Using Eye-Tracking with Dynamic Areas of Interest for Analyzing Interactive Information Retrieval

Vu T. Tran
Information Engineering
University of Duisburg-Essen
Duisburg, Germany
vtran@is.inf.uni-due.de

Norbert Fuhr
Information Engineering
University of Duisburg-Essen
Duisburg, Germany
norbert.fuhr@uni-due.de

ABSTRACT
Based on a new framework for capturing dynamic areas of interest in eye-tracking, we model the user search process as a Markov-chain. The analysis indicates possible system improvements and yields parameter estimates for the Interactive Probability Ranking Principle (IPRP).

Categories and Subject Descriptors
H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval—Search Process

General Terms
User Study, Interactive Retrieval

1. INTRODUCTION
Although interactive retrieval systems are a commodity today, the theoretic foundation for this type of systems is rather scarce. A first attempt in this direction was the IPRP formulated in [3]. However, there has been little work building upon this framework. Although there have been many user studies of search behaviour, none of them provides results that could be used for improving the functional level of an IR system.

In this paper, we present a new methodology for analysing interactive IR, extending current approaches by separating between eye-tracking and action level, and by considering dynamic areas of interest for the former. An example study demonstrates the feasibility of our approach, yielding suggestions for system improvements as well as parameter estimates for the IPRP.

In the following, after giving a survey over related work, we describe our test setting, and then analyse the experimental results, before we come to the conclusion.

The IPRP assumes that a user moves between situations \( s_i \), in each of which the system presents a list of choices, about which the user has to decide, and the first accepted choice moves the user to a new situation. Each choice \( c_{ij} \) is associated with three parameters: the effort \( e_{ij} \) for considering this choice, the acceptance probability \( p_{ij} \), and the benefit \( a_{ij} \) resulting from the acceptance. Below, we show how the first two parameters can be estimated in our setting, while the third one still is an open research issue.

By applying economic theory to IIR Azzopardi modelled the process of interaction between a user and a system and constructed a cost function to measure the user effort [1]. Cutrell and Guang [4] used eye-tracking to explore the effects of different presentations of search results. Joachims et al. [5] examined the reliability of click-through data for implicit relevance feedback of a web search engine by comparing them to eye-tracking data.

2. EXPERIMENTAL DESIGN
For our experiments, we used a collection based on a crawl of 2.7 million records from the book database of the online bookseller Amazon.com. As test subjects, we recruited 12 students of computer science, cognitive and communication science and related fields. After an introduction into the system, users had to work on three tasks, with a time limit of 20 minutes per task. Due to space limitations, we only discuss the results of the ‘complex’ task type here, where users were asked to find books about one of the following topics: 1) Trustworthy books about 9/11 terrorist attacks, 2) Controversial books discussing the climate change, 3) Highly acclaimed novels about racial discrimination.

The user interface of our system consists of four major areas: a query input field, a list of result items, a detail area showing all available data about the currently selected document, and a basket where users should place documents they deemed relevant. We wanted to collect eye-tracking data for each specific result list item, even when the user is scrolling down in this list; however, standard eye-tracking software allows only for the monitoring of static ‘areas of interest’ (AOI). Thus, we developed a framework called AOILog which automatically keeps track of position, visibility and size of all user interface objects at any point in time. Combining this information with the eye-tracking data, we always know at which object the user is currently looking.\(^1\)

3. RESULTS AND ANALYSIS
Our analysis is based on the combination of logging and eye-tracking data. For the latter, as in other studies, we focus on the so-called fixations, and also consider them only if they last for at least 80 ms, since this is the minimum time required for reading anything on the screen [2].

Corresponding to the four areas of the user interface, we can distinguish four types of user actions: formulating queries, looking at a result item, regarding document details, and

\(^1\)AOILog is an open source software; it can be easily integrated into any user interface based on Java Swing.
looking at the basket (which can be viewed as a more strategic action, where the user is thinking about the continuation of her search). Thus, a one-to-one mapping of eye-tracking data and user actions seems to be straightforward. However, a closer analysis showed that this is too simplistic.

First, we noted that users were frequently looking back and forth between the details and the basket. The behaviour was due to the fact that the book database often contains very similar entries (e.g., different editions of a book), thus forcing users to check if the current result item was substantially different from the books already placed in the basket. Occasionally, this also happened for items in the result list.

In a similar way, when formulating a new query, users did not only look at the query field, but also checked the details of the current document as well as the items in the basket.

In order to deal with these problems, we separate between two levels, the eye-tracking level and the action level. Then we define a mapping from the former to the latter, which also considers the logging data. By default, the area the user looks at defines the current action, with the following exceptions:

- When the user looks from a result item to the basket and back, without moving the item to the basket, this is counted as part of regarding the item.
- The same holds for looking back and forth between details and the basket.
- Query formulation starts when the user’s gaze wanders from the details or the basket to the query field for the first time, even when it returns to the basket/details several times before the query is actually submitted.

Applying these rules resulted in the transition probabilities and average times spent for the different actions displayed in Figure 1. From this data, a number of observations can be made:

1. Query formulation is roughly the same time as looking at two result items.
2. Retrieval quality is surprisingly low—only about 3% of the items in the result list are relevant. This is mainly due to the complexity of the retrieval tasks.
3. Since only 24% of the details regarded are judged as relevant, the quality of the result entries could be improved, in order to increase this rate (while the probability of the transition from result to details would decrease, thus leading to a reduction of the overall effort). Furthermore, we also see that click-through data is a poor indicator of relevance.
4. As mentioned above, users often compare the current document with entries from the basket. This could be supported by a similarity search function, which show the current document and similar one from the basket side-by-side.

Now we analyse this data with regard to the IPRP. The timings correspond to the effort $e_{ij}$ for evaluating a choice $c_{ij}$, while the transition probabilities give the chances $p_{ij}$ of accepting it. As a possible approach for quantifying the benefit $a_{ij}$ of a decision, we can regard the time needed for finding the first (next) relevant document. For that, we compute the expected time for reaching the basket. Here we can apply the method for computing “first passage times” in Markov networks, which leads to a linear equation system.

As results, we get 127.9 s for the query, 123.0 s for the result list and 109.5 s for the details stage. The benefits can then be defined as the time differences between the two situations of a transition invoked by accepting a choice. As could be expected, the biggest benefits are achieved when moving to the basket—but the corresponding acceptance probabilities are low. On the other hand, there are also choices with negative benefits (when going back to query reformulation or from detail to result). While the order of the benefits seems ok, negative benefits are in contradiction to the IPRP, which says that the corresponding choices are useless and thus should be avoided. This is a more general problem of parameter estimation for the IPRP: when reformulating the query, users do not really go back to the initial situation, they submit an improved query. For considering that, we would need a more complex Markov model, and a much larger number of observations.

4. CONCLUSION

In this paper, we extended current methodologies for analyzing interactive IR by separating between eye-tracking and action level, and by implementing dynamic areas of interest. Our example study shows the findings point to possible system improvements, and that click-through rates are a poor indicator of relevance. Finally, we can immediately derive parameters for the IPRP, although open problems remain.

Acknowledgement

This work was supported by the German Science Foundation under grant no. FU 205/24-1.

5. REFERENCES