Bauhaus-Universität Weimar
Faculty of Media
Degree Programme Computer Science and Media

The Ideal User of a Search Engine
Modeling and Evaluation Issues

Master's Thesis

Maximilian Michel

First Referee: Prof. Dr. Matthias Hagen
Second Referee: Prof. Dr. Sven Bertel

Submission date: August 13, 2014

Declaration

Unless otherwise indicated in the text or references, this thesis is entirely the product of my own scholarly work.

Weimar, August 13, 2014

. . . .

Maximilian Michel

Abstract

This thesis investigates how to simulate users of a search engine. We build a framework that allows to instantiate deterministic user models with cost-driven behavior. One user model is the ideal user model that represents the user with the best search behavior. Furthermore, we show how to simulate click behavior with the help of the spreading activation model. We use the TREC Session Track data to compare different instances of users. This allowed us to draw conclusions on the behavior and performance of the user models. The developed user models are also useful to evaluate rankings of search sessions.

Contents

CHAPTER 1
Introduction

Analyzing user search logs is a common research method in the field of search engine research. Time-tracked attributes like formulated queries or clicks in the result list give an insight on how users proceed to satisfy their information needs. Assuming that users more likely click relevant documents, user search logs can also be used in order to evaluate the performance of the search engine's ability with finding the relevant documents for a query. On the downside, user search logs are both difficult to acquire and difficult to adapt for the desired application. Search logs of the big commercial search engines are mostly not accessible to third parties and besides, they do not contain any information about the user's intentions and search tasks. Additionally, with search logs of real search engines come problems concerning the data privacy of individual users. To overcome these problems, one could track test persons solving search tasks under laboratory conditions. In this case the information need is predefined and the experimental setup is tailored for measuring the variables of interest. However, performing such user studies is very time consuming and therefore expensive; the outcome will be more structured and comparable, but to gain a significant quantity of data one has to invest a lot of effort. All in all, with the help of user search logs we can investigate search behavior, but measuring the direct impact of changes in the search engine's retrieval system becomes a challenging task.

The envisioned goal of this thesis is to build a framework that combines different models of user search behavior in order to automatically generate search logs of simulated instances of users. By comparing their performance on different search engines, we can dynamically evaluate the impact of manipulations or improvements in a search engine's retrieval system. Additionally, it is possible to analyze real user behavior and compare their performance to their simulated instances.

Given a task description, the simulated user chooses query terms in order to build a search session. After each query the simulated user scans the result list for relevant documents, performs clicks and decides when it is necessary to perform a new query. Each session is restricted by a predefined time limit; every action, e.g. clicking documents or scanning snippets, produces some time costs. Therefore, the simulated user should assess each decision not only by its benefits in form of information gain, but also according to residual time resources. Consequently, the ideal user is the user who accumulates the most information gain for a given time constraint.

In this thesis we focus on how to simulate the user's cost-driven decisions navigating through a given search session consisting of a set of queries and their corresponding result lists. After defining the ideal behavior, given the relevance levels for each document in the result list, we develop a way for performing relevance assessments based on a cognitive model. Finally, we compare different instances of simulated users and real users with the ideal behavior. Furthermore, we reason about the usefulness of queries and their role in the search session.

In order to compare search engines, we also investigate how the simulated users perform with the same queries on different retrieval systems. In the end we are able to determine for which search engine which user model needs which costs in order to accumulate a certain amount of information gain. As a result we get a new evaluation metric, that is transparent and easy to understand.

This thesis is structured as follows: In Chapter 2 we give an overview on the current state of search engine evaluation and we list some research that is related to our the objectives of this thesis. In Chapter 3 we introduce the general model of a user that is using a search engine and derive the model of an ideal user. In Chapter 4 we investigate the spreading activation model and we show, how we can use this cognitive model in order to simulate clicking behavior. In Chapter 5 we introduce further user models and compare them on the base of the TREC Session Track data sets. In Chapter 6 we introduce a first attempt of an evaluation metric based on our user models, which we then compare with established evaluation metrics. Finally, in Chapter 7 we summarize the content of this thesis and give an outlook on further work.

CHAPTER 2
Related Work

In the course of this thesis we introduce a way for using user models in order to evaluate the performance search engines. This chapter should give an overview on the current state of research in both information retrieval evaluation and user modeling.

2.1 Search Engine Evaluation

The most common methodology of information retrieval evaluation is based on the "traditional" evaluation approach of the Cranfield University first used in the 1960s. With their experiments the researchers tried to find out which indexing language is the most effective (the experimental setup is described in [Jon81, p. 19]). E. M. Voorhes later denotes this approach as the *Cranfield paradigm* [Voo02]. A information retrieval experiment, that follows the Cranfield paradigm basically needs three datasets: a set of documents, the *document corpus*, a set of *information needs* or *topics* and *relevance judgments* for each document that state which document is relevant for which topic. The retrieval system then retrieves relevant documents for each topic. The quality of the retrieved documents is assessed with the help of the metrics *recall* and *precision*. Recall is the fraction of relevant documents that were retrieved, and precision is the fraction of the relevant documents among the retrieved documents. This experimentation setup is highly reusable and therefore allows for adjusting and directly measuring the impact of certain parameters in the retrieval system. This is why this methodology is still the base of evaluation metrics of retrieval competitions like the TREC Session Track.

The metrics average precision (AP) and mean average precision (MAP) are evaluation metrics designed for the use of ranked result lists [Zhu04]. In order to calculate AP we sum up the precision value at each rank with a relevant document in the retrieved result list. In other words we sum up the precision values with increasing recall. AP is the average over all precisions; MAP is the mean of several AP scores for different result lists, preferably for the same information need [MRS08, p. 116]. A retrieval system with the highest MAP score is a system that returns result lists with all relevant documents for an information need at the top ranks without any non-relevant document in between. Although it is unarguable that a good retrieval system should show relevant documents at high ranks, we cannot assume that the precision of a result list actually matters to an information-seeking user. Furthermore, the recall of a result list cannot be perceived by the user, since in the most scenarios the user does not know how many relevant documents are in the collection and moreover, the user may stop investigating the result list before seeing all relevant documents. However, Stephen Robertson came up with a simple probabilistic user model that involves an estimate for stopping; yet he argues that stopping at the last relevant document in the result list is very unlikely [Rob08].

A further evaluation approach that follows the Cranfield paradigm is the *normalized discounted cumulative gain* (nDCG) [JK02]. It expresses how much *information gain* a user accumulates with viewing the result of a ranked result list. The nDCG is based on two assumptions: first, documents can have different relevance levels and users prefer documents with the highest relevance level; and second, results at lower ranks are less likely to be examined by the user and therefore contribute less to the cumulative gain. Consequently, the DCG factor is the sum of all relevance levels in a result list, usually until a certain rank position, with discounting the relevance level of each result accordingly to its rank. Because the result lists of different queries have different lengths and the amount of relevant documents may differ between different information needs, Järvelin and Kekäläinen introduce a normalization of the DCG of a result list with its ideal DCG. The ideal DCG is the DCG of a result list with a perfect ranking; that is, a ranking where the results are sorted by their relevance level such that the results with the highest relevance level are at the top ranks.

In order to evaluate the rankings of search sessions, Järvelin et al. came up with a variant of the nDCG: the *session-nDCG* [JPDN08]. Where the nDCG describes the information gain of only one result list, the session-nDCG extends this metric for a sequence of queries for one information need (a *search session*). Järvelin et al. claimed, that the result of further queries are "less valuable" than results presented earlier in a search session, because the user needs more effort to perform query reformulations. Accordingly, the session-DCG is calculated by summing up the DCG scores of each result list in the search session with discounting logarithmically the scores of result lists occurring later in a search session. In order to normalize the session-

DCG score it is necessary to determine the ideal session-DCG. Järvelin et al. propose to approximate the perfect session in two steps: first, determine the perfect ranking for one query that contains all relevant documents of the session and second, use this ranking as the result list for every query in the session. The ideal session-DCG is then the session-DCG of this perfect session. This construction of a perfect ranking is motivated by the claim, that an ideal search session consists of only one optimal query that contains all relevant documents.

In contrast to the DCG score, which represents the total information gain an average user accumulates with viewing the results of a result list, it is more difficult to interpret the session-DCG score. Where the discounting of results at lower ranks can be explained through the decreasing likelihood of views, the discounting of further queries is motivated on reasons of higher effort. However, one can argue that the higher effort of a query reformulation should not effect the cumulative information gain. When we are examining a search session, we already know that the user submitted a sequence of queries; therefore we can assume, that the user saw every result list of the search session and gains the same information independently from its position in the search session. All in all, the underlying user model of the session-nDCG is not sufficient enough to explain the discounts of relevance. It makes the performances of different settings in a Cranfield experiment comparable, but it is not clear what the score actually represent.

Alongside to the nDCG and session-nDCG, the TREC Session Track competition uses for evaluating rankings an additional evaluation metric: the *expected reciprocal rank* (ERR) [CMZG09]. In contrast to the user model of the DCG metrics, Chapelle et al. call them *position models*, the ERR metric is based on a *cascading user model*. According to the position model, the probability that a user views a result in a result list is only dependent on its rank position. As we already stated in the last paragraphs, a result at the lower ranks is less likely to be viewed, because the user probably stopped viewing the result list after the first ranks. The cascading user model elaborates this stopping behavior: a user stops viewing the results of a result list, when they are satisfied with the information gain they cumulated so far. We can explain this stopping behavior with the following example: A user scans the results of a result list that has highly relevant documents at the first four ranks from top to bottom. After having viewed the third rank, the user encountered three relevant documents. This may cause the user to stop and abandon the rest of the result list. Because of that, the highly relevant result at rank four may not be viewed regardless of its high-ranked position. The ERR metric is computed by summing up the relevance of each result in the result list with discounting each relevance value according to the result's "utility" (Chapelle et al. use the reciprocal of the result's rank) and according to the stopping probability. The stopping probability is higher the more relevant documents come before the result's rank. Chapelle et al. claim, that ERR is more sound than the DCG metrics since the underlying cascading user model is more sophisticated.

In the course of the last years, there were several comparisons of Cranfield-like evaluation metrics with user studies with actual users. For instance Turpin and Hersh found, that mean average precision performance has a weak correlation with real user performance for question answering tasks and Turpin and Schloler came to the same conclusion for "simple information-finding web search tasks" [TH01, TS06]. Smucker and Jethani claimed that the precision of a ranking has an influence on the user's behavior [SJ10]. In fact, with scanning the results of a result list with a lower precision, the users get more critical towards the relevance of the results and they become better with the distinction between relevant and non-relevant results. Smucker and Jethani found that the information gain of real users correlates with the precision overall, but when it comes to more complex interfaces this correlation gets weaker. Sanderson et al. performed a crowd-sourced experiment, where users have to decide among different result lists for a query, which one is the best ranking [SPCK10]. They found, that the preference of a certain ranking strongly correlates with its nDCG and ERR score. However, the experimentation setup does not resemble the process of a real web search.

All in all, researchers agree that more sophisticated evaluation metrics like ERR resemble the users' performance in general; but they all claim that Cranfield-style evaluation metrics "lack of realism" and sound user models [SC12].

An evaluation metric based on a user simulation was introduced by Smucker and Clarke: *time-biased gain* (TBG) [SC12]. With their approach, they try to fill in the gap between user studies and Cranfield-like experiments. The idea is to instantiate a simulation of user behavior based on real user data, that can be used to estimate how those users would perform under changing rankings. The user behavior is driven by a time limit; each action is connected with a time cost and the users need to estimate how a usage of the residual time resources. The underlying user model is a "semi-Markov" model, that consists of the following set of simple actions for processing a result list: view summary, view document, save document and view next summary. The transition probabilities between those actions are calibrated with interaction data from a group of real users that are solving an information need task in a 10 minute time limit. In order represent a population of users, the simulation works with a set of different user models, each representing a different search strategy. However, Smucker et al. did not elaborate this search strategy component further and calibrated an own user model for each of the 48 users that were involved in the data gathering process. The simulation then can be used in order to calculate a distribution of the expected information gain for different time limits and rankings. Additionally, the simulation allows for analyzing the difference of performance variance. Smucker and Clarke claim, that "If the variance of a difference is high, the effect on user experience will be low." [SC12, p. 1] All in all, the TBG approach of Smucker et al. has a comprehensible user model and with the user interaction logs a sound data basis that allows for deriving meaningful evaluation metrics like the expected number of clicked relevant documents for a given topic and time limit. However, this approach needs recalibration for different systems in order to be still representative. Addition comes with the need for representative data problems concerning the size and the composition of the group of users.

To sum up, choosing an experimental design for evaluating information retrieval systems comes always with making trade-offs between real world reference, repeatability and evaluation effort.

2.2 User Modeling

User modeling is used in order to predict and explain the user's behavior and intentions. In this section we want to give a short overview on different work in information retrieval research that is based on user modeling.

Using principles behind the cognitive architecture ACT-R, O'Brien and Keane created a model of a user who is using a search engine [OK07]. In their research they compare the model's predicted click behavior to the lick behavior of real users and additionally, they investigate whether the comparative or the threshold search strategy is more effective. Following the comparative strategy, the user first assesses all result list entries and then clicks on the most relevant; with the *threshold strategy* the user assesses the result list entries in a top-down fashion and clicks, if it is above a certain *threshold*. O'Brien et al. pointed out, that the threshold strategy should be the most effective, given the ranking of the search engine is good. With this strategy less documents need to be assessed, since the most relevant documents can be found at the top ranks. Their eye-tracking studies also showed, that the thresholding strategy is more common among their test subjects. Their user model makes utilizes the SNIF-ACT spreading activation model of information scent [FP07] in order to predict the user's clicks. This model computes activation values based on the association strength between *chunks* in the result description of result entries and topic. If the activation is above a fixed activation threshold, the user model performs a click. The model proceeds in a top-down fashion and stops searching, if it has found the relevant website. However, model does not provide the prediction when the user stops, since the relevant websites for the search task had been *hard-coded* into the experimental procedure. The evaluation is performed on user search session data, who were asked to solve 16 information search tasks. In order to investigate the search strategy and rank bias, every second result list is reversed. The result showed that users tend to click on top results, also in the reversed versions of the result list. Therefore the threshold strategy applies. Overall the model and the user data click the same ranks with the same frequency. The clicks of the result lists in normal order also show a high similarity to a power law distribution. Since this model shows a promising approach for predicting user clicks based on cognitive architecture, we will revisit the SNIF-ACT spreading activation model of information scent as one variant of simulating clicking behavior.

User-click models describe the click-behavior of information seeking users. With the help of them it is possible to infer the relevance of a document for a query from the click-through rates (the number of clicks on a result) obtained from query logs. Two examples of click models are the *user browsing model* of Dupret and Piwowarski and the *dynamic Bayesian network click model* of Chapelle and Zhang [DP08, CZ09]. In contrast to them, Zhang et. al claim that user behavior is related to the information task as a whole and therefore, the click behavior depends on previous queries and clicks for the same information task [ZCWY11]. Consequently, task-centric click model is uses the complete search session in order to infer the relevance of results. They introduce two new biases: query bias and duplicate bias. The former assume, that a user will not perform a click, if the query does not meets the user's intent, but they will however " (...) learn from the search results to re-formalize a new query".The duplicate bias states, that whenever the user encounters a webpage several times during a search session, it becomes less likely that they will click it. The click-model of Zhang et al. is a promising approach for inferring the relevance of a document from the user's click behavior in search sessions. However, click-models rely on big query logs and are not applicable for our user simulation framework.

Downey et al. introduced a model for predicting the user's next action during a search session [DDH07]. Their goal is to use this model, called *SAMlight*, in order to improve prefetching methods of search engines. After introducing a language for describing search activity models, the *event sequence space language*, they instantiated their user model using 51 events and parameters. This parameterization is done on a client-side interaction log of 250,000 users. Their model estimates the next action based on the previous actions; however, the results show that incorporating more than the last action into the estimation leads to an decrease in the predictive performance. Additionally, they figured out that among those 51 features the time between search actions is the most predictive one. In order to evaluate the predictive performance in the context of result prefetching, they compared the SAMlight model with more simpler approaches and came to the result, that their model is the best one in terms of cost reductions.

CHAPTER 3
The Ideal User

The information-seeking user uses a search engine in order to satisfy an information need. In a *search session* the user submits queries, scans the ranked results and eventually clicks on links that lead to documents that appear to be relevant. A search session for a given information need, e.g. a research task, can differ from user to user. Even if in two search sessions for one information need users submit the same queries, there are still numerous ways to decide whether to click a result or not and when to stop scanning the reults of one result list and switch to the next query.

In this chapter we formulate a general user model that represents all paths in a search session a user can follow and we introduce a method for detmining the ideal search behavior for a predifined information need. The user with ideal search behavior makes perfect relevance assessments and therefore never clicks on non-relevant results. Furthermore, the *ideal user* knows how to work cost efficient and chooses the path through the search session that shows the most relevant documents for a sufficiant cost.

3.1 Anatomy of a Search Session

In the context of this thesis we define a search session as a sequence of queries, that are submitted by a user in order to satisfy an information need. In the general literature there are two types of scenarios that motivates the user to perform a search session. In the first scenario the user tries to refind a specific website that is already known, but they cannot remember its URL. In this case the search engine becomes more of a *navigational* tool; that is why in literature this scenario is often called *navigational search*, *focused search* or *known-item search*. Second, there is a scenario where the user searches information about a certain topic or where the user is trying to find an answer to a question. Consequently, there are numerous websites that are relevant for the user's information need. In this case the search session becomes more *diverse* or faceted, since most topics have several facets or subtopics that need to be dealt with. In the literature this scenario is mostly called *informational search*. There is also a third search scenario in the literature where the user has the intent to buy something, namely the *transactional search*. However, in the course of this thesis we focused on informational search.

A user usually performs a search session because one query is not enough to satisfy the information need. If a user posts more than one query that matches one topic, they perform a *query reformulation*. Xiang et al. distinguish between three reformulation strategies: *specialization*, *generalization* and *general association* [XJP$^+$10]. A succeeding query is specialized, whenever the result list of the current query is too diverse or when there is ambiguity; in contrast, a query is generalized whenever the results do not cover enough of the information need. The work of Hagen et al. as well as the work of Xiang et al. are two examples where specialization is implied by adding new terms to the query and generalization by removing terms [HGBS13, XJP$^+$10]. Hollink et al. claim that these reformulations have opposite functions [HHdV12]. With the third reformulation strategy, the general association, the user submits a new query that may not contain any term of the preceding queries but that is semantically related to the rest of the search session. For instance if the user searched for *New York* and the consecutive query is *Hotel*, those queries have no lexical similarity but probably belong to the same search intent. Although in this case the second query is very underspecified, we know that the user might search for hotels in New York City. If the retrieval model of the search engine is *context aware*, it will preferably show websites of hotels in New York City.

In the course of this thesis we will concentrate on how user navigate through the result lists of the queries of a search session and we will leave the simulation of query formulation and query reformulation processes for future work.

3.2 The General User Model

The information seeking user has the goal to collect as much information as possible in order to satisfy an information need. Every action that is necessary to achieve that goal comes with a certain amount of effort or *cost* and leads at best to some *information gain*. So the user has to find a trade-off between cost and benefits for their decisions, because the total cost for a search session is limited. Therefore, our model is based on cost-driven behavior. In order to describe the general user model, we have to specify a set of actions

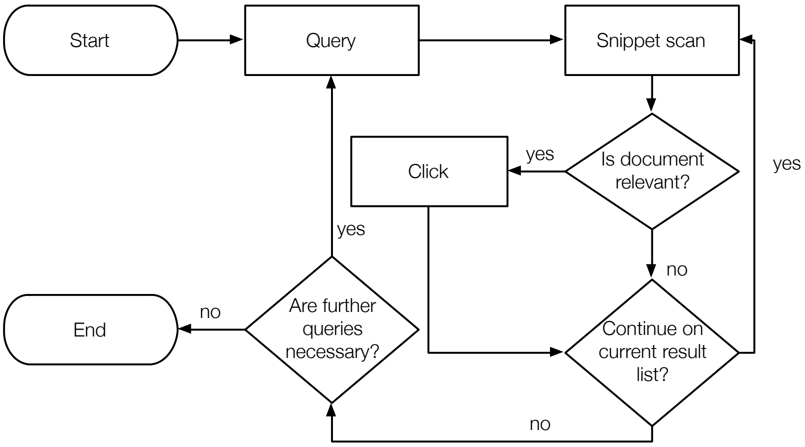


Figure 3.1: Flowchart of the general user model.

with their corresponding costs. Note that these declarations are similar to the "elementary action types" of the work of Baskaya et al. [BKJ12].

Each session S consists of an initial query q_0, a set of subsequent queries q_1 to q_{n-1} and an ending query q_n. Each submitted query leads to some costs $\mathtt{cost}_q(|q|)$ that depend on the length of the query. We define, that in a search session contains at least one query. After the user submits a query, the search engine presents the search result as a ranked list, where each result is represented with a short text snippet. The user continues with scanning those snippets. Each scan of a snippet s has some cost $\mathtt{cost}_{scan}$. We assume constant scanning costs per snippet that are independent from its length, because with scanning the user does not read every word. In our model, we define that there must be at least one snippet scan after a query, before another action can be done. After scanning a snippet the user estimates the result's relevance; if the result appears to be relevant for the given task, the user clicks it. Each click c has some cost $\mathtt{cost}_{click}$ and lead to some information gain. The result's information gain is the value of it's relevance level $\mathtt{rel}$.

In Figure 3.1 we can find a flowchart that describes our general user model. The user starts the search session with a query, which is followed by at least one snippet scan. After each snippet scan the user has to decide, if the result is relevant to the task or not. Whenever the user encounters a result, that according to its snippet's content appears to be relevant, they click it. For every click the user experiences an information gain in form of the result's relevance level. After each snippet scan and after each click, the user has to decide if they proceed with the result list of the current query and scan the next snippet in the result list, or if they submit a new query and proceed on the next result list. A search sessions ends, when there are no further queries necessary to fulfill the task.

To sum up, in our general user model the user has to make three decisions: first, when to click a result; second, when to submit a new query; third, when to end the search sessions. Every concrete user model that is derived from our general user model defines its behavior through those three decisions.

3.2.1 Implications

Based on our general user model, we can infer some characteristics of the produced search sessions. Our user model is a simplification or abstraction of complex behavior patterns that are based on cognitive processes that might differ from user to user; consequently, not all possible search behavior can be expressed with our general user model. In the following paragraphs we will have a look on some design decisions we made with our general user model.

First of all, there is the way of cumulating information gain. The only way to cumulate information gain is to click documents, which can only be done after a snippet scan. However, the user might find some relevant information only by reading the document's snippet. This might be the case in a scenario where the user has to solve a task like finding the right spelling of a word, or like converting units. In the course of this thesis we assume, that the information need is big enough that clicks are necessary in order to experience an information gain.

Second, there is the scanning strategy. In our model the user processes the result list of a query in a *top-down fashion*, starting with the first item in the result list. Klöckner et al. found with the help of eye movement experiments that this *depth-first strategy* is used by a majority of users [KWJ04]. However, our user model does not represent the 15% who either use the *breadth-first strategy* or both.

Last of all, there is the click decision. Our general user model is designed in a way such that the users assess the document's relevance right after scanning its snippet. If it is above a certain relevance *threshold*, they will click it. Users might proceed differently when they are navigating through websites that are no search engine result pages. The information foraging theory, for instance, states, that users at first assess all links that are presented to them and then decide which one of them leads to the most information gain [PC95]. However, this theory of Pirolli and Card is not tailored to the search engine scenario. Furthermore, O'Brien et al. found that the *thresholding strategy* is not only the most common strategy among users, but also the most efficient when it comes to interactions with search engines [OK07].

3.2.2 Duplicate Result List Entries

Our general user model implies that when a result list entry has been scanned, the click decision is merely dependent on the entry's relevance. However, there is a situation when the user might not click on a relevant document. Because during a search session the user submits reformulated queries, it is very likely that documents reoccur on different result lists. A relevant document that reoccurs leads only to an information gain the first time it was clicked, when we assume that the user reads it completely; therefore it is not useful to click it again. Consequently, the click decision is not only dependent on the document's relevance but also on the condition whether the document has been clicked before.

One may argue that there are situations, where the user benefits from clicking again a document; for example, when the user encounters a long document that contains several parts that are relevant to the user's search intent. The user may not get all of the information by reading the document the first time they encounter it, because they skipped parts. In this case, revisiting the document at a later point in the search session in order to get the missing parts would lead to an information gain. In this scenario it would be reasonable to count the information gain for one document several times. In order to detect such situations we would need to model how much information a user gets from a document by reading it after one query and how it differs from reading it from the viewpoint of another query. For the sake of simplicity, we assume that with clicking a document the user gains all the information that the document provides and that they gain nothing from clicking it again.

3.3 Determining the Ideal Search Behavior

In this section we investigate how we can simulate the ideal user; that is, the user that cumulates as much information gain as possible for a certain cost limit. We introduce an algorithm that takes as an input a search session, consisting of a sequence of queries with their corresponding result lists and relevance judgments for every document that occurs in the session and outputs the ideal path through this search session. From this path we can see for each result list in the session, until which rank the ideal user scans the results and which of them are clicked.

According to our general user model, the user needs to make three decisions: when to click a document, when to stop scanning one result list and change to the next one and when to end the search session. The latter decision is predefined by the search log, since we know which query is the last one of the search session. For the clicking decision the ideal user model utilizes *optimal clicking behavior.* That means the user clicks a document when it is relevant and when the user has not clicked it before. In order to define the ideal user model, we have to find a way for determining the optimal point when to change between the result lists.

We define the limit l as the rank in the result list R, where the ideal user stops scanning and submits a new query. So, for instance, if $l = 10$ the user scans the first 10 entries in a result list. While scanning the result list the user produces for each result snippet scanning costs $\mathsf{cost}_{\mathsf{scan}}$. Whenever the ideal user encounters a result $r \in R$ with a relevance level $\mathsf{rel}(r)$ above a certain relevance threshold τ_{rel} that they did not click before, the user clicks the item and additionally produces click cost $\mathsf{cost}_{\mathsf{click}}$. The document is then added to the list of clicked documents $\mathsf{Clicked}$. Consequently, the cumulated cost $\mathsf{Cost}(l, q, R)$ for a limit can be calculated as follows:

$$\begin{aligned}
\mathsf{Cost}(l, q, R) &= \mathsf{cost}_{\mathsf{query}}(|q|) + \sum_{i=1}^{\mathsf{limit}} \mathsf{cost}(r_i) \\
\mathsf{cost}(r_i) &= \begin{cases} \mathsf{cost}_{\mathsf{scan}} + \mathsf{cost}_{\mathsf{click}}, & \text{if } \mathsf{rel}(r_i) \geq \tau_{rel} \text{ and } r_i \notin \mathsf{Clicked} \\ \mathsf{cost}_{\mathsf{scan}}, & \text{otherwise} \end{cases}
\end{aligned}$$

For every click the user cumulates information gain. In our model we set the information gain equal to the relevance level of the entry. We can compute the cumulated information gain $\mathsf{Gain}(l, R)$ for one result list until a limit as follows:

$$\begin{aligned}
\mathsf{Gain}(l, R) &= \sum_{i=1}^{l} \mathsf{gain}(r_i) \\
\mathsf{gain}(r_i) &= \begin{cases} \mathsf{rel}(r_i), & \text{if } \mathsf{rel}(r_i) \geq \tau_{rel} \text{ and } r_i \notin \mathsf{Clicked} \\ 0 & \text{otherwise} \end{cases}
\end{aligned}$$

The process of determining the ideal search behavior can be formulated as a *multiple-choice knapsack problem.* For each result list of each query q in the session S we have to choose one limit such that the cumulated information gain is maximized and the cost limit $\mathsf{cost}_{\mathsf{max}}$ is not exceeded.

$$\begin{aligned}
\mathsf{maximize} \quad & \sum^{q,R\in S} \mathsf{Gain}(l, q, R) \\
\mathsf{while} \quad & \left(\sum^{q,R\in S} \mathsf{Cost}(l, q, R) \right) \leq \mathsf{cost}_{\mathsf{max}}
\end{aligned}$$

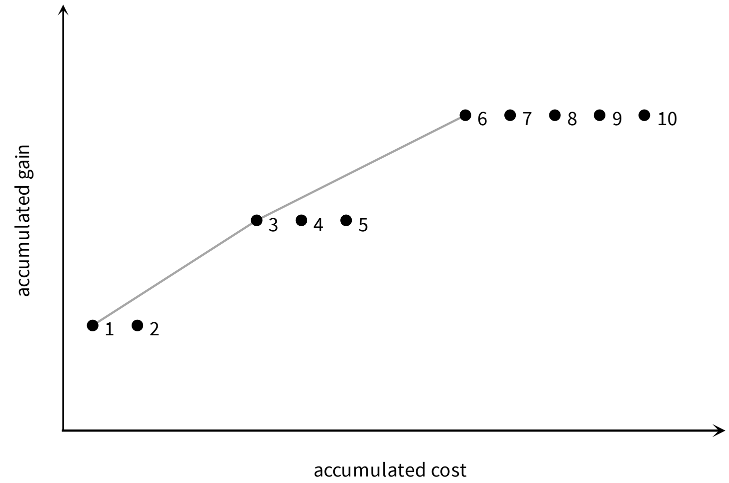


Figure 3.2: Each point shows the cumulated cost and gain for one limit. The dominating limits (1,3 and 6) build the convex hull for this problem space.

Kellerer et al. showed that the multiple-choice knapsack problem is NP-hard [KPP04, p. 318]. In order to shrink the problem space we can omit *dominated states*, because a dominated state is never part of an optimal solution for a knapsack problem [KPP04, p. 51]. In our case a state corresponds to a limit of a result list. We define dominated limits in the following way:

Definition 1 *Given a result list* R *for a query* q*, we define that one limit* l *is dominated by another limit* $\mathsf{l}' \neq \mathsf{l}$*, when one of the following condition holds:*

$$\begin{aligned}\mathsf{Cost}(\mathsf{l},\mathsf{q},\mathsf{R}) > \mathsf{Cost}(\mathsf{l}',\mathsf{q},\mathsf{R}) \quad &\textit{and} \quad \mathsf{Gain}(\mathsf{l},\mathsf{q},\mathsf{R}) \leq \mathsf{Gain}(\mathsf{l}',\mathsf{q},\mathsf{R}) \textit{ or}\\ \mathsf{Cost}(\mathsf{l},\mathsf{q},\mathsf{R}) \geq \mathsf{Cost}(\mathsf{l}',\mathsf{q},\mathsf{R}) \quad &\textit{and} \quad \mathsf{Gain}(\mathsf{l},\mathsf{q},\mathsf{R}) < \mathsf{Gain}(\mathsf{l}',\mathsf{q},\mathsf{R}) \quad .\end{aligned}$$

Figure 3.2 shows the limits of a sample result list. Each point represents the cost and gain the user would cumulate, if they stop scanning the result list at this rank and click every relevant document. In this example the relevant documents can be found at rank 1, 3 and 6. We can see that, for instance, limit 4 is dominated by limit 3, because until limit 3 the user cumulates the same information gain as for limit 4 but needs less costs. For the ideal user model each rank where the user performs a click is a dominating limit.

In order to find the ideal behavior, we have to choose from each result list in the session the limit that in total leads to the optimal solution; that is, the highest information gain possible for a given cost limit. There are several algorithmic solutions for such a multiple-choice knapsack problem. David Pisinger proposed a solution based on dynamic programming [Pis95]

```
  Data: $S = \{(q_0), \ldots (q_n, R_n)\}$, $\mathtt{rel}(r_i) \mapsto \{0, 1\}\ \forall r_i \in R$
  Result: $\mathcal{P}$
1 $\mathcal{L} \leftarrow \{\emptyset\}$
2 for $\forall (q, R) \in S$ do
3     $L \leftarrow \{\mathtt{rank}(r) : \forall r \in R | \mathtt{rel}(r) = 1\}$
4     $\mathcal{L} \leftarrow \mathcal{L} \cup \{L\}$
5 end
6 $\mathcal{P} \leftarrow L_0 \times \cdots \times L_k\ \forall L \in \mathcal{L}$
7 return $\mathcal{P}$
```

Algorithm 1: Algorithm for determining the path distribution of a search session.

and Dyer et al. developed one approach based on the branch-and-bound strategy [DKW84]. However, we cannot apply any of these approaches to our problem for the following reason: since we do not allow for clicking a relevant document if it has been clicked before, each click has an influence on the information gain of a succeeding result list. If the user clicks a relevant document in the current result list, it is no longer relevant for the next result lists. In other words, we cannot treat the result lists independently, since the click decisions of one result list can have an effect on other result lists in the session. This is why we have to check every combination of dominating states and check whether this is the ideal path. In detail, we can formulate this procedure as follows:

We define a path through a search session as a list of dominating limits for every result list in the search session $P = \{l_0, \ldots l_n\}$ and $\mathcal{P}$ as the path distribution of all possible paths. In order to find the path that represents ideal user behavior, we have to calculate for every path in $\mathcal{P}$ the total cost $\mathtt{Cost}(P, S)$ and the total gain $\mathtt{Gain}(P, S)$ as follows:

$$
\begin{aligned}
\mathtt{Gain}(P, S) &= \sum^{\substack{l \in P, \\ q, R \in S}} \mathtt{Gain}(l, q, R) \\
\mathtt{Cost}(P, S) &= \sum^{\substack{l \in P, \\ q, R \in S}} \mathtt{Cost}(l, q, R) \quad .
\end{aligned}
$$

We then choose the path that does not exceed the cost limit and that has the highest gain. Algorithm 1 shows how we can obtain the path distribution $\mathcal{P}$ from a search session S and a relevance judgment $\mathtt{rel}(r) \mapsto \{0, 1\}$ for every result list r in R. We start with determining all dominating limits in every result list in the session (Line 3); that is, the ranks of the relevant results. In

```
  **Data**: $\mathcal{P}$, $\mathtt{cost}_{max}$, $S$
  **Result**: $P_{ideal}$
1 $\mathcal{P} \leftarrow \mathtt{sort}_{Cost(P,S)}\left(\mathtt{sort}_{Gain(P,S)}(\mathcal{P})\right)$
2 $\mathcal{P}' \leftarrow \{P : \forall P \in \mathcal{P} | Cost(P,S) \leq \mathtt{cost}_{max}\}$
3 $P_{ideal} \leftarrow \arg\max_{Gain(P,S)}(\forall P \in \mathcal{P}')$
4 **return** $P_{ideal}$
```

Algorithm 2: Algorithm for selecting for a given cost limit the ideal path with the highest information gain from the path distribution.

Line 6 we build the combinations of all dominating limits of every result list and finally, we return this distribution of possible paths.

Algorithm 2 shows how we choose the ideal path for a cost limit $\mathtt{cost}_{max}$ from the path distribution $\mathcal{P}$ we obtained from Algorithm 1. We first filter the distribution for all paths that do not exceed the cost limit (Line 2) and then we choose the path with the highest gain (Line 3). In order to make this algorithm more efficient it is advisably to first sort the paths in the path distribution primary for their cost and secondary for their gain (Line 1).

3.4 Summary

In this chapter we introduced the general user model. A user model that is derived from this general user model needs to model the clicking behavior and the search strategy. The clicking behavior defines the decision whether to click a result or not. The search strategy defines the decision when to stop scanning one result list and change to the next one and the decision when to end the search session. We then derived the ideal user model, that cumulates the as much information gain as possible for a given cost limit. Additionally, we showed how to determine the ideal path the user will choose for a given search session and cost limit. In the next chapters we will derive further user models form the general user model that utilize different clicking behaviors and search strategies.

CHAPTER 4
Modeling Relevance Assessments

In the previous chapter we introduced a deterministic model that includes clicking and stopping decisions of a user who aims for the most information gain for a given cost limit. In order to determine the ideal user behavior, we assume that the relevance information of each document presented to the user is given. However, if we want to perform user simulations on a document corpus without relevance information, we need to find a substitute. In this chapter, we develop an approach for simulating the user's assessment of a document's relevance in reference to a task description, which we can use to simulate alternative clicking behavior to the optimal clicking behavior of the ideal user model. As a base, we decide to use a cognitive model based on the *information foraging theory*: the *spreading activation model of information scent*. This approach utilizes a relevance criterion, the *activation level*, which can be computed independently from the search engine's document corpus. After introducing this cognitive modeling approach, we further investigate its implementation.

4.1 Cognitive Modeling

In order to model the user's relevance assessment, we decided to use a *cognitive model* that is based on information foraging theory. Cognitive models explain basic cognitive processes (e.g. perceiving, learning, moving or decision making) and their interactions in order to infer models of more complex cognitive processes. In contrast to conceptual frameworks that describe cognitive processes in verbal form, cognitive models are described formal or in computer languages [BD09]. Therefore, cognitive models can not only be used in order to descibe cognitive processes, they also can be used for their simulation and since they are based on basic principles of cognition, they allow for valid generalizations. Consequently, with cognitive models we can make predictions that go far beyond the original data they are built on. This is a big advantage to statistical models; instead of inferring a posterior description of generated data, with cognitive models we can find explanations for cognitive processes in an inductive way.

4.2 Information Foraging

Pirolli and Card's information foraging theory describes the way how users proceed when they search for information [PC95]. It is based on the term *informavore* by Miller, who stated that the humans have the constant need to consume information to keep their mind alive [MM83]. Miller compared the consumption of information to the consumption of food; this is why the terminology of this topic is often related to nutrition. The information foraging theory states that the information seeking user behaves similar to our animal ancestors while searching and hunting for food. They are faced with traces in form of navigational cues (for instance, links), that emit *information scent*. The cue that emits the most information scent is the most promising information source.

According to information foraging theory, the user follows the navigational cue with the most information scent. This rational behavior aims for an effective trade-off between cost and benefit and matches our user model of ideal search behavior. However, the underlying user model of information foraging theory is searching with a different strategy than our user model [FP07]. In contrast to the thresholding strategy of our user model (see Section 3.2.1) information foraging is based on the *comparison strategy*. That means that the user first scans all links of a website and then decides which link they attend on. This strategy has the advantage, that the model does not need a relevance threhsold. Still, in the context of a search engine interface, the comparison strategy is much more costly than the thresholding strategy, because every click decision includes the scanning costs for the whole result list.

Fu and Pirolli developed the cognitive architecture SNIF-ACT, which is based on information foraging [FP07]. In order to model the relevance assessment, we adapt their approach of calculating information scent with the help of the spreading activation model.

4.3 Spreading Activation Model of Information Scent

Fu and Pirolli used the spreading activation model of information scent in order to calculate the utility of navigational choices. This model goes back

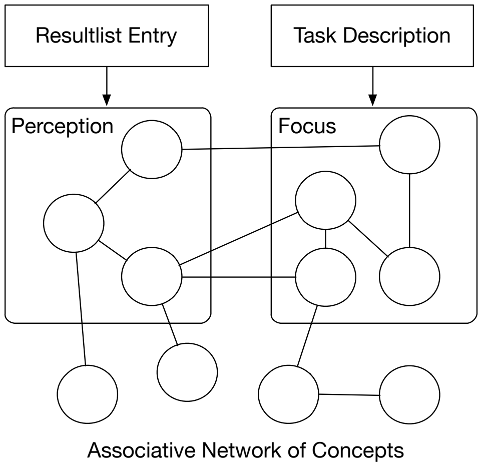


Figure 4.1: Spreading Activation in the Associative Network of Concepts.

to how the human brain actually organizes and retrieves knowledge in the *declarative memory*. The neuronal structure in the *medial temporal lobe* of the brain can be compared to a large associative network, that consists of interconnected *concepts*. A concept is an atomic piece of information like the terms “car” or “New York”. Anderson et al. described such concepts as *storage units in the declarative memory*; they call them *chunks* in the context of their cognitive architecture ACT-R [AML97]. Those chunks or concepts are connected through associations of different strengths. When the user reads or hears something, some of the concepts in this associative network are *activated*. This activation then spreads through the network and according to the associative strength other concepts are activated as well. A user who is sovling an information task is using this associative network of concepts in order to assess whether something they perceive is relevant to their intentions or not. Figure 4.1 shows an associative network, in which we marked two regions that are important for the relevance assessment of a document’s snippet: the region that is activated by reading the snippet, and the region that represents the user’s focus and intention.

A user who reads a task description builds a conceptual model of the task. That means that the user puts some concepts into the focus. While scanning the results of a result list, the user encounters the concepts of the results’ snippets and the corresponding concepts in the associative network are activated. Those activations spread through the network and eventually activate task concepts that are in the focus. The relevance is then assessed according to the total activation level of the task concepts; if the activation is above a certain threshold, the document is perceived as relevant.

	min	max	mean	median
terms in snippet	2	48	33.25	34
terms in task description	16	74	43.43	42
concepts per snippet	1	15	7.52	8
concepts per task description	4	19	9.55	10

Table 4.1: Statistics on the terms and concepts in the snippets and task descriptions of the TREC 2012 Session Track data.

In order to implement the spreading activation model of information scent, we need to make three design decisions: how to extract concepts from task descriptions and document snippets, how to calculate the spreading activation and how to choose the threshold in order to distinguish between relevant and non-relevant results.

4.3.1 Concept Extraction

The process of concept extraction is the representation of a piece of text in the task description or a document's snippet as a set of concepts in the associative network. We distinguish between two types of concepts: *topical concepts* that relate to the content of the snippet or task and *instructional concepts* that relate to certain actions that the user has to do. The latter type involves concepts like "find information" and "article" which are rather task stopwords and contribute nothing to the association strength between a snippet and the task. Consequently, it is important to recognize instructional concepts and exclude them from activation calculations in the declarative memory.

In order to extract topical concepts, we use keyphrase extraction. That means that we use an algorithm that takes a token stream of a text, performs part of speech (POS) detection and eventually extracts phrases according to certain POS patterns. Usually, keyphrase extraction involves a ranking of the extracted phrases. However, we will not use such a ranking, since we will perform attentional weighting according to the order of appearance when we compute the activation. This is why for the concept extraction we use *base noun phrases*; that are "sequences of nouns and adjectives ending with a noun and surrounded by non-noun/adjectives" [BC00, p. 3]. In Table 4.1 we can see how many terms and concepts are extracted per snippet and task description on average. The numbers are extracted from task descriptions and snippets of the TREC Session Track dataset of 2012. In Appendix A.1, we can find more information on this corpus. In general, document snippets contain less terms than task descriptions, and both have the same concept

Concept	p
who	.093
information	.089
year	.039
company	.033
name	.033
find information	.033
find	.029
help	.023
friend	.023
article	.023

Table 4.2: Ten most frequent concepts in the task descriptions of the TREC Session Track 2011-2013 data sets.

density; a phrase with the length of about 4-5 terms contains one concept on average.

We determine instructional concepts by investigating concepts that occur in task descriptions independently from the task's topic. One method to find such concepts is to test the independence of the occurrence of a concept in respect to its topic statistically with the Pearson's chi-squared test. However, since in the TREC dataset most of the tasks differ topically, it is reasonable to investigate the most frequent concepts among all task descriptions. Table 4.2 shows the ten most common concepts of the task descriptions of the TREC dataset of 2011 to 2013. In addition to this frequency analysis, we assess the task descriptions manually and add rather infrequent instructional concepts to the set of concepts that we want to omit for the spreading activation calculation.

4.3.2 Spreading Activation Calculation

After extracting concepts from the task description and the document snippets, we are now able to compute the spreading activation between them. Since we are only interested in a fraction of the concepts in the large associative network, we simplify the relevance assessment situation to a simple bipartite directed graph. Figure 4.2 shows the simplified graph for one example task. On the left-hand side there are the concepts extracted from a scanned snippet and on the right-hand side there are the concepts extracted from the task description. Apparently, the task was to find information on kabob recipes. In this associative network, we assume that all the concepts from the left-hand side are directly connected to the concepts of the right-hand side. Additionally, we omit activations that spread between concepts of one side. Based on this, we can compute the total activation level A of the task concepts C_T that spread from the snippet concepts C_S. The total activa-

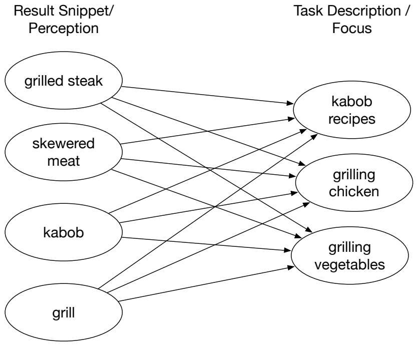


Figure 4.2: Simplified associative network in the relevance assessment situation.

tion level of a snippet for a given task is the sum of the attentional weighted association strength of every concept in C_T and every concept in C_S and can be expressed mathematically in the following way [FP07, p. 362]:

$$A(C_S, C_T) = \sum_{i \in C_T} \sum_{j \in C_S} \mathrm{association}(i, j) \cdot \mathrm{attention}(j)$$

The *attentional weighting* models the temporal decay of activations in the associative network. This follows the assumption that the user spends less attention on the latter concepts in a snippet and adds the a length normalizing property to the activation computation. In other words, we prevent boundless activations. We model this attentional decay as an exponential function:

$$\mathrm{attention}(j) = b \cdot e^{d \cdot j}$$

In this function there are two parameters: one scaling parameter b and one decay parameter d. Fu and Pirolli proposed to initialize these parameters with $b = 1$ and $d = -0.1$ [FP07, p. 363].

In order to calculate the association strength between two concepts, we use a collection of natural language texts and measure the likelihood of the co-occurrence of those concepts in respect to the likelihood of their individual occurrence in the collection. This *pointwise mutual information* (PMI) is one of the conventional measures to calculate the association strength between two concepts and can be computed in the following way [CH90]:

$$\mathrm{association}(i, j) = \log \frac{p(i, j)}{p(i)p(j)}$$

	Count	Mean	Std	Min	25%	50%	75%	Max
All	2147.00	14.93	14.29	0.00	3.24	11.09	23.08	110.29
Not Relevant	1690.00	12.79	13.59	0.00	2.35	7.97	19.73	110.29
Relevant	457.00	22.84	14.02	0.00	13.54	20.39	30.65	83.43

Table 4.3: Activation levels for all snippets of the TREC'12 interaction log.

The probabilities $p(i, j)$, $p(i)$ and $p(j)$ can be approximated with the help of the document frequencies df of those concepts. The variable N is the number of documents in the collection.

$$\mathrm{association}(i, j) = \log \frac{df(i, j) * N}{df(i) df(j)}$$

Alongside to PMI, Budiu et al. proposed to use *latent semantic analysis* (LSA) and generalized LSA (GLSA) in order to calculate association strengths between concepts [BRP07]. Both methods are based on a term co-occurrence matrix, that is used to represent a term vector of a small text snippet as a semantic vector. The semantic similarity is computed as the cosine of the angle between the semantically extended vectors of the concepts. Budiu et al. found, that GLSA is the best method when it comes to finding synonym relationships and PMI is the best method for finding semantic similarity. Additionally, they claimed that PMI is also the fastest of those three methods. This is why for computing the association strength of two concepts we are using PMI.

For the implementation of PMI we are using an inverted index, that allows for phrasal search. As a basic collection we use 100 000 English Wikipedia articles, that were drawn randomly. Because these articles can be very long and diverse, Budiu et al. propose to add the constraint, that the document frequency of two concepts $df(i, j)$ is the number of documents that contains the concepts i and j in a window of at most 16 terms.

4.3.3 Thresholding

The total activation level A indicates how relevant a result appears to the user. Since in the course of this thesis we are assuming binary relevance (relevant and not relevant), we set an activation threshold $\tau_{activation}$ that separates the activation of a non-relevant result from the activation of a relevant one. Consequently, we can define the relevance of a snippet in respect to a task description $rel(C_S, C_T)$ in the following way

$$\mathrm{rel}(C_S, C_T) = \begin{cases} 1 & \text{if } A(C_S, C_T) \geq \tau_{\mathrm{activation}} \\ 0 & \text{otherwise} \end{cases}$$

In this section we investigate, how to choose $\tau_{\mathrm{activation}}$ in two ways: first, we determine a static threshold with the help of a detailed interaction log and second, we will have a look on a variant with a dynamically changing threshold, which is adapted to the user's preference towards higher rankings (the ranking bias).

Static Threshold In order to determine the static threshold, we will use the corpus of the TREC Session Track 2012. We computed for every result in the result lists of the TREC'12 log the activation level and compared it with their relevance level. The relevance level was manually assessed; we defined every result with relevance level of at least 2 as relevant and the others as not relevant. Figure 4.3 and Table 4.3 show the distribution of the activation levels for relevant and non-relevant results. As we can see, the activation levels of relevant and non-relevant snippets are distributed differently; in fact, the means of both distributions are significantly different[1].

A common way to find a threshold between two probability distributions is the *maximum a posteriori estimation* (MAP), that finds the threshold with *minimum error* [Kay98, p. 77]. With this method we choose $\tau_{\mathrm{activation}}$ such that an activation level $A \geq \tau_{\mathrm{activation}}$ fulfills the following inequality:

$$\frac{p(A|\mathrm{rel} = 1)}{p(A|\mathrm{rel} = 0)} \geq \frac{p(\mathrm{rel} = 0)}{p(\mathrm{rel} = 1)}$$

The left-hand side of this equation is called *likelihood ratio* between the a posteriori probabilities; that are, the probabilities of seeing a document snippet with activation A among relevant $\mathrm{rel} = 1$ or non-relevant $\mathrm{rel} = 0$ document snippets. The likelihood ratio must be bigger or equal than the ratio of the prior probabilities, which are the probabilities of seeing a relevant or a non-relevant result in general without considering activation levels. In order to estimate the posterior probabilities we approximate the distributions of the activation levels for relevant and non-relevant results with a *probability density function* (PDF). We use the Kolmgorov-Smirnov test in order to find a PDF that matches our data the best; however none of the 78 PDFs provided by the statistical module of the SciPy library [2] matches our data with a satisfying

[1] T-test $t = 13.69$, $p << 0.01$

[2] http://docs.scipy.org/doc/scipy/reference/stats.html

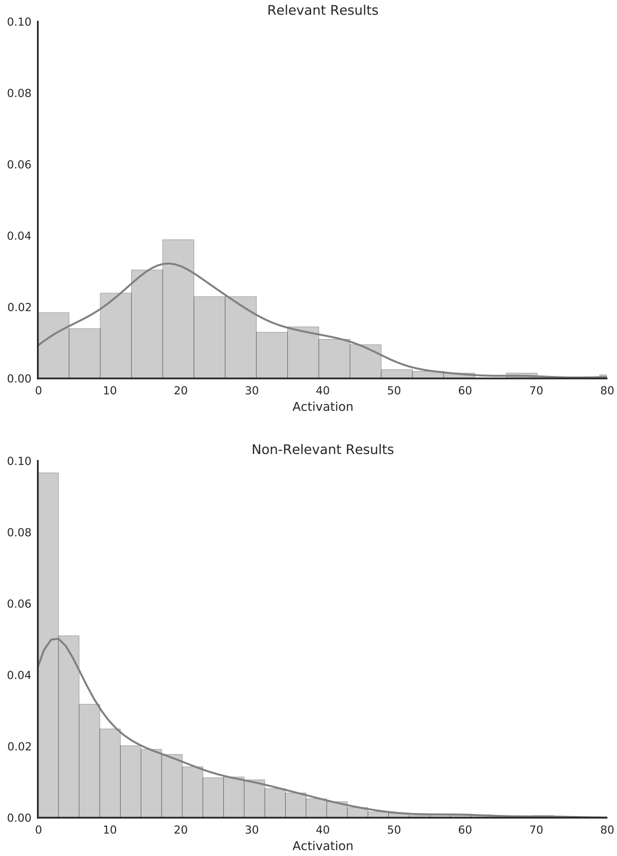


Figure 4.3: Distribution of activation levels for relevant and non-relevant results.

p-value. This is why we decided to approximate the distribution with the help of *kernel density estimation* (KDE) [Sil86, p. 14]. This method describes each data point through a kernel function (in this case a Gaussian function) and sums up all of them in order to obtain a continuous function, which is then normalized such that the integral between the minimum and maximum is 1.

In Table 4.4, we compared three thresholding methods in terms of Accuracy, F-Score and Precision and Recall. Alongside to the MAP estimated threshold, we investigated a variant of MAP that ignores the prior probabilities. This so called *likelihood comparison* chooses $\tau_{\mathrm{activation}}$, such that for each activation level $A \geq \tau_{\mathrm{activation}}$ the posterior probability of relevance $p(A|\mathrm{rel} = 1)$ is greater than the posterior probability of no relevance $p(A|\mathrm{rel} = 0)$. Furthermore we take a thresholding method into this comparison, where threshold is chosen in a way, such that it leads to the best F-score.

Method	$\tau_{\mathrm{activation}}$	Accuracy	F-Score	Precision	Recall
MAP Estimation	40	**0.78**	0.19	0.42	0.13
Likelihood Estimation	12	0.63	0.47	0.34	**0.77**
Best F-Score	14	0.66	**0.48**	**0.36**	0.73

Table 4.4: Comparison of different thresholding approaches.

If we perform the MAP estimation with the data of the TREC'12 interaction log, the threshold is at activation level 40, which is a very high threshold when you consider that the 75% quantile overall activations is at 23. As a consequence, such a high threshold leads to a lot of false negatives; in fact, only 13% of the relevant documents are above this threshold (see recall value at Table 4.4). This high threshold is due to the big difference of the priors; of all the results only 21% are relevant. As a consequence false negatives have no big influence on the accuracy. We can see that the F-score of the MAP Estimation is much smaller than the F-score of the likelihood estimation, which is similar to the best F-score possible.

Dynamic Thresholding In addition the static thresholding strategy we introduce a variant that adapts the threshold dynamically. This strategy is based on the assumption that in the result list the results are ordered by relevance. Since the users process the result list in a top-down manner, they expect every further result at a lower rank to be less relevant. In other words, we assume that the users get more and more *skeptical* as they go down the result list and therefore their activation threshold for distinguishing between relevant and non-relevant results increases. We model this strategy by adding the condition that the user only perceives a result as relevant, when its activation is higher than the last result they assessed as relevant. The user starts with a fixed activation threshold for the first rank, which represents the bias towards the retrieval system. If we set, for instance, $\tau_{\mathrm{activation}} = 0$ for the first rank, the user always assumes that the first result in the result list is relevant and always clicks it blindfold. Every further result on a lower rank must have a higher activation level than the last relevant result; hence, the activation $\tau_{\mathrm{activation}}$ is constantly growing.

This dynamic thresholding approach supports the findings of Kean and O'Brien concerning the user's rank bias [KOS08]. They found that users prefer to click results at the top of a result list; even if the results are presented in reversed order. Kean and O'Brien explain this behavior with a bias towards higher ranks coming from the experience of the users. An alternative rationale of this click behavior could be that a result on the top of the result list is accessible with lower effort than the results at lower ranks. A result with mediocre relevance which is accessible with low effort may still be more appealing than a result with high relevance at a low rank, which is more costly to reach and of which the user does not even know about. This user behavior is called *satisficing*, a portmanteau of satisfaction and suffice, and means that the user rather makes a fast decision that is sufficient than evaluating all possible actions in order to find the optimum [Man99, p. 184].

4.4 Summary

In this chapter we introduced a way to simulate the process of relevance assessment motivated from a cognitive perspective. Our model assumes that users perceive a result in a result list as relevant, when there is a strong connection between the concepts in the result's snippet and the concepts in the task description that represents the user's focus. This connection between concepts is simulated with the help of an associative network of concepts that allows us to measure the activation spreading from the perceived concepts of a result's snippet to the concepts in the user's focus. The higher the total activation is, the more relevant the user perceives a result list entry. Having calculated the activation, the user has two ways to make a click decision: either, every result list entry with an activation level above a certain threshold is clicked (static thresholding), or the activation level must be above the activation level of the last clicked result (dynamic thresholding). In the next chapter we will use this relevance assessment simulations for user models that, in contrast to the ideal user model, do not make optimal click decisions. Still these activation user models have the goal to cumulate as much information gain as possible for a given cost limit, but instead of knowing the actual relevance information, they use spreading activation as an relevance estimate.

CHAPTER 5
Comparing User Models

5.1 Further User Models

The general user model introduced in Section 3.2 allows us to formulate a variety of user models. Each of those models basically consist of two components: first, the *clicking behavior* defines when to click a document, and second, the *search strategy* defines when to stop processing a result list and submit the next query and when to end the search session. In Chapter 3 we already defined the ideal user model, that clicks every relevant result in the result list and makes stopping decisions in a way, such that for a given time limit the user cumulates the most information gain possible. In this section we introduce further user models that differ in terms of clicking behavior and search strategy. We first summarize clicking behavior and search strategies we mentioned in the previous chapters and additionally introduce new ones. Afterwards we combine them in order to describe a set of user models we want to investigate further.

5.1.1 Clicking Behavior

In order to simulate clicking behavior we use three approaches. The first approach represents the user who only clicks relevant results. This *optimal* clicking behavior is used by the ideal user model introduced in Chapter 3. With optimal clicking behavior, the user searches very cost efficient, since every click, which is the most costly search action, leads to an information gain. Therefore, the user with optimal clicking behavior saves costly clicks on non-relevant results.

The second approach is based on the spreading activation model as introduced in Chapter 4. The user clicks on a result, when the activation level of the result's snippet in respect to the user's search intent (for instance a task description) is above an activation threshold. We introduced two approaches for choosing the activation threshold: static thresholding and dynamic thresholding. The activation based clicking models are motivated by models of cognitive processes. In contrast the optimal clicking behavior, the activation approach may lead to clicks on non-relevant results, since this approach is just a model of a real relevance judgment.

In contrast to the optimal and the activation based clicking behavior, the *clicking all* approach makes click decisions independently from the relevance of a result: every result that is scanned is also clicked. This click behavior represents the user, who does not rely on the result's snippet and who wants to see every document in a result list. This clicking behavior is the least cost efficient one, since clicking every result means also clicking every non-relevant result among the scanned results.

5.1.2 Search Strategies

The search strategy of a user model defines which documents of a search session the user is viewing and which documents the user omits. Given a search session the user has to decide when to abandon one result list, submit a reformulated query and to view the results in the next result list. This stopping decision has a big influence on the information gain. If a user stops too early with viewing one result list, they might miss some relevant results; on the other hand, the more cost the user invests in one result list with scanning and clicking results, the less cost remains to view the results in further queries. In this thesis we will investigate four search strategies.

Zhang et al. analyzed queries and clicks of a commercial search engine that were submitted during one week. They observed, that users tend to click more at the end of a session [ZCWY11]. Their explanation is that with every query reformulation the user improves the quality of the query and eventually ends up with a query that describes the user's information need the best. Therefore, the result list of the last query in a search session is the result list that contains the most relevant documents for the user's search intent. Zhang et al. assume, that the user probably scans some of the results in earlier queries and use the information from the results' snippets in order to formulate better queries. Based on these observations we define the *prefer last queries strategy*. The user model that is using the prefer last queries strategy is spending most of their costs in order to view the results of the last queries in a search session. A formalized description of this strategy can be defined as follows:

Definition 2 *A path* P *consists of a list of limits* $l_0 \dots l_n$ *that represent for each query in a search session the lowest rank the user views. The path* P *follows the prefer last query strategy when the following condition holds:* $l_i > l_{i-1} \quad \forall l_i \in P$.

In contrast to the findings of Zhang et al., the user model behind the session nDCG, a metric for evaluating rankings of search sessions (see Section 2.1), is based on the claim, that results of reformulated queries are less valuable, because for viewing these results, the user has to invest more effort [JPDN08]. According to this model, the user gets the most information gain from the first queries. Based on this, we define the *prefer first queries strategy*. In contrast to the prefer last queries strategy, the user is spending their costs in order to view the results of the first queries in a search session. Besides to the claims of Järverlin et al., with this strategy the user could proceed in the following way: The user is very confident with formulating their information need in the initial query of the search session. The user then investigates the result list of this initial query extensively. All succeeding query reformulations then concentrate on minor aspects of the topic of the information need and therefore it is not necessary to investigate their result lists extensively. We can formalize the prefer first queries strategy as follows:

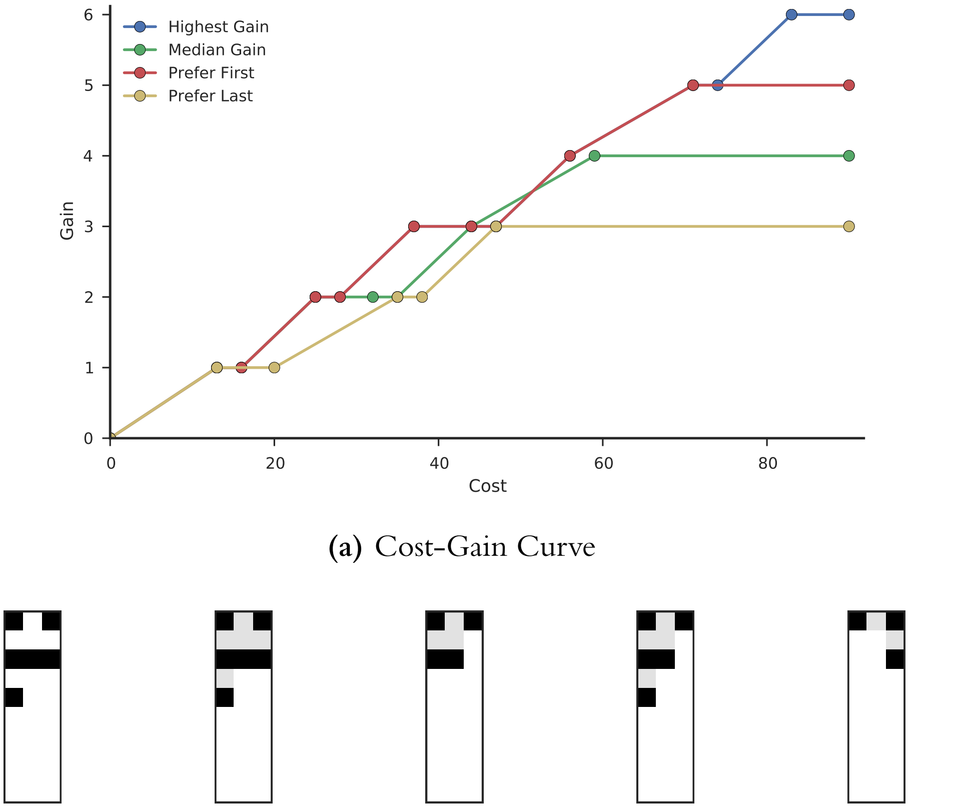
Definition 3 *A path* P*, consisting of a list of limits* $\mathsf{l}_0 \ldots \mathsf{l}_n$*, follows the prefer first query strategy when the following condition holds:* $\mathsf{l}_i > \mathsf{l}_{i+1} \quad \forall \mathsf{l}_i \in \mathsf{P}$.

In addition to the prefer first/last queries strategies, we investigate the *highest gain strategy*. The user following this strategy views as much documents that appear to be relevant as possible for a given time limit. In other words, the user spends their costs in order to click as many documents that according to the click behavior should be clicked as possible. A user model with optimal clicking behavior and highest gain strategy represents the ideal user we introduced in Chapter 3. We can formalize the highest gain strategy as follows:

Definition 4 *Let* $\mathcal{P}$ *be the set of all possible paths for a given cost limit and search session and let* $\mathsf{gain}(\mathsf{P})$ *be a function returning the cumulated information gain of a path* P*. The path that follows the highest gain strategy is the path for which the following condition holds:* $\mathsf{gain}(\mathsf{P}) = \mathsf{max}(\{\mathsf{gain}(\mathsf{P}_i) \forall \mathsf{P}_i \in \mathcal{P}\})$.

The last search strategy we are introducing in this thesis is the *median gain strategy*. With the median gain strategy the user cumulates an information gain that represents the median of all information gains of all possible paths through a search session for a given cost limit. The median gain strategy therefore represents what a user with a certain clicking behavior can achieve on average. We can formally define the median gain strategy as follows.

Definition 5 *Let* $\mathcal{P}$ *be the set of all possible paths for a given cost limit and search session and let* $\mathsf{gain}(\mathsf{P})$ *be a function returning the cumulated information gain of a path* P*. The path that follows the median gain strategy is the path for which the following condition holds:* $\mathsf{gain}(\mathsf{P}) = \mathsf{median}(\{\mathsf{gain}(\mathsf{P}_i) \forall \mathsf{P}_i \in \mathcal{P}\})$.



(a) Cost-Gain Curve

(b) Session (c) Highest G. (d) Median G. (e) Prefer First (f) Prefer Last

Figure 5.1: Comparison of different search strategies for an example session. In this example we are using the optimal clicking behavior.

In Figure 5.1 we compare the four search strategies for an example search session. In this example the user clicked with optimal clicking behavior. Figures 5.1c to 5.1f visualize the click distribution of the different search strategies. In each of these matrices each column represents a result list of a query and each row represents the rank. One cell in the matrix represents one result; for instance the cell in the top-left corner represents the result at the first rank of the first result list in the session. If the cell is black the result is scanned and clicked, if it is gray the result is only scanned and if the cell is white, the document is not scanned at all. Figure 5.1b represents the distributions of relevant documents in the search session. As one can see, the example session consists of three queries; the result list of the first query in the session contains three in the second only one and the last query two relevant results. We set the cost limit to 90. As you can see in Figure 5.1c the user model with the highest gain strategy clicks all relevant documents. Because in this example most of the relevant documents are at the beginning of the session, the prefer first queries strategy cumulates more gain than the prefer last queries strategy.

Figure 5.1a shows the cost-gain curve for the example session. It shows that how the information gain increases over time (cost). As one can see, the prefer first queries strategy and the highest gain strategy the information gain level 3 a bit earlier than the prefer last queries strategy. With the prefer last queries strategy the user abandons the first result list after the first snippet scan and needs to perform two queries and two snippet scans before encountering the next relevant document. In the cost-gain curve we also see, that for the example session each of the search strategies eventually cumulated a different amount of information gain.

5.1.3 Combining Click Behavior and Search Strategy

In the course of this thesis we define a user model through two components: the clicking behavior and the search strategy. In order to simulate how a user with certain clicking behavior and a certain search strategy proceeds in a given search session, we try to find one path that leads through the search session that does not exceed the cost limit and that represents the search strategy the best. Finding this path involves performing these four steps:

1. Based on the click behavior, determine for each result in the session whether it is clicked or not.

2. Determine all paths that do not exceed the cost limit.

3. From the path distribution, choose the path that matches the search strategy.

4. When there is more than one path that matches the search strategy, choose the one with the highest information gain.

In the last sections we introduced four models of clicking behavior and four models of search strategies. In Table 5.1 you can see a user model matrix which represents every possible combination of the search strategies and clicking behaviors we introduced in this chapter. There are 16 hypothetical user models of which we choose 8 meaningful user models that we will investigate further in the course of this thesis. Alongside to the ideal user model, that combines optimal clicking behavior and the highest gain search strategy, we will investigate the following user models:

Strategy Click	Highest Gain	Median Gain	Prefer First	Prefer Last
Optimal	Ideal User	Median User	-	-
Activation (Dynamic)	Activation User (Dynamic)	-	-	-
Activation (Static)	Activation User (Static)	-	-	-
Clicking All	Clicking All User	-	Prefer First User	Prefer Last User

Table 5.1: The user model matrix shows every possible combination of click model and search strategy. We named the users we are going to compare with the TREC user.

Median User Model The median user is the user model that combines optimal clicking behavior with the median gain strategy. With the help of this user model we can investigate what a user with a search strategy of average quality can achieve in terms of information gain. Furthermore, we want to investigate how the performance of this user model differs from the ideal user, since this could give an insight on how the quality of the rankings of the result list is, because a good ranking may not require a special search strategy in order to achieve a high information gain.

Activation User Models The activation user model combines activation-based clicking behavior with the highest gain strategy. We use two versions of the activation user model: one with static and one with dynamic thresholding. As we showed in Section 4.3.3, the spreading activation approach will not lead to perfect relevance assessments, because the quality of the relevance assessment depends on the quality of task description and snippets and besides, the spreading activation approach is still an approximation of more complex processes that are involved in a relevance assessment of a document. Therefore, the activation user models represent users, that do not perform perfect click decisions. However, we hypothesize that the cumulated information gains of the activation user models will correlate with the ones of the ideal user. Consequently, for evaluating search engines the activation user model could be used to determine performance trends, when real relevance judgments are not available.

Clicking All User Model The clicking all user model uses the clicking all behavior. This model proceeds in a way such that all result lists are viewed equally. Although this behavior seems very trivial, the clicking all user model represents behavior of a special kind of user. In fact, the all clicking user is the envisioned user of each retrieval system. When users click every result in the result list in a top-down fashion, every result appears relevant to them. Consequently, if the clicking all user achieves the same information gain as the ideal user model, the ranking of the result list is very good. This is why the all clicking user is an important reference model that we will investigate in our evaluation.

Prefer First/Last Queries User Models In contrast to the all-clicking user model, that distributes their clicks equally throughout the search session, we introduce two variants that preferably clicks documents in the result list of the first or respectively the last query. The prefer first and last queries user models combine the clicking all behavior with the prefer first or respectively prefer last strategy. Comparing those two variants, we can get an insight on how the relevant documents are distributed in a search session. If for example the prefer first queries user model achieves a higher information gain than the prefer last queries user model, we can assume that there are more relevant documents in the first result lists than in the last ones of that search sessions.

5.2 Comparison with TREC User

In the course of the annual TREC Session Track, the NIST provides a set of search sessions performed by real users. Each of those search sessions includes information on what task the user tried to solve, what queries the user submitted, what the result documents were, which of the results the user clicked and which of those results are relevant to the task. In this section we have a look on how the actual TREC users perform in comparison to our user models. This involves considerations on how the users manage their cost, how good their relevance assessments are and how much information gain they accumulate during a session. In addition, we analyze differences in the behavior with the help of a Markov model.

In general, we expect the TREC user's performance to differ a lot from the ideal user behavior in terms of information gain. This hypothesis comes from the fact, that an individual user will not be able to perform perfect relevance assessments, because they are biased towards higher ranks and because they have to assess the relevance from a short document snippet. Additionally, the TREC users will not make perfect stopping decisions and therefore may not view all relevant documents.

5.2.1 Instantiation of the TREC User Model

In order to make the real user's search sessions of the TREC Session Track data comparable with our user models, we have to instantiate a TREC user model for each search session. That means that in order to perform a comparison with our user models, we need to interpret the search session data as if they

Action	Cost
Click	15s
Snippet Scan	2s
Query	1s/word

Table 5.2: Costs for each possible action of all user models.

were produced by a user model that was derived from our general user model (see Section 3.2). In detail this means that we pare down the actual behavior to the following rules: the user processes the result list in a top down fashion, performs after a query at least one snippet scan and does the click decision right after a snippet scan.

Certainly not all users behave according to our general user model. For example in 11% of the result lists of the TREC Session Track 2012 users clicked a result at a lower rank before clicking a result at a higher a rank; that means these users probably did their click decisions after reading a sequences of snippets and not right after each snippet scan. Such behavior cannot be produced by our general user model. Furthermore, we have no information which results the user actually scanned, since this is data which can only be obtained from eye-tracking observations. All in all, a comparison between our user models and the real users is difficult; therefore, all outcomes of the comparison reported in the course of this thesis is based on a model of the real TREC users.

5.2.2 Comparison of Cumulated Information Gain

In the first experiment we investigate how much information gain the user models cumulate in a search session on average. With this information we want to find answers to the following questions:

1. Aside from the ideal user model, which user model cumulates the most information gain?
2. Which user model correlates the most with the TREC user model in terms of information gain?
3. Does the information gain of user models with the same search strategy correlates more than users with the same clicking behavior?

In order to answer these questions, we investigate the search sessions of the TREC Session Tracks of 2011-2013. First of all, we instantiate a TREC User model for each search session; in other words, we calculated the cumulated

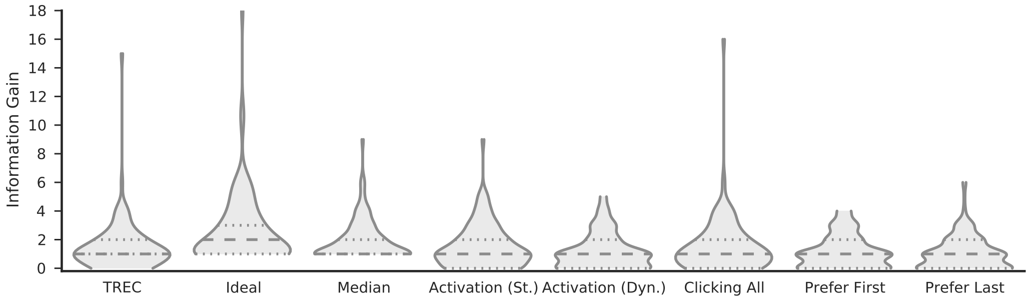


Figure 5.2: Distributions of the cumulated information gain for the search sessions of the TREC Session Track 2011-2013.

	TREC	Ideal	Median	Act. (St.)	Act. (Dyn.)	Clicking all	Pref. First	Pref, Last
count	110.00	110.00	110.00	110.00	110.00	110.00	110.00	110.00
mean	1.44	2.64	1.94	1.50	1.17	1.49	1.15	1.03
50%	1.00	2.00	1.00	1.00	1.00	1.00	1.00	1.00
std	1.76	2.51	1.40	1.52	1.26	1.94	1.16	1.17
max	15.00	18.00	9.00	9.00	5.00	16.00	4.00	6.00

Table 5.3: Comparison of the information gain the user models cumulated for the search sessions of the TREC Session Track 2011-2013.

information gain, and based on the users' actions, the total cost the users needed in order to solve the session's task. Afterwards, we determine for each search session how much information gain each user model cumulates for the same cost the TREC user needed. Table 5.2 shows how we set the cost for each possible action. We obtained the snippet scan cost and the click cost from the eye-tracking study of Tran and Fuhr [TF12] and the typing cost from the observations of Arif and Stuerzlinger [AAS09].

The data sets of the TREC Session Tracks 2011-2013 provide 288 search sessions for 160 topics. For the user model comparison we filtered out the search sessions for which it is not possible to any model derived from our general user model to achieve any information gain for the TREC user's cost limit. Such sessions either contain no relevant documents, or the TREC user's cost limit is not sufficient to scan and click one of the relevant documents. Therefore, in this comparison we use the 110 residual search sessions.

Cumulated Information Gain Table 5.3 shows the distribution of the cumulated gain over all search sessions. As expected, the ideal user model accumulates the most information gain on average, followed by the median user model that utilizes optimal clicking behavior and the medium gain search strategy. The clicking all user, the activation user with static thresholding and

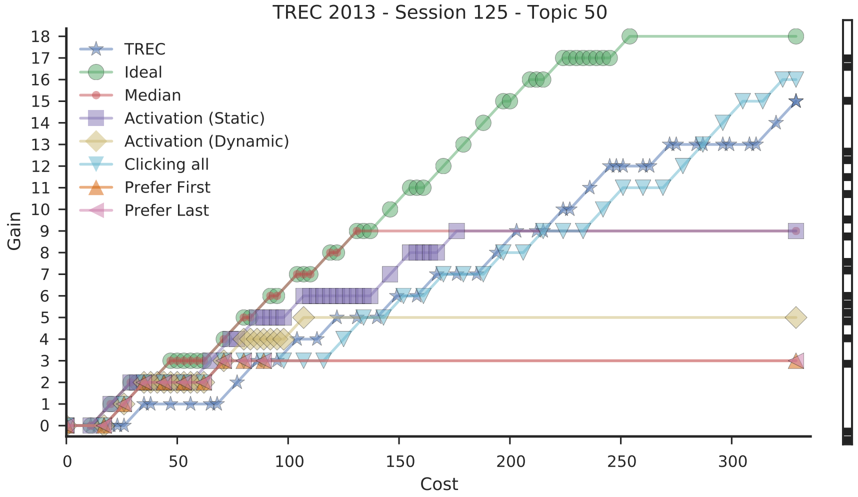


Figure 5.3: Session that caused outliers in the information gain distribution of the ideal user, clicking all user and TREC user.

the TREC user cumulate similar information gain on average. In fact, the differences between the clicking all user and the TREC user and the differences between the clicking all user and the activation user (static thresholding) are not significant; the differences between the activation user and the TREC user have only semi-strong significance[1]. The significance levels for all differences of all combinations of user models can be found in the appendix in Table A.1.

The prefer first model cumulates significantly more information gain than the prefer last model on average. Therefore, we can assume, the search sessions of our data contains more relevant documents in the result lists of the first queries than in the result lists of the last queries.

Figure 5.2 shows the distribution of the information gain as a violin plot. Violin plots are one way to illustrate distributions of data. Similar to box plots, the data points are aligned for each data set (in our case: the user models) on vertical axes. Like a vertical histogram, the thickness of the *violin* indicates relatively for how many sessions the user model achieved a certain information gain. As we can see in Figure 5.2, the maximum values of the clicking all user, the TREC user and the ideal user are outliers. In fact, these outliers belong to one search session. In Figure 5.3 we can see the cost gain curve of this session and on its right-hand side there is the distribution of relevant

[1] We tested the significance of differences with the Wilcoxon signed-rank test. We call $0.05 < p \leq 0.1$ semi-strong significance

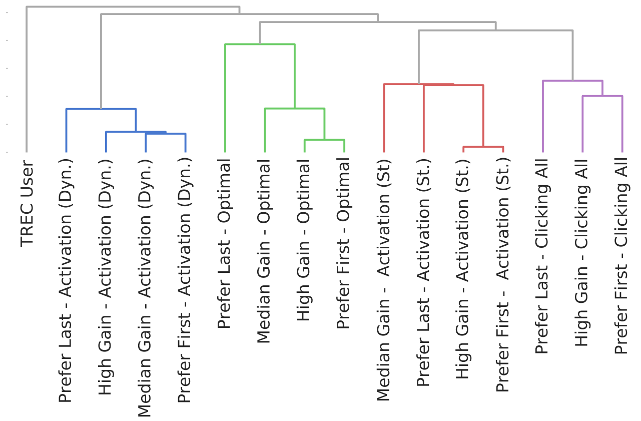


Figure 5.4: Dendrogram showing the most correlating user models. The user models were clustered with single link clustering based on their Spearman's rank correlation coefficient.

documents as a one column relevance distribution matrix. We can see, that in this search session the user submitted only one query for which the search engine returned a result list with 19 relevant documents. The TREC scanned the results until rank 40.

Correlation of User Models In order to find out which user models correlate with each other, we calculated the Spearman's rank correlation coefficient for each combination of user models. In Table A.2 we show the correlation values not only for the *meaningful* user models we introduced in Section 5.1.3 and the TREC user, but also for the other seven *hypothetical* user models that represent the other combinations of click behavior and search strategy. With the help of this correlation matrix we want to answer two questions: first, do user models with the same clicking behavior correlate more than user models with the same search strategy and second, which user model correlates the most with the TREC user?

In order to answer the first question we cluster the user models based on their Spearman's rank correlation coefficient. We decide to use single linkage clustering that iteratively links the user models with the highest correlation [Sib73]. As a result we obtain a hierarchy that indicate the correlations; the higher the hierarchy level between two user models the lower is the correlation between them. Figure 5.4 shows the result of the single link clustering in form of a dendrogram. We can see three main characteristics: First, in this clustering the TREC user is an outlier. That means, the TREC user correlates the least with all of our user models. Second, the user models with the same click behavior correlate more than user models with the same search

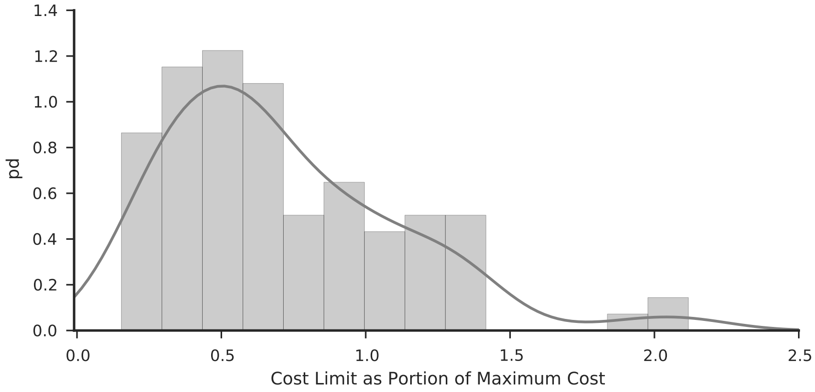


Figure 5.5: Distribution of cost limits of the TREC users.

strategy, because user models with the same click behavior are clustered on a lower hierarchy level. Therefore we conclude that the choice of the click behavior has a higher impact on the user model's performance than the choice of the search strategy. Third, among the user models with the same search strategy, the user models with the highest gain and the user models with the prefer first strategy correlate the most. This characteristic suits the observation that in the search sessions you can find more relevant documents in the first queries of a search session than in the last ones.

In order to answer the second question, we lookup the user model with the highest correlation with the TREC user in the correlation matrix in Table A.2. Among all possible user models, the TREC user correlates the most with the user model that utilizes activation based click behavior (dynamic thresholding) and prefer first queries search strategy[2]. Among the meaningful user models the activation user model (dynamic thresholding) correlates the most with the TREC user[3]. As we discussed in Section 4.3.3, dynamic thresholding models the bias towards higher ranks with a lower threshold; therefore, we conclude that the correlation with the user models with activation click behavior based on dynamic thresholding are probably caused by the rank bias of real users.

5.2.3 Comparison of Cost Usage

The tasks of TREC Session Tracks had no predefined time limit in which the users had to solve the task. Consequently, we can assume that the TREC users did not make their decisions under time pressure and decided by them-

[2] Spearman's rank correlation test $\rho = 0.65$, $p < 0.01$

[3] Spearman's rank correlation test $\rho = 0.62$, $p < 0.01$

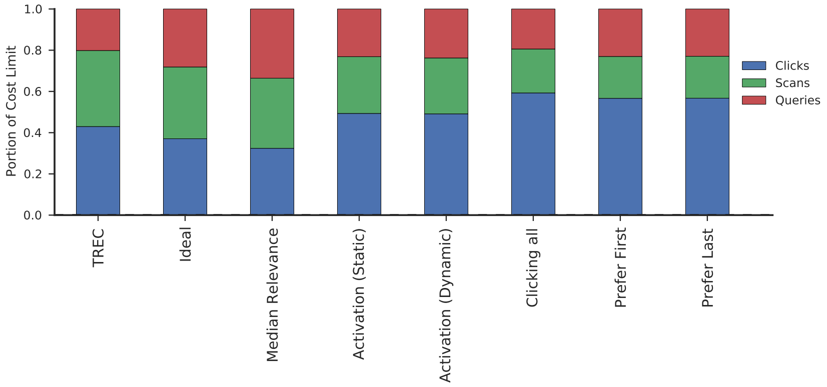


Figure 5.6: Comparison of cost composition.

selves how much cost effort they invest in order to solve the session's task. In this subsection we want to investigate the TREC users' cost effort that they invested into the search sessions in order to fulfill the sessions' task by, first, analyzing how they rationed their cost and second, by analyzing the search behavior with the help of Markov models. By comparing the results with our user models, we can find differences in terms of cost effort.

Figure 5.5 shows the distribution of the cost limits of the TREC users, presented as the portion of the maximum cost. We define the maximum cost of a search session as the cost the user would need in order to scan and click all relevant documents in a session session. On average the TREC users used 71% of the maximum cost; for half of the sessions the users invested 61% of the maximum cost. This rather small effort reflects the satisficing theory, we already discussed in Section 4.3.3. The users do not search for all relevant documents; they stop when they have viewed a sufficient amount of documents. However, in 19% of the sessions the users invest even more effort than necessary in order to view all relevant results. This high cost effort appears mostly in search sessions, where the user submits queries at the end of the session that return result lists that contain no relevant documents. In this case the user was not satisfied by the results they saw so far and wanted to search further.

In order to compare how the user models use this cost limit, we investigate the composition of the cost. Figure 5.6 shows how the costs are composed by the TREC user and the user models. As we can see, all user models spend the most cost with clicking results. However, the ideal user and the median user model tend to invest approximately the equal amount of cost into each

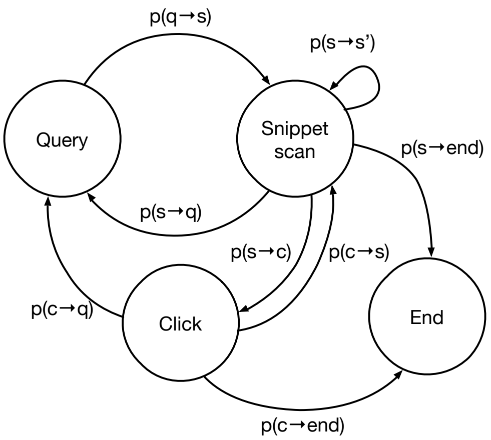


Figure 5.7: Markov model of search behavior.

of the actions. Therefore, we can conclude that the ideal user and the median user perform more snippet scans and less clicks.

He and Wang as well as Tran and Fuhr proposed to use Markov models in order to investigate search behavior of users [HW11, TF13]. A Markov model consists of a set of states and describes behavior with the probability of transitioning from one state to another state. Markov models are built on the Markov assumption, that is, the probability of transitioning to the next state is only dependent on the current state. In Figure 5.7 we can see the general Markov model. We denote the transition probability between a state a and a state b as $p(a \rightarrow b)$. We obtain the transition probability of two states from the frequency f of their subsequent appearance. For instance, we can calculate the transition probability from a snippet scan s to a click c as follows:

$$p(s \rightarrow c) = \frac{f(s,c)}{f(s)}$$

Table 5.4 shows the transition probabilities of our user models and the TREC user and Figure 5.8 shows in a box plot how these transition probabilities differ from each other. We can see that the user models differ the most with the probability of transitioning form one snippet scan to a subsequent snippet scan $p(s \rightarrow s')$ and the probability of transitioning from a snippet scan to a click $p(s \rightarrow c)$. These two transitions describe whether a result is clicked or not after it was scanned; for the ideal user model, the median user model and the TREC user it is more likely to continue with the next snippet scan, for the other user models it is more likely that they will click it. For

	TREC	Ideal	Median	Act. (St.)	Act. (Dyn.)	Clicking all	Prefer First	Prefer Last
$p(q \rightarrow s)$	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
$p(s \rightarrow q)$	0.03	0.09	0.12	0.07	0.01	0.02	0.03	0.03
$p(s \rightarrow s)$	0.56	0.52	0.51	0.30	0.35	0.03	0.00	0.02
$p(s \rightarrow c)$	0.39	0.34	0.31	0.59	0.61	0.93	0.93	0.93
$p(c \rightarrow s)$	0.55	0.47	0.34	0.57	0.45	0.58	0.50	0.47
$p(c \rightarrow q)$	0.25	0.28	0.35	0.25	0.31	0.23	0.28	0.28
$p(s \rightarrow end)$	0.02	0.05	0.07	0.05	0.02	0.02	0.04	0.02
$p(c \rightarrow end)$	0.20	0.26	0.32	0.19	0.24	0.19	0.22	0.25

Table 5.4: Transition probabilities between the actions query q, click c, snippet scan s and end of session end.

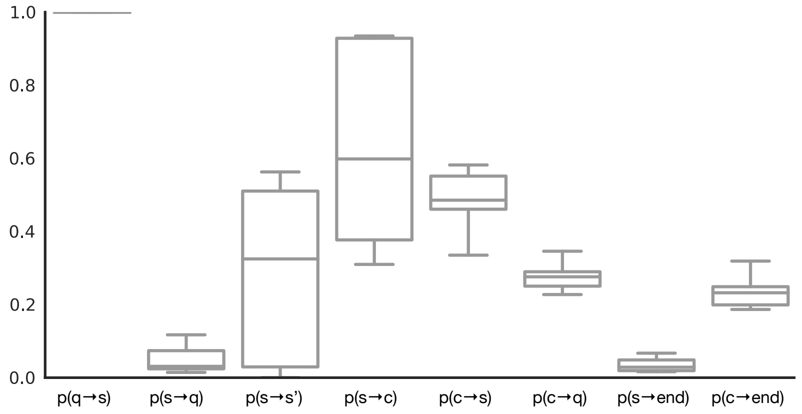


Figure 5.8: Transition probabilities

the clicking all user model such as the prefer first and prefer last user models, the probability $p(s \rightarrow c)$ is predefined because they use the clicking all click behavior. We can see that the user models with optimal clicking behavior (ideal and median user model) as well as our instance of the TREC user scan more results and click less.

5.3 Summary

In this chapter we revisited the general user model of Chapter 3 and introduced new user models, that differ in their search strategy and clicking behavior. The search strategy defines which results of the search session are viewed and the clicking behavior defines, when a result is clicked. For clicking behavior we are also using the spreading activation model, we introduced in Chapter 4. We then compared the user models with the TREC user in terms of information gain and cost effort. Our comparison shows the following results:

- The TREC user cumulates on average only 55% of the information gain that is accessible for the same cost.
- User models with the same clicking behavior correlate more in terms of information gain than user models with the same search strategy.
- The user model that correlates the most with the TREC user model is the one that combines the prefer first queries strategy with activation click behavior based on dynamic thresholding. However, the correlation is still smaller than the correlation between all other user models compared with each other.
- In the search sessions of the TREC session tracks there are more relevant results in the result list of the first queries than in the result list of the last ones.
- The TREC users invested on average 71% of the cost they would need to view all relevant results in the search session.
- The ideal user model, the median user model and the TREC user distribute their cost equally into clicking, scanning results and submitting queries. According to our Markov model, the other user models have a higher probability of clicking a result after scanning its snippet.

After describing, characterizing and comparing the user models, we will investigate how we can use them in order to evaluate rankings of result lists.

CHAPTER 6

Evaluating Search Sessions with User Models

In the last chapter we introduced a variety of user models that differ in terms of search strategy and clicking behavior and compared them with the TREC user. In this chapter we investigate how we can use these simulated instances of users in order to reason about the *quality* of the ranking of a result lists in a search session and to reason about how changes in the retrieval system effect the user behavior.

In Section 2.1 we described the current state of information retrieval evaluation. Most commonly used metrics like *normalized discounted cumulative gain* (nDCG) and *expected reciprocal rank* (ERR) estimate how much information gain a user cumulates with viewing the results of a result list. These evaluation metrics can be described in the following form: The expected cumulated information gain E can be calculated by summing up the relevance level $\mathtt{rel}$ of each result r in the result list multiplied with a discount d according to the result's rank:

$$E = \sum_{i=1}^{i=n} \mathtt{rel}(r_i) \cdot d(i) \quad .$$

The nDCG metric uses a logarithmic discount; the ERR metric uses a discount based on the number of relevant documents viewed before the rank i. In order to apply those evaluation metrics for search sessions, Järvelin et at. proposed to additionally discount the result's relevance level the more queries have been submitted before viewing the result [JPDN08]. As we pointed out, the problem with these evaluation metrics is that they are not based on a sophisticated user model.

In this chapter we introduce a metric based on our user models that also give an estimate of the information gain for a search session. Our metric is

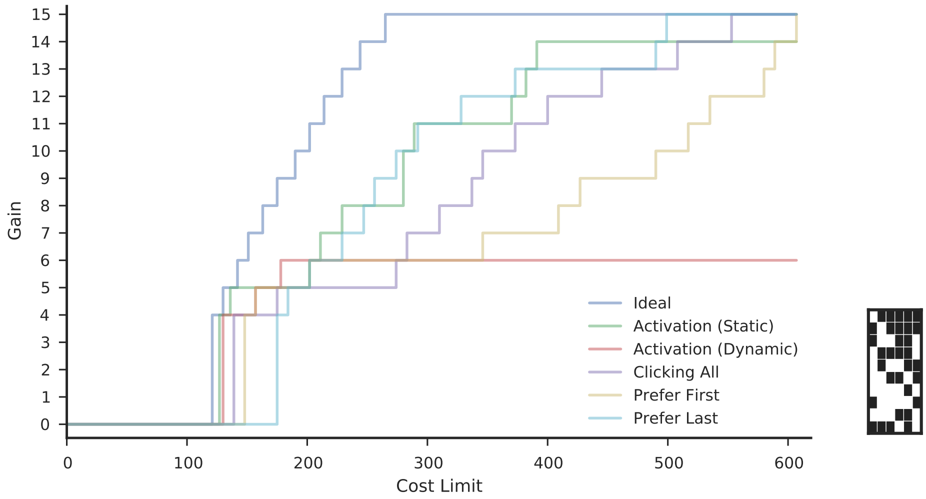


Figure 6.1: Cumulated information gain of different user models in respect to the cost limit.

based on the time-biased gain approach of Smucker and Clarke [SC12]. After we showed how to calculate the estimate, we evaluate the search sessions of the TREC Session Track and compare the result with the results of other evaluation metrics used in the course of the TREC Session Track competition.

6.1 Estimated Information Gain

The behavior of the user models introduced in this thesis are cost-driven. That means that they choose their actions in a way such that in total the cost of those actions do not exceed a given cost limit and such that they still cumulate as much information gain as possible. Therefore, the cumulated information gain is dependent on how much cost effort the user invests into the search session in order to solve the search task. Based on this, we can describe the cumulated information gain of a search session as a function of increasing cost limit: $\mathtt{Gain}(\mathtt{cost}_{\mathtt{max}})$.

In Figure 6.1 we can see the cost limit-gain curves of seven user models for one session of the TREC Session Track 2012. The session matrix on the right hand side shows the distribution of relevant document in this session. Note that in contrast to the cost-gain curves shown in Section 5.2.2, the curves do not show one path through the search session, they show the maximum gain the user models can cumulate for a certain cost model. Consequently, each point in a cost limit-gain curve represents one path through the search session.

With a cost-limit gain curve we can for instance describe the ideal user model. As we can see in the session matrix, the sample session consists of 6 result lists that contain 33 (15 distinct) relevant documents. For the ideal user model the shortest path through the search session consists of submitting all 6 queries and scanning the first result of each result list. Since this user model clicks every result that is relevant, and in the first rank there are four distinct relevant documents, the ideal user model experience an information gain of 4. With each further cost investment, the ideal user can view more results and therefore they can choose longer paths through the search session. As we can see in Figure 6.1, the ideal user can achieve an information gain for every small cost limit increment and at the cost limit of 250, the ideal user model chooses a path through the search session that leads to all relevant documents in the search session. The other user models need a higher cost limit in order to see all relevant documents, because they also click on non-relevant results or make imperfect stopping decisions. In this example, the activation user model with dynamic thresholding does not manage to click all relevant documents, because of its clicking behavior.

As we pointed out in Section 5.2.3, according to our general user model the users in the TREC Session Track did not invest all the cost effort needed in order to view and click all relevant results. Mostly, they were satisfied with the information gain they get from a smaller effort. In order to give an estimate on how much information gain a user model will cumulate in a search session, we need to take into account, how the users choose their cost limit.

Let $f(\mathrm{cost}_{max})$ be a probability density function that represents the likelihood of choosing a cost limit. This *cost limit likelihood function* is normalized such that the integral between the minimum and the maximum of the function equals to 1. Smucker and Clarke proposed to use this function f in order to estimate the cumulated information gain E of a session S as follows: [SC12]

$$E(S) = \int_0^\infty \mathrm{Gain}(S, \mathrm{cost}_{max}) \cdot f(\mathrm{cost}_{max})\ d\,\mathrm{cost}_{max} \quad .$$

In Figure 6.2 we show the probability density function we obtained from the observations of Section 5.2.3. We normalized the cost limit with the *maximum cost limit*, that is, the cost limit needed to scan and click all relevant

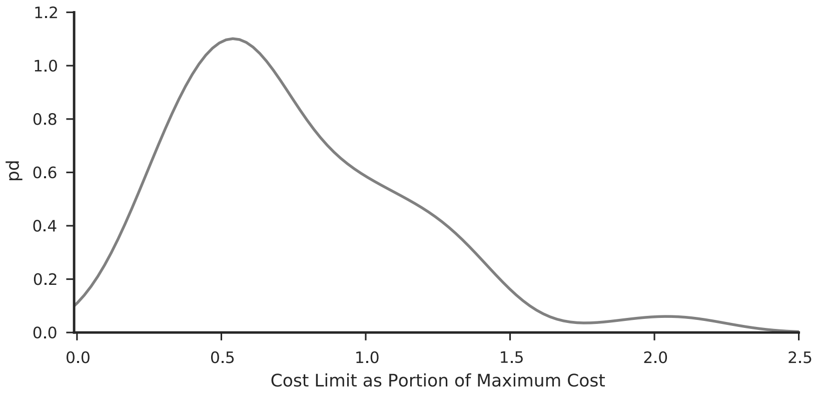


Figure 6.2: Cumulated information gain of different user models in respect to the cost limit.

results in a session. The maximum cost limit of a session S can be computed as follows:

$$\mathrm{maxcost}(S) = \mathrm{cost}_{\mathrm{scan}} \cdot |D| + \mathrm{cost}_{\mathrm{click}} \cdot |D_{\mathrm{rel}}| * \sum_{i=1}^{|S|} \mathrm{cost}_{\mathrm{query}} \cdot |q_i| \quad .$$

Where $|D|$ represents the number of documents in the session and $|D_{\mathrm{rel}}|$ represents the number of relevant documents in the session. We obtained the curve in Figure 6.2 with the help of kernel density estimation. In Appendix A.3 we show, how we can approximate this cost limit likelihood function alternatively with the help of an exponential Weibull distribution.

In order to calculate the estimated information gain of a search session for a certain user model, we sum up the gain and the likelihood of the cost limits between 0 and an upper cost bound. In the course of this thesis we set this upper cost bound to $2.5 \cdot \mathrm{maxcost}(S)$, since this is the highest cost limit a TREC user model needed in a search session (see Figure 6.2). As an increment incr for the sum we use the cost it takes in order to perform one snippet scan and one click. Consequently, we can compute the estimated gain E of a session S as follows:

$$\begin{aligned} E(S) &= \sum_{i=0} \mathrm{Gain}(S, \mathrm{incr}(i)) \cdot F(i) \\ F(i) &= \int_{\mathrm{incr}(i-1)}^{\mathrm{incr}(i)} f(\mathrm{cost}_{\mathrm{max}})\ d\,\mathrm{cost}_{\mathrm{max}} \\ \mathrm{incr}(i) &= i \cdot (\mathrm{cost}_{\mathrm{click}} + \mathrm{cost}_{\mathrm{scan}}) \quad . \end{aligned}$$

	Avg. Estimated	sDCG	sERR	MAP
Avg. Estimated	1.00	0.95	0.83	0.73
sDCG		1.00	0.91	0.81
sERR			1.00	0.79
MAP				1.00

Table 6.1: Correlation (Spearman's rho) between the average of the estimates of our user models and the traditional evaluation metrics sDCG, sERR and MAP.

In order to calculate the integral of the cost limit likelihood function f in one increment step i, we use the rectangle method in order to obtain an approximation that can be calculated efficiently.

6.2 Comparison with Evaluation Metrics

In this section, we compare the information gain estimation based on our user models with the established evaluation metrics expected reciprocal rank (ERR), the session version of discounted cumulative gain (sDCG) and mean average precision (MAP), which we introduced in detail in Section 2.1. In order to determine whether our information gain estimation is qualified as an evaluation metric or not, we compute the correlation with the established metrics. Although a high correlation does not proof the quality of our metric, a low or oppositional correlation would be unacceptable.

We compute for every session in the TREC Session Track 2011-2013 and for each of our user models the estimated information gain. We then calculate the average estimated information gain from those models for every search session. In order to apply the expected reciprocal rank for sessions (sERR), we summed up the ERR value for every result list in the session. For the discounted cumulated gain we used the unnormalized version with session discount (sDCG). Additionally we calculated the mean average precision for every result list.

Table 6.1 shows the correlation values between the average estimated information gain, sERR, sDCG and MAP; in Table A.4 in the Appendix we can find the correlation values for every user model. We used Spearman's rank correlation coefficient, since we do not require a linear relation between the metrics. We can see, that the sDCG metric correlates the most with our cumulative gain estimation. The MAP metric is the one that correlates the least with the other metrics, since it is the only normalized metric we are testing. Among our user models, the activation model with dynamic thresholding has the lowest correlation values.

All in all, we can see that our information gain estimation correlates with established evaluation metrics.

6.3 Summary

In this chapter we made a first attempt on formulating an evaluation metric based on our user models. Based on the time-biased gain approach of Smucker and Clarke, we describe a search session with the help of an estimate of the information gain the user cumulates for average cost effort [SC12]. We utilize the distribution of cost effort from the outcomes of Chapter 5 in order to obtain a likelihood function for cost limits. Lastly, we compared the estimated information gain values with evaluation metrics that also give an estimate on cumulated information gain. We found that our estimation highly correlates with the other unnormalized metrics.

All in all, the metric introduced in this chapter shows only one simple way to use the user models introduced in this thesis in order to obtain an estimate on cumulated information gain. For future work we could determine the sensitivity of our metric with the help of the *bootstrap hypothesis test* of T. Sakai [Sak06].

CHAPTER 7
Conclusion

In the last chapter of this thesis we shortly summarize the main outcomes of this thesis and give an outlook on future work.

7.1 Summary

In this thesis we built a framework that allows for simulating different instances of users that utilize different search strategies and clicking behaviors in order to make predictions on the cumulated information gain of a search session. The cumulated information gain represents how many distinct relevant documents the user views in a session and is dependent on how much effort the user is willing to invest in order to satisfy their information needs. We measure the user's effort by assigning costs to every action the user performs during a search session. Based on the cost effort, we can estimate the information gain of a search session for each user model.

We defined a general user model, that describes how users proceed in a search session. We assume that users read result lists in a top down fashion and they perform clicking decisions right after scanning the results. From this general user model we derived a set of concrete user models that differ in their search strategy, that defines what results in a session are scanned, and in their clicking behavior, which defines when to click on a scanned result. First of all, we investigated the model of an ideal user, that utilizes optimal click behavior and a high-information gain search strategy. This user model represents a user who found the perfect trade-off between action cost and information gain and therefore achieves the highest information gain possible for a given cost limit.

We obtained an alternative clicking behavior model from the field of cognitive modeling. With the help of the spreading activation model of information scent that utilizes an associative network of concepts, it is possible to calculate the utility of a result in respect to an information need in form of a task description. The activation based user models decide to click a result, when the activation level is above a certain threshold; we used two versions: static and dynamic thresholding. In addition to the ideal user and the activation based user model, we defined further user models that, for instance, prefer results of certain queries in a search session or that click every result that they encounter. Furthermore, we investigated the TREC user model, which we derived from the clicks in the interaction logs of the TREC Session Track competition.

In order to get an insight on how the different user models perform in a search session we compared them in terms of information gain and cost effort. The outcomes of this comparison include, that among the user models that do not use optimal clicking behavior the activation user model with static thresholding cumulates the most information gain. We found that the TREC user model viewed only about half of the relevant documents the search session provides. More surprising is the outcome, that in the search sessions most of the relevant documents can be found in the result lists of the first queries. This outcome states the opposite behavior in comparison to the *query bias assumption* of Zhang et al. [ZCWY11]. With the help of Markov models, we concluded that the TREC users and our user models with optimal clicking behavior click less than our other user models.

Lastly, we investigated how we can use our user models in order to estimate how much information gain a user cumulates in a search session. This estimate, that is based on the time-biased gain approach of Smucker and Clarke, is a first attempt of an evaluation metric which can be used as an alternative to the established metrics used in the course of the TREC competitions. Finally, we showed that our estimate correlates with the established metrics.

The main outcome of this thesis is a framework that allows for formulating deterministic user models with cost-driven behavior. We investigated eight instances and compared them with the user model obtained from the TREC data. This allowed us to draw some conclusions on the behavior and performance of the user models. We then showed how we can use the user models from our framework in order to evaluate rankings of search sessions. We believe that estimating the information gain with the help of user models of our framework can lead to a evaluation metric that is more transparent than the established evaluation metrics for two reasons: first, this metric works without any artificial information gain discount and second, we can reproduce for every instance of a user how it achieved a certain information gain. Therefore, we can evaluate the performance of a retrieval system and the influence of changes in rankings on different instances of users on a higher level of detail.

7.2 Future Work

In order to enforce the advantage of transparency of our user simulation framework, it is necessary to involve more meaningful user models. In addition to the rather abstract user models of this thesis, we could use the outcomes of user studies that investigate search behavior in order to instantiate other typical users. Nielsen et al. for instance investigated with the help of eye-tracking studies how users process web pages and came up with different types of users [Nie07]. With instantiating more user models, we can get a deeper insight on how the retrieval system of a search engine influences the search behavior of different user types.

In the course of this thesis we mostly worked with binary relevance. A result document can either be relevant to the users search intent or not. This binary relevance is an abstraction of the real process of information gain. Users gain more information from some results than form other results and omit relevant documents that contain information they already gained from other sources. A more precise model of information gain that could replace the Cranfield-style relevance level paradigm comes from Pavlu et al. [PRGA12]. They propose to use *information nuggets* in order to express the information content of a document. Involving information nuggets in our user simulation could not only make the process of cumulating information gain more realistic, we could also overcome the problem of handling duplicates in the search session. Furthermore, we could use information nuggets in order to model the user satisfaction that have an influence on the stopping behavior; a component our simulation is missing.

The cost model we used in the course of this thesis remained constant for all the experiments we performed. We obtained this cost model from an eye-tracking study of Tran and Fuhr, who measured the time the users need for certain search actions in a desktop environment [TF13]. However, users do not always sit in front of a desktop computer and do not necessarily use keyboard and mouse as input devices. For instance, users with touch based input devices need more time to type in query terms; therefore their query costs are higher. Azzopardi et al. found that, in fact, the cost of an action has an influence on the user's search behavior [AKB13]. With changing action costs we can simulate different environments.

In addition, we can extend our framework such that our models can also perform queries and query reformulations. Such a query simulation would need a model on how users chooses query terms and how users combine these in order to formulate and reformulate queries in a search session. Azzopardi et al. already came up with a model for simulating known-item queries and Dang and Croft propose a query simulation based on anchor texts [AdRB07, DC10]. With the help of such a query simulation we can simulate complete sessions based on search task descriptions and therefore we can obtain alternative data sets to the TREC Session Track data set.

Evaluating with user models instead of using traditional Cranfield-style evaluation metrics has still more advantages than we showed in the last chapter. In contrast to calculating one score for the rankings in a session, with a set of user models we can obtain a distribution of performances. In this thesis we only investigated the mean of this distribution: the average estimated information gain. When two rankings for one search session differ in their average estimated information gain, we assume that the ranking with the higher average estimated information gain is the better one. In addition to the changes of the mean value of the performance distribution, the changes in the variance could give us an insight on how many users are actually effected by the differences in the ranking. In other words, we can use the distribution of estimated information gain values, obtained form our user models, in order to measure the significance of the changes between two rankings. This is not possible with the traditional evaluation metrics. However, in order to demonstrate the evaluation of performance variance, we would need data sets that contain different rankings for a search sessions.

CHAPTER A
Appendix

A.1 Comments on the TREC Session Track Data

The TREC Session Track competition is part of the annual Text REtrieval Conference TREC. In the course of this competition the participants get a dataset consisting of several search sessions that users performed in order to solve search tasks. The search sessions of the data sets contain the following information:

- The complete text of each search task.
- Additional information on the task that was provided to the people who performed the relevance judgments.
- A sequence of queries that were submitted by the user.
- The corresponding result lists for all the queries except for the last one.
- Title and snippets for every result in the result list.
- Clicks on results by the user.
- The document corpus (2011-2012: ClueWeb09 corpus, 2013: ClueWeb12 corpus)

The participants of this competition then have the task to submit a suitable ranked result list for the last query of each session. The submitted result lists are then evaluated with the Cranfield-like evaluation metrics: normalized cumulative gain (nDCG), expected reciprocal rank (ERR) and mean average precision (ERR). Those metrics need relevance judgments for every result, which were performed by a number of experts who got the search task and with some additional information and then judged the documents of the submitted result lists (pooling).

Although in the pool of judged documents are only the documents of the submitted result lists, there are relevance judgments for all documents in the provided search sessions. The only sessions, we have no relevance judgments for are the sessions of the TREC Session Track 2013 with the following topic IDs:

$$3, 4, 7, 9, 10, 19, 20, 27, 46, 56, 59, 68$$

A.2 Comparison of Information Gain Distributions

	TREC	Ideal	Median	Act. (St.)	Act. (Dyn.)	Clicking all	Pref. First	Pref. Last
TREC	1.00	0.00	0.00	0.10	0.00	0.60	0.00	0.00
Ideal	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00
Median	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00
Act. (St.)	0.10	0.00	0.00	1.00	0.00	0.39	0.00	0.00
Act. (Dyn.)	0.00	0.00	0.00	0.00	1.00	0.00	0.59	0.01
Clicking all	0.60	0.00	0.00	0.39	0.00	1.00	0.00	0.00
Pref. First	0.00	0.00	0.00	0.00	0.59	0.00	1.00	0.01
Pref. Last	0.00	0.00	0.00	0.00	0.01	0.00	0.01	1.00

Table A.1: P-values of the Wilcoxon signed-rank test for determining the significance levels of the differences of the information gain distributions. We use four significance levels: $p \leq 0.10$ semi-strong significance, $p \leq 0.05$ strong significance, and $p \leq 0.01$ very strong significance.

	Name	Abbreviation
Search Strategies	Highest Gain	HG
	Medium Gain	MG
	Prefer First Queries	PF
	Prefer Last Queries	PL
Click Behavior	Optimal Clicking	OC
	Activation (Static)	AS
	Activation (Dynamic)	AD
	Clicking All	CA

Table A.3: Abbreviation of click behaviors and search strategies used in Table A.2.

	HG - OC	TREC	HG - AS	HG - AD	HG - CA	PF - CA	PL - CA	MG - AS	PF - AS	PL - AS	MG - AD	PF - AD	PL - AD	MG - OC	PF - OC	PL - OC
HG - OC	1.00	0.68	0.88	0.67	0.79	0.80	0.72	0.77	0.87	0.80	0.60	0.64	0.70	0.95	0.98	0.87
TREC	0.68	1.00	0.73	0.71	0.69	0.58	0.64	0.75	0.70	0.74	0.81	0.79	0.79	0.73	0.68	0.67
HG - AS	0.88	0.73	1.00	0.80	0.84	0.78	0.68	0.87	0.99	0.87	0.75	0.78	0.82	0.85	0.87	0.82
HG - AD	0.67	0.71	0.80	1.00	0.79	0.71	0.75	0.83	0.78	0.73	0.92	0.94	0.92	0.72	0.68	0.67
HG - CA	0.79	0.69	0.84	0.79	1.00	0.92	0.83	0.79	0.83	0.75	0.76	0.77	0.80	0.81	0.79	0.72
PF - CA	0.80	0.58	0.78	0.71	0.92	1.00	0.89	0.69	0.78	0.69	0.67	0.70	0.72	0.81	0.80	0.71
PL - CA	0.72	0.64	0.68	0.75	0.83	0.89	1.00	0.71	0.68	0.70	0.71	0.73	0.75	0.74	0.71	0.72
MG - AS	0.77	0.75	0.87	0.83	0.79	0.69	0.71	1.00	0.87	0.79	0.79	0.82	0.79	0.83	0.79	0.71
PF - AS	0.87	0.70	0.99	0.78	0.83	0.78	0.68	0.87	1.00	0.90	0.74	0.77	0.81	0.85	0.87	0.84
PL - AS	0.80	0.74	0.87	0.73	0.75	0.69	0.70	0.79	0.90	1.00	0.71	0.73	0.82	0.77	0.80	0.89
MG - AD	0.60	0.81	0.75	0.92	0.76	0.67	0.71	0.79	0.74	0.71	1.00	0.98	0.92	0.66	0.62	0.62
PF - AD	0.64	0.79	0.78	0.94	0.77	0.70	0.73	0.82	0.77	0.73	0.98	1.00	0.91	0.68	0.65	0.66
PL - AD	0.70	0.79	0.82	0.92	0.80	0.72	0.75	0.79	0.81	0.82	0.92	0.91	1.00	0.75	0.72	0.76
MG - OC	0.95	0.73	0.85	0.72	0.81	0.81	0.74	0.83	0.85	0.77	0.66	0.68	0.75	1.00	0.96	0.79
PF - OC	0.98	0.68	0.87	0.68	0.79	0.80	0.71	0.79	0.87	0.80	0.62	0.65	0.72	0.96	1.00	0.84
PL - OC	0.87	0.67	0.82	0.67	0.72	0.71	0.72	0.71	0.84	0.89	0.62	0.66	0.76	0.79	0.84	1.00

Table A.2: Correlation of information gain between all 16 user models.

	Ideal	Act. (St.)	Act. (Dyn.)	Click. All	Pref. First	Pref. Last	Avg.	sDCG	sERR
Ideal	1.00	0.91	0.73	0.91	0.94	0.92	0.99	0.94	0.82
Act. (St.)	0.91	1.00	0.72	1.00	0.92	0.86	0.94	0.86	0.71
Act. (Dyn.)	0.73	0.72	1.00	0.72	0.72	0.75	0.78	0.73	0.64
Click. All	0.91	1.00	0.72	1.00	0.92	0.86	0.94	0.86	0.71
Pref. First	0.94	0.92	0.72	0.92	1.00	0.91	0.96	0.93	0.77
Pref. Last	0.92	0.86	0.75	0.86	0.91	1.00	0.93	0.87	0.76
Avg.	0.99	0.94	0.78	0.94	0.96	0.93	1.00	0.95	0.83
sDCG	0.94	0.86	0.73	0.86	0.93	0.87	0.95	1.00	0.91
sERR	0.82	0.71	0.64	0.71	0.77	0.76	0.83	0.91	1.00

Table A.4: Correlation of estimated information gain of our user models with the not normalized metrics session DCG and session ERR (Spearmans ρ, $p < 0.1$).

A.3 Approximation of the Cost Limit Likelihood Function

According to the Wilcoxon-Smirnoff test, this function follows with a probability of 0.9 an exponential Willbull distribution with the form:

$$f(x) = a \cdot c \cdot (1 - e^{-x})^{a-1} \cdot e^{-x^c} \cdot x^{c-1} \quad .$$

With the help of the statistical module of the SciPy framework[1] we performed a fit on the cost limits the TREC user model used in the TREC session tracks 2011-2013 and obtained following shape parameters:

$$\begin{aligned} a &= 1.1491 \\ c &= 1.3302 \quad . \end{aligned}$$

We can use this function in order to approximate the cost limit likelihood function.

[1] http://docs.scipy.org/doc/scipy/reference/stats.html

Bibliography

[AAS09] Ahmed Sabbir Arif, Ahmed Sabbir Arif, and Wolfgang" Stuerzlinger. Analysis of text entry performance metrics. *IN PROC. IEEE TIC-STH 2009. IEEE*, pages 100–105, 2009. 5.2.2

[AdRB07] Leif Azzopardi, Maarten de Rijke, and Krisztian Balog. Building simulated queries for known-item topics: an analysis using six european languages. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '07, pages 455–462, New York, NY, USA, 2007. ACM. 7.2

[AKB13] Leif Azzopardi, Diane Kelly, and Kathy Brennan. How query cost affects search behavior. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '13, pages 23–32, New York, NY, USA, 2013. ACM. 7.2

[AML97] John R. Anderson, Michael Matessa, and Christian Lebiere. ACT-R: A theory of higher level cognition and its relation to visual attention. *Hum.-Comput. Interact.*, 12(4):439–462, December 1997. 4.3

[BC00] Ken Barker and Nadia Cornacchia. Using noun phrase heads to extract document keyphrases. In *Proceedings of the 13th Biennial Conference of the Canadian Society on Computational Studies of Intelligence: Advances in Artificial Intelligence*, AI '00, pages 40–52, London, UK, UK, 2000. Springer-Verlag. 4.3.1

[BD09] Jerome R. Busemeyer and Adele Diederich. *Cognitive Modeling*. SAGE Publications, Inc, 2009. 4.1

[BKJ12] Feza Baskaya, Heikki Keskustalo, and Kalervo Järvelin. Time drives interaction: simulating sessions in diverse searching environments. In *Proceedings of the 35th international ACM SIGIR* *conference on Research and development in information retrieval*, SIGIR '12, pages 105–114, New York, NY, USA, 2012. ACM. 3.2

[BRP07] Raluca Budiu, Christiaan Royer, and Peter Pirolli. Modeling information scent: A comparison of LSA, PMI and GLSA similarity measures on common tests and corpora. In *Large Scale Semantic Access to Content (Text, Image, Video, and Sound)*, RIAO '07, pages 314–332, Paris, France, France, 2007. Le Centre de Hautes Études Internationales d'Informatique Documentaire. 4.3.2

[CH90] Kenneth Ward Church and Patrick Hanks. Word association norms, mutual information, and lexicography. *Comput. Linguist.*, 16(1):22–29, March 1990. 4.3.2

[CMZG09] Olivier Chapelle, Donald Metlzer, Ya Zhang, and Pierre Grinspan. Expected reciprocal rank for graded relevance. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, CIKM '09, pages 621–630, New York, NY, USA, 2009. ACM. 2.1

[CZ09] Olivier Chapelle and Ya Zhang. A dynamic bayesian network click model for web search ranking. In *Proceedings of the 18th International Conference on World Wide Web*, WWW '09, pages 1–10, New York, NY, USA, 2009. ACM. 2.2

[DC10] Van Dang and Bruce W. Croft. Query reformulation using anchor text. In *Proceedings of the third ACM international conference on Web search and data mining*, WSDM '10, pages 41–50, New York, NY, USA, 2010. ACM. 7.2

[DDH07] Doug Downey, Susan Dumais, and Eric Horvitz. Models of searching and browsing: Languages, studies, and applications. In *Proceedings of the 20th International Joint Conference on Artifical Intelligence*, IJCAI'07, pages 2740–2747, San Francisco, CA, USA, 2007. Morgan Kaufmann Publishers Inc. 2.2

[DKW84] M.E. Dyer, N. Kayal, and J. Walker. A branch and bound algorithm for solving the multiple-choice knapsack problem. *Journal of Computational and Applied Mathematics*, 11(2):231 – 249, 1984. 3.3

[DP08] Georges E. Dupret and Benjamin Piwowarski. A user browsing model to predict search engine click data from past observations. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '08, pages 331–338, New York, NY, USA, 2008. ACM. 2.2

[FP07] Wai-Tat Fu and Peter Pirolli. SNIF-ACT: A cognitive model of user navigation on the world wide web. *Hum.-Comput. Interact.*, 22(4):355–412, November 2007. 2.2, 4.2, 4.3.2, 4.3.2

[HGBS13] Matthias Hagen, Jakob Gomoll, Anna Beyer, and Benno Stein. From search session detection to search mission detection. In *Proceedings of the 10th Conference on Open Research Areas in Information Retrieval*, OAIR '13, pages 85–92, Paris, France, France, 2013. Le Centre de Hautes Études Internationales d'Informatique Documentaire. 3.1

[HHdV12] Vera Hollink, Jiyin He, and Arjen de Vries. Explaining query modifications: An alternative interpretation of term addition and removal. In *Proceedings of the 34th European Conference on Advances in Information Retrieval*, ECIR'12, pages 1–12, Berlin, Heidelberg, 2012. Springer-Verlag. 3.1

[HW11] Yin He and Kuansan Wang. Inferring search behaviors using partially observable markov model with duration (POMD). In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 415–424. ACM, 2011. 5.2.3

[JK02] Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of IR techniques. *ACM Trans. Inf. Syst.*, 20(4):422–446, October 2002. 2.1

[Jon81] Karen Sparck Jones. *Information retrieval experiment*. Butterworth-Heinemann, 1981. 2.1

[JPDN08] Kalervo Järvelin, SusanL. Price, LoisM.L. Delcambre, and MarianneLykke Nielsen. Discounted cumulated gain based evaluation of multiple-query IR sessions. In *Advances in Information Retrieval*, volume 4956 of *Lecture Notes in Computer Science*, pages 4–15. Springer Berlin Heidelberg, 2008. 2.1, 5.1.2, 6

[Kay98] Steven Kay. *Fundamentals of Statistical Signal Processing, Volume II: Detection Theory*. Prentice Hall, 1998. 4.3.3

[KOS08] Mark T. Keane, Maeve O'Brien, and Barry Smyth. Are people biased in their use of search engines? *Commun. ACM*, 51(2):49–52, February 2008. 4.3.3

[KPP04] Hans Kellerer, Ulrich Pferschy, and David Pisinger. *Knapsack problems*. Springer, 2004. 3.3

[KWJ04] Kerstin Klöckner, Nadine Wirschum, and Anthony Jameson. Depth- and breadth-first processing of search result lists. In *CHI '04 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '04, pages 1539–1539, New York, NY, USA, 2004. ACM. 3.2.1

[Man99] K. I. Manktelow. *Reasoning and thinking*. Psychology Press, East Sussex, UK, 1999. 4.3.3

[MM83] Fritz Machlup and Una Mansfield, editors. *The Study of Information: Interdisciplinary Messages*. John Wiley & Sons, Inc., New York, NY, USA, 1983. 4.2

[MRS08] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008. 2.1

[Nie07] Jakob Nielsen. Fancy formatting, fancy words = looks like a promotion = ignored, September 2007. 7.2

[OK07] Maeve O'Brien and Mark T. Keane. Modeling user behavior using a search-engine. In *Proceedings of the 12th international conference on Intelligent user interfaces*, IUI '07, pages 357–360, New York, NY, USA, 2007. ACM. 2.2, 3.2.1

[PC95] Peter Pirolli and Stuart Card. Information foraging in information access environments. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '95, pages 51–58, New York, NY, USA, 1995. ACM Press/Addison-Wesley Publishing Co. 3.2.1, 4.2

[Pis95] David Pisinger. A minimal algorithm for the multiple-choice knapsack problem. *European Journal of Operational Research*, 83(2):394 – 410, 1995. {EURO} Summer Institute Combinatorial Optimization. 3.3

[PRGA12] Virgil Pavlu, Shahzad Rajput, Peter B. Golbus, and Javed A. Aslam. Ir system evaluation using nugget-based test collections. In *Proceedings of the Fifth ACM International Conference on Web Search and Data Mining*, WSDM '12, pages 393–402, New York, NY, USA, 2012. ACM. 7.2

[Rob08] Stephen Robertson. A new interpretation of average precision. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '08, pages 689–690, New York, NY, USA, 2008. ACM. 2.1

[Sak06] Tetsuya Sakai. Evaluating evaluation metrics based on the bootstrap. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '06, pages 525–532, New York, NY, USA, 2006. ACM. 6.3

[SC12] Mark D. Smucker and Charles L. A. Clarke. Modeling user variance in time-biased gain. In *Proceedings of the Symposium on Human-Computer Interaction and Information Retrieval*, HCIR '12, pages 3:1–3:10, New York, NY, USA, 2012. ACM. 2.1, 6, 6.1, 6.3

[Sib73] Robin Sibson. Slink: an optimally efficient algorithm for the single-link cluster method. *The Computer Journal*, 16(1):30–34, 1973. 5.2.2

[Sil86] B. W. Silverman. *Density Estimation for Statistics and Data Analysis*. Chapman and Hall, 1986. 4.3.3

[SJ10] Mark D. Smucker and Chandra Prakash Jethani. Human performance and retrieval precision revisited. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '10, pages 595–602, New York, NY, USA, 2010. ACM. 2.1

[SPCK10] Mark Sanderson, Monica Lestari Paramita, Paul Clough, and Evangelos Kanoulas. Do user preferences and evaluation measures line up? In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '10, pages 555–562, New York, NY, USA, 2010. ACM. 2.1

[TF12] Vu Tuan Tran and Norbert Fuhr. Using eye-tracking with dynamic areas of interest for analyzing interactive information retrieval. In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '12, pages 1165–1166, New York, NY, USA, 2012. ACM. 5.2.2

[TF13] Vu T. Tran and Norbert Fuhr. Markov modeling for user interaction in retrieval. In *SIGIR 2013 Workshop on Modeling User Behavior for Information Retrieval Evaluation (MUBE 2013)*, August 2013. 5.2.3, 7.2

[TH01] Andrew H. Turpin and William Hersh. Why batch and user evaluations do not give the same results. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '01, pages 225–231, New York, NY, USA, 2001. ACM. 2.1

[TS06] Andrew Turpin and Falk Scholer. User performance versus precision measures for simple search tasks. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '06, pages 11–18, New York, NY, USA, 2006. ACM. 2.1

[Voo02] Ellen M. Voorhees. The philosophy of information retrieval evaluation. In *Revised Papers from the Second Workshop of the Cross-Language Evaluation Forum on Evaluation of Cross-Language Information Retrieval Systems*, CLEF '01, pages 355–370, London, UK, UK, 2002. Springer-Verlag. 2.1

[XJP$^+$10] Biao Xiang, Daxin Jiang, Jian Pei, Xiaohui Sun, Enhong Chen, and Hang Li. Context-aware ranking in web search. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '10, pages 451–458, New York, NY, USA, 2010. ACM. 3.1

[ZCWY11] Yuchen Zhang, Weizhu Chen, Dong Wang, and Qiang Yang. User-click modeling for understanding and predicting search-behavior. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '11, pages 1388–1396, New York, NY, USA, 2011. ACM. 2.2, 5.1.2, 7.1

[Zhu04] Mu Zhu. Recall, precision and average precision. *Department of Statistics and Actuarial Science, University of Waterloo, Waterloo*, 2004. 2.1

List of Tables

List of Figures