could be observed that on the medium static screen, the users have a relatively low tendency ($\hat{\tau} = 0.211$) on average to stick to scrolling the document list, and such a $\hat{\tau}$ value also led to a fairly good interaction cost measure in our simulation experiments as observed in Figure 1. In other words, the real users are generally able to utilize the tags nearly optimally in facilitating their navigation on the medium static screen. On the small static screen, on the other hand, the users have a high inertia ($\hat{\tau_1} = 0.845$) of keeping scrolling through the document list, but a relatively low inertia ($\hat{\tau_2} = 0.370$) of scrolling through the tag list. Such a combination of $\hat{\tau_1}$ and $\hat{\tau_2}$ values resides in the lower-left portion in the two heat maps in Figure 3, which led in sub-optimal interaction cost measures in our simulation experiments. The users navigating on the small static interface do not tend to switch to the tag list when they are scrolling through the documents, and even when they switch to the tag list, they quickly switch back to the document list without exploring more tags when they could not find a relevant tag. The reason is most likely that the small screen only has space for either the document list or the tag list, and is initially showing the document list, so a lot of users merely follow the document list, and might only consider the switch as a glimpse of what tags might be there and do not recognize the power of exploring more tags; on the contrary, the medium screen always displays both the documents and the tags, so the users are free to explore both lists without taking any extra effort in switching between the lists.

[2] https://www.mturk.com/

The authors in [28] conducted real user experiments to compare the ICM and static interfaces on small and medium screens, and concluded that the ICM interface is more efficient in helping users navigate, and also that the benefit of ICM over the static interface is more striking on the small screen than on the medium screen. In our experiment and analysis, with the tag-based search interface simulator as an extension of our proposed search simulation framework, we reached the same conclusion that the ICM interface is better as shown in Figure 2 and 4, which validates that our proposed search simulation framework could reliably assess the effectiveness of search interfaces. More interestingly, by comparing the real users' actions with the spectrum of our simulators' action model, we observe that the real users adopt a nearly optimal strategy on the medium screen yet a sub-optimal strategy on the small screen, which are novel insights into the reason why the difference between ICM and static interfaces in user navigation efficiency is more significant on the small screen as concluded in [28]. Such novel insights would be hardly possible to draw without establishing the proposed search simulation framework. These results also highlight another important benefit of the proposed simulation framework for understanding user behavior in detail by fitting simulators to real user interaction log data.

6 CONCLUSIONS AND FUTURE WORK

We presented a new general framework for evaluating arbitrary information retrieval (IR) systems based on search session simulation. The motivation for this framework is to enable reproducible experimental evaluation of sophisticated IR systems, particularly interactive IR systems, in the same spirit as the Cranfield evaluation methodology. The main idea is to generalize the current Cranfield evaluation method based on a test collection to one based on a set of user-task simulators and measures defined on a whole interaction session. We examine multiple commonly used measures in IR evaluation in this framework and show that they can all be derived as special cases of the framework under various assumptions about the user that they (implicitly) intend to simulate. Analysis of these assumptions reveals insights about how to improve these measures, which not only are practically useful, but also point out interesting new research directions. We also propose a way to construct user simulators for evaluating a set of tag-based search interfaces, and conduct simulation experiments to assess the effectiveness of different interface layout strategies. We show that such systems, which cannot be evaluated using any existing method in a principled way, can now be evaluated using the constructed simulators with interesting observations.

The proposed framework lays a theoretical foundation for experimental studies of sophisticated IR systems and opens up many new research directions. For example, we can use the framework to derive potentially better metrics than the existing ones that we analyzed, and to further analyze many more evaluation metrics of various tasks. The framework also opens up many interesting opportunities to leverage search log data to build various realistic user simulators for evaluating potentially very complicated search systems.

REFERENCES

[1] Leif Azzopardi. 2014. Modelling Interaction with Economic Models of Search. In *SIGIR '14*. 3–12.

[2] Leif Azzopardi, Maarten de Rijke, and Krisztian Balog. 2007. Building simulated queries for known-item topics: an analysis using six european languages.. In *SIGIR '07*. 455–462.

[3] Leif Azzopardi and Guido Zuccon. 2016. An Analysis of the Cost and Benefit of Search Interactions. In *ICTIR '16*. 59–68.

[4] Luca Busin and Stefano Mizzaro. 2013. Axiometrics: An Axiomatic Approach to Information Retrieval Effectiveness Metrics.. In *ICTIR '13*. 8.

[5] Ben Carterette, Ashraf Bah, and Mustafa Zengin. 2015. Dynamic Test Collections for Retrieval Evaluation.. In *ICTIR '15*. 91–100.

[6] Olivier Chapelle, Donald Metlzer, Ya Zhang, and Pierre Grinspan. 2009. Expected reciprocal rank for graded relevance. In *CIKM '09*. 621–630.

[7] Aleksandr Chuklin, Ilya Markov, and Maarten de Rijke. 2015. *Click Models for Web Search*. Morgan & Claypool Publishers.

[8] Charles L.A. Clarke, Maheedhar Kolla, Gordon V. Cormack, Olga Vechtomova, Azin Ashkan, Stefan Büttcher, and Ian MacKinnon. 2008. Novelty and Diversity in Information Retrieval Evaluation. In *SIGIR '08*. 659–666.

[9] Susan Dumais, Edward Cutrell, JJ Cadiz, Gavin Jancke, Raman Sarin, and Daniel C. Robbins. 2003. Stuff I've Seen: A System for Personal Information Retrieval and Re-use. In *SIGIR '03*. 72–79.

[10] Donna Harman. 2011. *Information Retrieval Evaluation*. Morgan & Claypool Publishers.

[11] Kalervo Järvelin, Susan L Price, Lois ML Delcambre, and Marianne Lykke Nielsen. 2008. Discounted cumulated gain based evaluation of multiple-query IR sessions. In *ECIR*. Springer, 4–15.

[12] Chris Jordan, Carolyn R. Watters, and Qigang Gao. 2006. Using controlled query generation to evaluate blind relevance feedback algorithms.. In *JCDL*. ACM, 286–295.

[13] Kalervo JÃđrvelin. 2009. Interactive relevance feedback with graded relevance and sentence extraction: simulated user experiments.. In *CIKM* (2009-11-17). ACM, 2053–2056.

[14] Diane Kelly. 2009. Methods for Evaluating Interactive Information Retrieval Systems with Users. *Foundations and Trends in Information Retrieval* 3, 1-2 (2009), 1–224.

[15] Heikki Keskustalo, Kalervo JÃđrvelin, and Ari Pirkola. 2008. 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 (2008), 209–228.

[16] David Maxwell and Leif Azzopardi. 2016. Simulating Interactive Information Retrieval: SimIIR: A Framework for the Simulation of Interaction. In *SIGIR '16*. 1141–1144.

[17] David Maxwell, Leif Azzopardi, Kalervo JÃđrvelin, and Heikki Keskustalo. 2015. An Initial Investigation into Fixed and Adaptive Stopping Strategies.. In *SIGIR*. ACM, 903–906.

[18] Alistair Moffat and Justin Zobel. 2008. Rank-biased Precision for Measurement of Retrieval Effectiveness. *ACM Trans. Inf. Syst.* 27, 1, Article 2 (Dec. 2008), 27 pages.

[19] Peter Pirolli and Stuart Card. 1999. Information Foraging. *Psychological Review* 106 (1999), 643–675.

[20] Stephen Robertson. 2008. A New Interpretation of Average Precision. In *SIGIR '08*. 689–690.

[21] Tetsuya Sakai and Zhicheng Dou. 2013. Summaries, Ranked Retrieval and Sessions: A Unified Framework for Information Access Evaluation. In *SIGIR '13*. 473–482.

[22] Mark Sanderson. 2010. Test Collection Based Evaluation of Information Retrieval Systems. *Foundations and Trends in Information Retrieval* 4, 4 (2010), 247–375.

[23] Mark D. Smucker and Charles L.A. Clarke. 2012. Time-based Calibration of Effectiveness Measures. In *SIGIR '12*. 95–104.

[24] Karen Sparck Jones and Peter Willett (Eds.). 1997. *Readings in Information Retrieval*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

[25] Paul Thomas, Alistair Moffat, Peter Bailey, and Falk Scholer. 2014. Modeling decision points in user search behavior.. In *IIiX*. ACM, 239–242.

[26] Suzan Verberne, Maya Sappelli, Kalervo JÃđrvelin, and Wessel Kraaij. 2015. User Simulations for Interactive Search: Evaluating Personalized Query Suggestion.. In *ECIR*, Vol. 9022. 678–690.

[27] Fan Zhang, Yiqun Liu, Xin Li, Min Zhang, Yinghui Xu, and Shaoping Ma. 2017. Evaluating Web Search with a Bejeweled Player Model. In *SIGIR '17*. 425–434.

[28] Yinan Zhang and Chengxiang Zhai. 2015. Information Retrieval As Card Playing: A Formal Model for Optimizing Interactive Retrieval Interface. In *SIGIR '15*. 685–694.

[29] Yinan Zhang and Chengxiang Zhai. 2016. A Sequential Decision Formulation of the Interface Card Model for Interactive IR. In *SIGIR '16*. 85–94.