Time Drives Interaction: Simulating Sessions in Diverse Searching Environments

Feza Baskaya, Heikki Keskustalo, Kalervo Järvelin
School of Information Sciences
FI-33014
University of Tampere, Finland
{ Feza.Baskaya, Heikki.Keskustalo, Kalervo.Jarvelin }@uta.fi

ABSTRACT

Real life information retrieval takes place in sessions, where users search by iterating between various cognitive, perceptual and motor subtasks through an interactive interface. The sessions may follow diverse strategies, which, together with the interface characteristics, affect user effort (cost), experience and session effectiveness. In this paper we propose a pragmatic evaluation approach based on scenarios with explicit subtask costs. We study the limits of effectiveness of diverse interactive searching strategies in two searching environments (the scenarios) under overall cost constraints. This is based on a comprehensive simulation of 20 million sessions in each scenario. We analyze the effectiveness of the session strategies over time, and the properties of the most and the least effective sessions in each case. Furthermore, we will also contrast the proposed evaluation approach with the traditional one, rank based evaluation, and show how the latter may hide essential factors that affect users' performance and satisfaction - and gives even counter-intuitive results.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Search process

Keywords

Session-based evaluation, simulation, time-based evaluation

1. INTRODUCTION

Interaction through search interface and environment greatly affects the user behavior, user experience, and user performance.

Many earlier studies have extended the traditional Cranfield view of IR and discussed various aspects of interactive searching (see, e.g., [4], [5], [6], [13], [21]), user interaction, and query modification (see, e.g., [3], [10], [14], [28]).

During interaction the user selects between *subtasks*, e.g., whether to scan the result or launch a new query instead, and how to construct the query. Such selections obviously affect session gains. However, different subtasks also have costs, e.g., they take time. This is important because real life IR often takes place under (time) constraints. In particular, keeping the overall session cost reasonable may be essential for end users.

SIGIR'12, August 12-16, 2012, Portland, Oregon, USA.

The costs of subtasks may vary for many reasons between searching environments. For example, regarding the query side, small devices and touch screens are inconvenient for typing [11]. Recently, novel kinds of searching devices, including personal phone-based mobile devices, have become increasingly popular.

In order to minimize the overall session costs, a mobile phone user might e.g., avoid typing and prefer result scanning. Low input costs might change the situation from the user's point of view, leading to longer queries. Therefore, if we assume two users having identical needs and identical cost constraints regarding the overall session time, it is possible that different devices render different subtask combinations optimal in searching.

Traditional IR evaluation focuses on the quality of the ranked output. In this view, the costs of posing queries are non-problematic, even uninteresting. In this paper we will utilize simple scenarios to bring time factors into the research setting. Scenarios formalize and quantify the gains and costs of interactive sessions. We construct two cases – a personal desktop computer (PC) and a smart phone (SP) case, with subtask costs derived from the literature. We will simulate session interaction involving multiple queries based on prototypical but empirically grounded query modification strategies using a test collection. We then explore the effectiveness of searching via the exhaustive set of querying-scanning combinations possible, and evaluate the effectiveness of both scenarios in terms of Cumulated Gain (CG) [16] under time constraint (overall session time). We use non-normalized metrics, because normalized metrics may yield misleading results, especially if time is taken into account.

Early papers on IR evaluation had a comprehensive approach to interactive IR evaluation. Cleverdon et al. [8] pointed out, among others, physical and intellectual user effort as an important factor in IR evaluation. Salton [24] identified user effort measures in the context of IR evaluation. More recently Su [30] gave a comparison of 20 different evaluation measures for interactive IR, including actual cost of search, several utility measures, and worth of search results vs. time expended. The interactive aspect of IR requires attention because previous studies have repeatedly shown that discrepancy exists between interactive and non-interactive evaluation results. Hersh et al. [12] showed that a weighting scheme giving maximum improvement over the baseline in non-interactive batch evaluation failed to surpass others when real users performed a simulated task. Turpin and Hersh [31] observed that a system superior over the baseline in batch evaluation, measured by mean

average precision, was not superior in an interactive situation. Turpin and Scholer [32] found no significant relationship between the search engine effectiveness measured by mean average precision and real user success in a precision-oriented task. Smith and Kantor [25] observed that users of degraded systems were as successful as those using non-degraded systems. They suggested that users achieved this by altering their behavior.

Dunlop [9] proposed "time-to-view" graphs, which incorporate user interface and system as well as the time component into the same framework for evaluation of system effectiveness. However he did not analyze time constraints, query modification strategies and different devices.

Smucker [26] brought time factors into the traditional Cranfield setting by augmenting it with the use of the GOMS [7] model (acronym for Goals, Operators, Methods, and Selections). He suggests a user model for IR where the search process is seen as a sequence of actions (e.g., typing; clicking; evaluating a summary; waiting for the results to load) with associated times and probabilities (e.g., whether the simulated user will click on a relevant summary). He used the model in a simulated study to demonstrate the impact of changes in the IR system interface (e.g., when the speed and accuracy of the summary evaluation is varied) on user performance (the number of relevant documents read within a given time frame). While his experiment was limited to single query situations, the approach can be extended to multiple query scenarios, e.g., for computing the costs of specific query reformulations.

Azzopardi [2] addressed the cost aspect by treating interactive IR as an *economical* problem and studied the trade-off between querying and browsing while maintaining a given level of normalized CG (NCG) [15] in sessions. His analysis focused on querying – scanning depth combinations for various formal retrieval methods that deliver a given level of NCG.

Our approach in the present paper differs from earlier studies. Our study is based on the simulation of multiple-query sessions generated with various query modification and scanning strategies in different searching environments.

In the next section we start by discussing session generation with costs, and present the research questions. In Section 3 we describe the research setting. In Section 4 we will run an experiment in a test collection based on scenarios and discuss the results. We close the paper by discussing the significance of our approach in the last section.

2. CONSTRUCTION OF SESSIONS

A *use case* is "a relatively informal description of system's behavior and usage, intended to capture the functional requirements of the system by describing the interaction between the outside actors and the system, to reach the goal of the primary actor" [19]. We utilize simplified use cases, which we call scenarios, to present an alternative way to look at the effectiveness of IR approaches based on the user viewpoint. The next subsections will first explain the session generation formally, and then explain the specific query modification (QM) and scanning strategies utilized in the scenarios.

2.1 Session generation

For session simulation, we first formally generate all possible sessions under constraints. We will represent sessions as sequences of actions with costs. For example the tuple $<(a_1,c_1), (a_2,c_2), \ldots, (a_n,c_n)>$ is a session of n actions and each pair (a_i,c_i) in the session representation represents an action a_i and its cost c_i. The elementary action types are:

- Initial query, represented as ('iq', ic), where 'iq' is the action label and ic ($\in$R) the cost in seconds.
- Query reformulation ('q', qc), where 'q' is the action label and qc ($\in$R) the cost in seconds.
- Document snippet scan ('s', sc), where 's' is the action label and sc ($\in$R) the cost in seconds.
- Next page request ('n', nc), where 'n' is the action label and nc ($\in$R) the cost in seconds.

The constraints are:

- MaxSLen, maximum session length in terms of elementary actions, here 50 actions.
- MaxSCost, maximum session cost (seconds), here 60, 90 or 120 seconds.
- A session always begins with an initial query.
- All queries (initial and reformulation) are followed by at least one snippet scan.
- The longest scan sequence we consider is a scan of 10 snippets (i.e. one typical result page).

In effect, the shortest possible session therefore is initial action IA = <(iq, ic),(s, sc)>, consisting of an initial query followed by the scan of one snippet (with costs). To generate longer sessions, we define the set NA for the possible subsequent actions:

$$NA = \{<(q, qc), (s, sc)>, <(s, sc)>, <(n,nc), (s,sc)>\}$$

Note here that the next actions are tuples of one or two elementary actions; a scan may appear individually, while a reformulation / next page requires a scan to follow. Sessions are generated by concatenating next actions to the initial action. Concatenation of two tuples $S_1 = <e_1, e_2, \ldots, e_n>$ and $S_2 = <f_1, f_2, \ldots, f_m>$ is denoted by $< S_1, S_2 > = <e_1, e_2, \ldots, e_n, f_1, f_2, \ldots, f_m>$. This operation generalizes over a set of session tuples S_i, denoted as:

$$\bigtimes_{i=1\ldots n} S_i = <<\ldots <<S_1, S_2>, S_3>, \ldots>, S_n>.$$

The cost of a session S is, informally, the sum of its action costs. More formally, we derive this cost by the function *s-cost* as follows:

$$s\text{-}cost(S) = \sum_{(a,c) \in S} c$$

[N.B. we extend the definition of the set membership operator from sets to tuple components in an obvious way.] For example, the cost of the session S1 = <('iq', ic),('s', sc), ('q', qc),('s', sc)> is *s-cost*(S1) = ic+sc+qc+sc.

The condition of maximum scan length of n in a session S is enforced by the Boolean predicate *max-scan*(S, n). It yields 'true' for a given session S if S does not contain a subsequence of scan actions $<('s', sc)_1, ('s', sc)_2, \ldots, ('s', sc)_n>$, otherwise 'false' (formal definition here omitted for brevity).

To generate sessions, we first generate all sessions up to the max number of actions MaxSLen. This session set is MLS:

$$MLS = \bigcup_{i=1\ldots MaxSLen} \{<IA, \bigtimes_{j=1\ldots i} ac_j > \mid ac_j \in NA\}$$

We then select the subset of sessions fulfilling the time constraint MaxSCost and the scan length constraint as follows. All sessions in MLS with maximal cost MaxSCost (or less) form the set MCS:

$$MCS = \{S \in MLS \mid s\text{-}cost(S) \leq MaxSCost \wedge max\text{-}scan(S, 11)\}$$

Note that this approach does not define the query contents or modifications in sessions. However, it keeps them within constraints and guarantees that the last action is a document snippet scan. In our experiments, we excluded the next page action from NA due to the max scan length constraint of 10. The next two sub-sections explain and justify the query modification and scanning strategies used in the experiment.

2.2 Query Modification Strategies

We will simulate interactive search sessions as querying-scanning iterations having a goal, a procedure to reach the goal, and constraints regarding the procedure. We define the goal in terms of maximizing CG during the session under the constraint on the overall session time available. The procedure is defined in terms of QM and scanning strategies.

The previous section did not define any particular QM strategies. We assume that a set of individual words $\{w_1, w_2, w_3, w_4, w_5\}$ is available for each particular topic, and QM strategies determine how elements from this set are combined to form queries (either the initial query, or one of the subsequent queries). In other words, given a set of individual search words for the topic, the QM strategy defines how to form a sequence of queries.

Five QM strategies (S1 – S5) were used in the experiment. These prototypical strategies are based on term level changes which have grounding in the observed real life behavior and are justified by literature (see [1], [20], [33]):

- **S1**: an initial one-word query (w_1) is followed by repeatedly varying the search word :
  Q_1: w_1 -> Q_2: w_2 ->Q_3: w_3 ->Q_4: w_4 ->Q_5: w_5
- **S2**: an initial two-word query ($w_1 w_2$) is followed by queries formed by repeatedly varying the second word :
  Q_1: w_1 w_2 -> Q_2: w_1 w_3 -> Q_3: w_1 w_4 -> Q_4: w_1 w_5
- **S3**: an initial three-word query ($w_1 w_2 w_3$) is followed by queries formed by repeatedly varying the third word :
  Q_1: w_1 w_2 w_3 -> Q_2: w_1 w_2 w_4 -> Q_3: w_1 w_2 w_5
- **S4**: an initial one-word query (w_1) is followed by adding one word to each subsequent query :
  Q_1: w_1 -> Q_2: w_1 w_2 -> Q_3:w_1 w_2 w_3 -> Q_4:w_1 w_2 w_3 w_4 -> ...
- **S5**: an initial two-word query ($w_1 w_2$) is followed by adding one word to each subsequent query :
  Q_1: w_1 w_2 -> Q_2: w_1 w_2 w_3 -> Q_3: w_1 w_2 w_3 w_4 -> ...

This means that the sessions consist of at most 3 to 5 queries; this reflects real life behavior [22]. Generally speaking, constructing a query entails a cost due to the cognitive user load plus the edit costs. We will return to the cost factors in Section 2.4.

2.3 Scanning Strategies

The user may simply scan one or more documents after each query before formulating the next query candidate or ending the session. After a *single* query Q_i a sequence of one or more document snippets may be scanned:

Q_1->s_{11}->s_{12}->s_{13}->...

The cost of this session manifests as:

$qc_1 + sc_{11} + sc_{12} + sc_{13} + ...$

When a *set* of queries is available for one topic, the user can scan varying numbers of document snippets after any particular query, leading to a vast number of possible querying-scanning *sessions*, e.g.,

Q_1->s_{11}->Q_2->s_{21}->Q_3->s_{31}-> ... or

Q_1->s_{11}->s_{12}->Q_2->s_{21}->... or

Q_1->s_{11}->s_{12}->s_{13}->Q_2->s_{21}-> s_{22}->Q_3->s_{31}->... etc.

In real life a session typically continues until the user has found what he was looking for, at least partially, and/or when he runs out of time or queries. The scanning lengths may fluctuate for many reasons. In this paper we study the properties of optimal and less optimal interactive behaviors in sessions below the given overall time constraint. Therefore we produced *all* possible sessions as follows. For all five QM strategies we formed all possible combinations of scanning lengths exhaustively (from 1 to 10 documents) using a sequence of all possible queries available per topic (cf. equation MCS in Section 2.1). We focus on the top documents because only few top documents may be inspected by the user in real life [14], [23], and only these may matter for the user [1]. As we had 5 words for each topic, sessions had at most 5 queries, controlled by the QM strategy and time constraint. As the query words were ordered by quality (see 3.1), the query words were used in that particular order, not permuted.

2.4 Cost Factors

There is a cost involved with the subtasks of formulating the query and scanning. We assume that the *absolute cost* is partially determined by the scenario. Empirical studies show that it takes significantly more time to enter queries by using a small smart phone keypad than by using an ordinary keyboard [17]. To study the significance of subtask costs under overall session cost constraint we define two scenarios, i.e., a Desktop PC scenario (PC) and a Smart phone scenario (SP). These scenarios have different subtask costs. This is justified because the properties of the devices partially determine the subtask costs [17].

Obviously, also forming queries under different QM strategies S1 – S5 have very different *relative costs*. All queries in strategies S1, S2 and S3 have a fixed query length in sessions (one, two or three words, correspondingly) while in strategies S4 and S5 the queries grow longer. In real life the typing speed is affected by, e.g., the experience and knowledge of the person, the size of the keyboard, the layout of the keyboard (e.g., nine-key multi-tap vs. qwerty keyboard) [17], [18], and whether predictive text feed is available and used. We used literature to derive the cost values in scenarios PC and SP regarding the initial query cost and the subsequent query cost (Table 1).The query costs in S1 – S5 in the Desktop PC case are based on the typing costs of 3.0 seconds per word. The corresponding Smart Phone costs are based on [17]. The authors performed a large-scale log analysis of cell phone usage and observed that an average smart phone query length was 2.56 words and the average query-entry time was 39.8 seconds (average typing cost of 15.5 seconds per word). We assume in our simulations that the cost of *adding* one word to a query (that is, S4 and S5) or *replacing* one word at the end of the previous query (that is, S1, S2, S3) is a constant, i.e., either 3.0 or 15.5 seconds depending on the scenario.

Table 1. Average subtask costs (in seconds) of five QM strategies (S1-S5) for two scenarios: (i) initial query cost, (ii) subsequent query cost, and (iii) the cost of scanning one document snippet

Scenario 1: Desktop PC					
QM strategy	**S1**	**S2**	**S3**	**S4**	**S5**
Initial query	3.0	6.0	9.0	3.0	6.0
Subsequent query	3.0	3.0	3.0	3.0	3.0
Snip. scanning cost	3.0	3.0	3.0	3.0	3.0
Scenario 2: Smart Phone					
QM Strategy	**S1**	**S2**	**S3**	**S4**	**S5**
Initial query	15.5	31.0	46.5	15.5	31.0
Subsequent query	15.5	15.5	15.5	15.5	15.5
Snip. scanning cost	3.0	3.0	3.0	3.0	3.0

To check whether these costs are reasonable we also performed a small-scale experiment where four test persons typed the initial and subsequent queries according to strategies S1-S5 using two types of interfaces (Desktop PC and Smart Phone) for three test topics. The experiment corroborated that the query time estimates were reasonable.

The document snippet scanning costs in real life are affected by the motor and perceptual costs plus the cognitive load related to the task. In this study we assume that the document snippet scanning cost is constant in both scenarios and across the searching strategies S1 – S5 (see Table 1). In the SP case we defined a scanning cost of three seconds per snippet. We justify this by an observation by Kamvar and Baluja [17] that the average cell phone user used 30 seconds to scan the search results before selecting one, after receiving 10 search results. For the snippet scanning costs in the Desktop PC case we decided to use the same value. Obviously, our methodology is well-suited to experiment with different costs. The overall cost constraint of a session was defined as 60, 90, or 120 seconds. In the simulations all subtasks (querying and scanning) had to be performed within this time constraint. We excluded the eventual thinking time in producing query words.

2.5 Research questions

We set forth the following research questions:

1. How effective are the five QM strategies (S1 to S5) in terms of CG when we compare the Desktop PC and the Smart Phone scenarios under overall time constraint?
2. What are the characteristics of the best and the worst sessions achieved in terms of average scan length, and average number of queries?
3. How stable are the observed trends when the overall time constraint changes? Can we recommend QM strategies based on the scenario - what to do, and what not to do, assuming a specific time constraint?
4. What is proper evaluation methodology when time is part of the evaluation setting?

3. RESEARCH SETTING

3.1 Test Collection and Search Engine

We used a subset of the TREC 7-8 document collection with 41 topics for the experiment. The documents have graded relevance assessments on a four-point scale with respect to the topics. [27]

The present authors obtained query words for session generation for the test topics from [20] where the authors used real test persons to suggest keywords of various lengths for queries on the 41 topics. The test persons were asked to directly propose good search words from topic descriptions (descriptions and narratives) in a structured way. Among others, they produced query versions of various lengths: (i) one word, (ii) two words, and (iii) three or more words. These were collected per topic as ordered word lists of 5 words for each topic. During the query formulation experiment the test persons did not interact with a real retrieval system. While this may have affected negatively the quality of queries, Keskustalo and colleagues [20] suggest that the test persons were able to construct the query words in a descending order of effectiveness.

Retrieval system *Lemur* with language modeling and two-stage smoothing options was used in the experiment.

3.2 Session Data

For each topic we utilized a minimum of 1 query and a maximum of 5 queries in each session. A minimum of 1 document snippet and a maximum of 10 document snippets were scanned per query.

In Table 2, the number of possible scanning paths is given for consecutive queries. If the session comprises at most 2 queries, first there are 10 possible paths after the first query, and for every path there are 10 possible paths after the second query. So the combinations of these at most two queries sum up to 10+10*10 =110 possible paths. In our experiment design, users can pose up to 5 queries depending on session strategy; this presents altogether 111,110 possible paths, which are taken into consideration.

Table 2. Number of possible sessions per number of queries, when at most 10 documents can be scanned after each query

Queries	1	2	3	4	5	∑
Possible sessions	10	100	1000	10,000	100,000	111,110

We ran all 41 topics * 5 QM strategies * Q queries, $Q \in \{3, 4, 5\}$ depending on the strategy, and collected their results. Then we generated all 111K possible sessions from the query results, pruned the ones exceeding the time constraint in each scenario, and by using the recall base (qrels), evaluated the CG of the scanned snippets for each session. For example, for the session Q_1->s_{11}-> s_{12}->s_{13}->Q_2->s_{21}->s_{22}->Q_3->s_{31}, the CG is calculated on the basis of the snippet sequence s_{11}, s_{12}, s_{13}, s_{21}, s_{22}, s_{31}. Altogether about 45 million sessions (41 topics * 5 QM strategies* 111,110 possible scanning sessions * 2 scenarios) were evaluated. As the collection has graded relevance assessments, CG was incremented by 3 points for the highly relevant documents, 2 points for the fairly relevant documents and 1 point for the marginal ones. Whenever a duplicate was retrieved by a subsequent query in a session, its gain was nullified. Finally, we ranked all sessions within a topic and a strategy by their CG scores. In this data set per topic, strategy and time constraint, each session is represented by its tuple of actions (see 2.1) and its gain.

3.3 Data Analysis

The action tuples allow the analysis of the number of queries and the length of each scan in a session. The ranked order of sessions allows identification of the best and the worst session across topics, strategy, scenario, and time constraint. We analyze the sets of 10 best sessions, and 10 worst sessions per topic as averages instead the single best or worst session. This approach smoothes minor random variations in human behavior and thus the set of top (bottom) 10 sessions provide more reliable measurements compared to the single best/worst session when we explore their properties under varying conditions. Since the present study does not aim to prove one retrieval method better than another, we report the findings without tests on significance of statistical differences.

4. EXPERIMENTS

4.1 Results for the 60 Seconds Time Frame

First we discuss the CG results under the two scenarios, PC and SP. We present the best case and worst case results regarding all querying-scanning sessions based on the five QM strategies: S1 (*sequence of individual* words); S2 (*two-words;* last word varied); S3 (*three-words;* last word varied); S4 (incremental extension starting from *one* word); and S5 (incremental extension starting from *two* words). Table 3 gives the averaged CG values, the number of queries and scans per query for 10 best and 10 worst cases for every QM strategy for the 60 second time constraint, which are utilized in the following figures in this section.

Table 3. Averaged CG, number of Queries (#q) and Scans per Query (s/q) for scenarios PC and SP, for 5 strategies for the 10 best (b) and 10 worst (w) sessions, time constraint 60 seconds

Time (60 s)	Environment	best/worst	Query Modification Strategies				
			S1	S2	S3	S4	S5
avg. CG	PC	b	4.9	8.3	8.5	7.9	8.1
		w	1.2	5.4	7.1	4.7	6.0
	SP	b	2.8	3.6	2.3	4.4	3.7
		w	1.2	2.8	2.3	2.0	2.9
avg. #q	PC	b	2.7	2.6	2.5	4.2	3.0
		w	5.0	4.0	3.0	5.0	4.0
	SP	b	1.9	1.5	1.0	2.0	1.5
		w	2.7	1.7	1.0	2.6	1.7
avg. s/q	PC	b	6.4	6.3	6.2	3.8	5.3
		w	3.0	3.8	5.0	3.0	3.8
	SP	b	4.7	3.6	2.5	4.0	3.6
		w	1.6	2.5	2.5	1.7	2.5

Table 4 and Table 5 are equivalent to Table 3 but for the time constraints 90 and 120 seconds, respectively. Figure 1 shows the CG of the best (worst) sessions for each strategy in both scenarios under the overall cost constraint of 60 seconds. Note that all sessions require 60 seconds or less if no further action fits in (the absolutely worst imaginable session without any time requirement, in terms of the CG, would naturally consist of the initial action (IA) only). In other words, regarding the worst results, we report CG for the worst possible 60 second performance.

When the best sessions of the PC and SP cases are compared in Figure 1, the PC case performs at a considerably higher level (average CG is above 8 in three strategies) than the SP case (average CG is below 5 in all strategies).

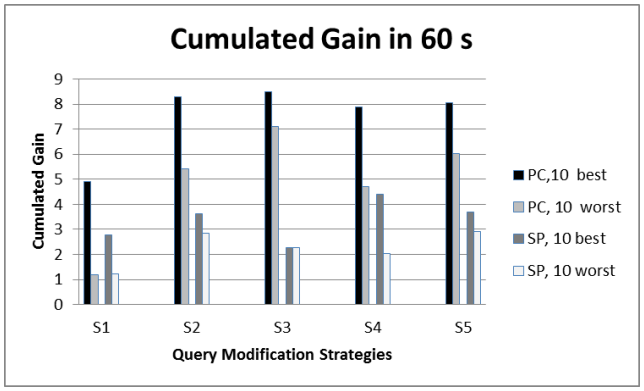

Fig 1. Cumulated Gain under cost constraint of 60 seconds.

Second, when the best and the worst cases are compared within the scenarios, not surprisingly, the best case results are typically clearly better than the worst case results except in SP case for S3. In the latter case both the best and the worst session may not contain more than one query because of high query entry cost.

Third, among the best cases for PC the strategies S2 and S3 are almost equally good. For the SP case, the strategy S2 (varying the second word), S4 (extending from one word), and S5 (extending from two words) lead to much higher gain than S1 and S3. An interesting trade-off in the SP scenario can be observed when the scanning length is considered. In the best case the gain reached increases from S1 to S2. However, the average scanning length decreases (Fig. 2). In other words, a better result is achieved using the longer queries although a smaller number of documents are scanned on the average; the ranking is simply better.

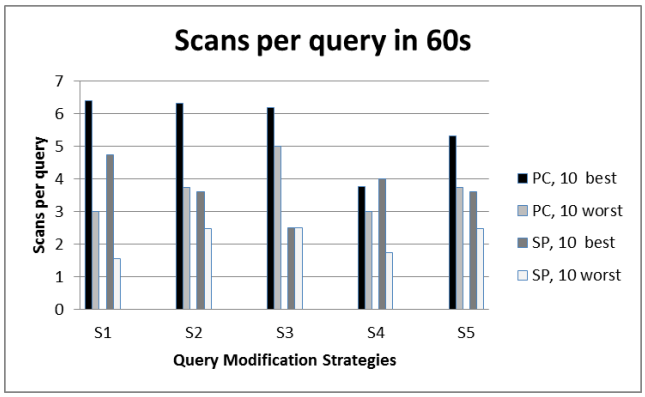

Fig 2. Average number of scanned document snippets per query under cost constraint of 60 seconds.

Table 4. Averaged CG, number of Queries (#q) and Scans per Query (s/q) for scenarios PC and SP, for 5 strategies for the 10 best (b) and 10 worst (w) sessions, time constraint 90 seconds

Time (90 s)	Environment best/worst		Query Modification Strategies S1	S2	S3	S4	S5
avg. CG	PC	b	5.8	10.5	10.4	10.4	10.5
		w	1.9	7.6	10.3	7.6	8.7
	SP	b	5.0	7.3	6.0	7.1	7.0
		w	0.9	2.5	3.0	2.5	2.6
avg. #q	PC	b	3.8	3.5	3.0	5.0	4.0
		w	5.0	4.0	3.0	5.0	4.0
	SP	b	2.0	2.0	1.9	2.5	2.0
		w	4.1	3.4	2.6	4.1	3.4
avg. s/q	PC	b	6.9	7.3	8.3	5.0	6.3
		w	5.0	6.3	8.3	5.0	6.3
	SP	b	8.8	6.8	4.7	6.3	6.8
		w	1.6	1.4	1.7	1.4	1.4

4.2 Results for the 90 Seconds Time Frame

Figure 3 shows the CG results when the sessions take 90 seconds. In this case, the observations comply with the 60 second case. Difference between S3's best and worst CG values is closing in the PC scenario; this is because of lacking further scanning options, there is now enough time to scan almost all 10 documents for each query. S3 strategy has a maximum of 3 queries to execute before the 5 keywords run out. This in turn confines the possible scanning space. It is also conspicuous that the difference between best and worst CG values in SP case is much larger than in PC case.

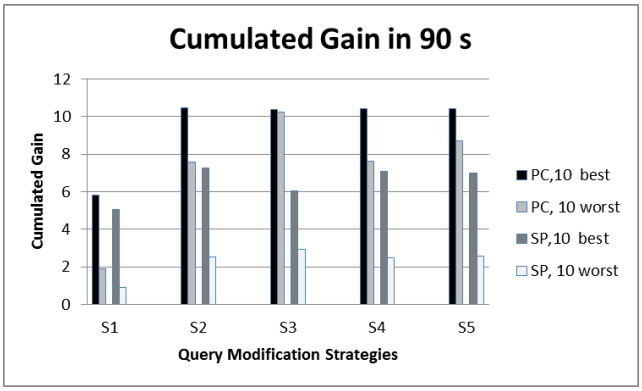

Fig 3. Cumulated Gain under cost constraint of 90 seconds.

When scanning in the best sessions of the PC and SP cases is compared (Fig. 4), we notice that even though the scans per query values for SP case are higher than or similar to the PC case, the CG values are always poorer (Fig. 3). This is due to the smaller number of posed queries in SP case than in PC case. This follows from the trade-off between query vs. scan costs.

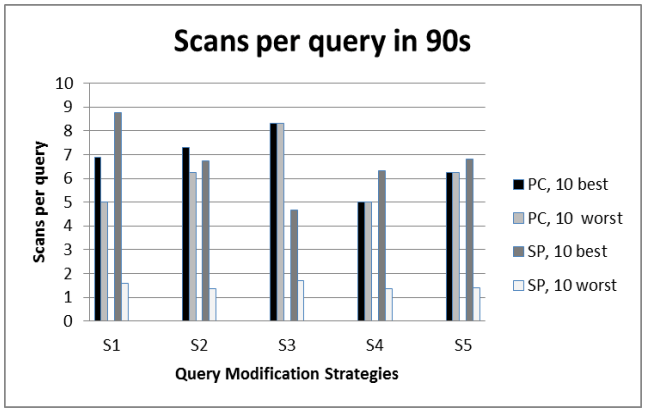

Fig 4. Average number of scanned document snippets per query under cost constraint of 90 seconds.

Interestingly, the difference between the best and the worst sessions both in terms of gain and average scan length remains great in SP case, but fades away in PC case. In the latter, 90 seconds allows the searcher to launch almost all queries and scan the best results in all cases. When the results are compared between different strategies, the strategy S4 with on average 5 scans in PC case and approximately 6 scans in SP case (Fig. 4) produce similar CG values as the other QM strategies (Fig. 3). Again, larger queries yield better rankings. On the other hand, S3 in SP case has less than 5 scans per query, and still achieves slightly better CG results than S1 strategy.

Table 5. Averaged CG, number of Queries (#q) and Scans per Query (s/q) for scenarios PC and SP, for 5 strategies for the 10 best (b) and 10 worst (w) sessions, time constraint 120 seconds

Time (120 s)	Environment best/worst		Query Modification Strategies S1	S2	S3	S4	S5
avg. CG	PC	b	6.4	11.1	11.4	11.7	11.5
		w	3.4	10.5	11.4	10.0	10.9
	SP	b	5.6	9.1	9.2	9.1	8.9
		w	1.1	5.1	6.7	4.5	5.7
avg. #q	PC	b	4.8	4.0	3.0	5.0	4.0
		w	5.0	4.0	3.0	5.0	4.0
	SP	b	3.0	2.9	2.0	3.0	2.9
		w	5.0	4.0	3.0	5.0	4.0
avg. s/q	PC	b	7.3	8.8	10.0	7.0	8.8
		w	7.0	8.8	10.0	7.0	8.8
	SP	b	7.6	6.6	8.8	7.6	6.5
		w	2.8	3.5	4.7	2.8	3.5

4.3 Results for the 120 Seconds Time Frame

Figure 5 shows the CG values under the cost constraint of 120 seconds. In the PC case, the gaps between the best and worst CG values are diminishing. This can be explained so that every strategy except S1and S4 has enough time to pose all the queries and employ much scanning. According to the experiment design, worst cases must also use up the allocated time, and this results in that there is enough time to launch all queries and scan the results. When the best sessions of the PC and SP cases are compared, we notice that there are no large differences. Again, in Figure 6 we can see as many scans per query (S/Q) for S1 and S4 in the SP case as in the

PC case for best sessions. Besides all the strategies for PC case show the same S/Q for 10 best and 10 worst sessions. Although in SP case S/Q values diverge from each other, Figure 6 exhibits similar patterns as Figure 4. From Figure 5 one can conclude that, if there is enough time for searching, one should use at least two word queries for good results. If the queries are of lower quality like S1, then scanning matters. In short, the more you scan, the more you get.

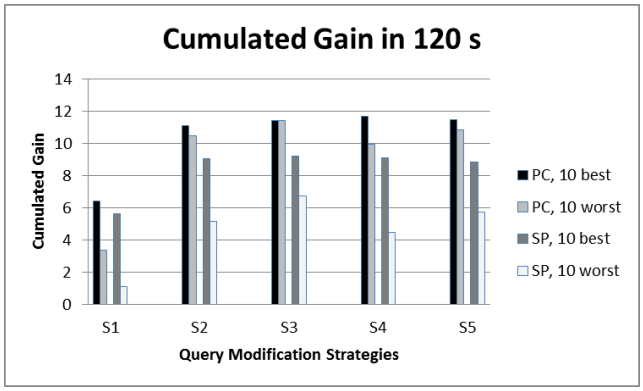

Fig 5. Cumulated Gain under cost constraint of 120 seconds.

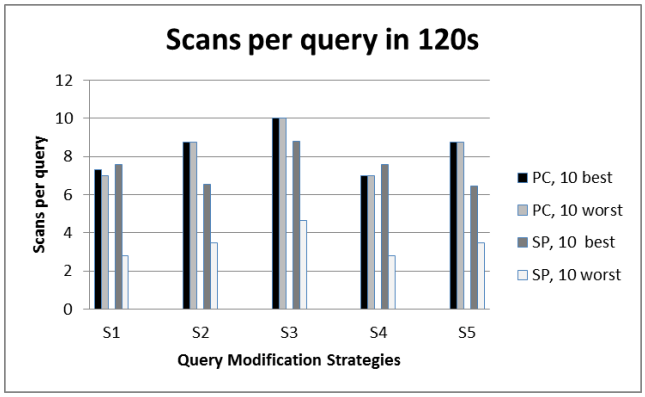

Fig 6. Average number of scanned document snippets per query under cost constraint of 120 seconds.

5. DISCUSSION

We had three empirical and one methodological research question. The three empirical ones were about effectiveness of different QM strategies under time constraints, characteristics of the best and the worst QM sessions, and the stability of the observed trends. The methodological one was about proper evaluation of sessions under time constraints. We will consider each of the questions below.

Strategy effectiveness. Given a stringent time frame in the PC scenario, the user cannot use the entire vocabulary (all queries) and perform exhaustive scanning for all queries. Short queries (strategy S1) are clearly inferior regarding session effectiveness. It seems reasonable to invest on two to three word queries (S2, S3) because the evidence thereby added for ranking significantly improves the quality of the results. This can also be seen in strategies S4 and S5, when they have enough time to advance beyond the first query. When more time is allocated to searching, the weaker strategies catch up because there is more time for scanning the results and the weaker ranking effectiveness is not that critical.

In the SP scenario the rules of the game change a bit. In a stringent time frame there is no time for tedious query input, and one must compromise toward short scanning of weaker quality rankings. The more effective strategies cannot be applied at all due to high query input cost. Again, when more time is allocated, weaker strategies catch up. In the longest sessions of S2-S5, the gap between the best vs. worst sessions begins to close.

Session characteristics. In the PC scenario, under stringent time constraints, the best sessions involved less queries and longer scans than the worst sessions (Table 3). However, as the time allocation grows, the differences disappear. Between the best strategies in the PC case, both the number of queries and the average scan lengths increase as time allocation grows (Tables 3-5). Correspondingly, in the worst sessions, the number of queries does not change as time grows, but the scan lengths grow. This is because the worst sessions consume all possible queries even under the shortest time frame. Similarity with best sessions grows.

In the SP scenario, under stringent time constraints, the best sessions also involved less queries and longer scans than the worst sessions (Table 3). As the time allocation grows, the differences remain, probably due to shortage of time even in the longer sessions. Between the best strategies in the SP case, both the number of queries and the average scan lengths increase as time allocation grows, the latter dramatically between 60 and 90 seconds (Tables 3-4). Correspondingly, in the worst sessions, the number of queries grows along time, but the scan lengths remain low. The worst behavior here means investing the effort in query input. Also here there were interesting differences in scan lengths between queries in sessions.

All in all, if time allows, two to three first query words that one identifies, followed by a longer scan, seem to provide reasonable performance, no matter what the strategy among S2-S5 is.

Effect of time. With limited time allowance, it seems important to make a good compromise between providing evidence for ranking (longer queries) and scanning the search results. The compromise depends on the overall cost levels related to the stringency of the time frame and on the relations between cost types. This depends on the searching device. Expensive input favors scanning at length, cheap input favors better queries. The more time is available the less it matters how one searches – there will be time to identify the relevant documents.

Evaluation methodology. Typical IR evaluation metrics are based on the quality of ranking alone. In session-based evaluation they must be applied with great care because they may be insufficient or even misleading. They may be partially insensitive to the user's experience and observed costs and benefits. This is particularly critical, when user's costs (time expenditure) are taken into account and the metric employs normalization, i.e. scaling the measurements to a predefined range such as [0, 1]. For example, the popular NDCG metric [15] and its non-discounted counterpart NCG should be avoided in any comparisons between searching environments, and between strategies within a given searching environment *when* input costs are taken into account. This is because the ideal gain vector used for normalization is read to vastly different lengths between strategies or environments. For example, consider Figure 7, which plots NCG over time for strategy S2 in the two scenarios.

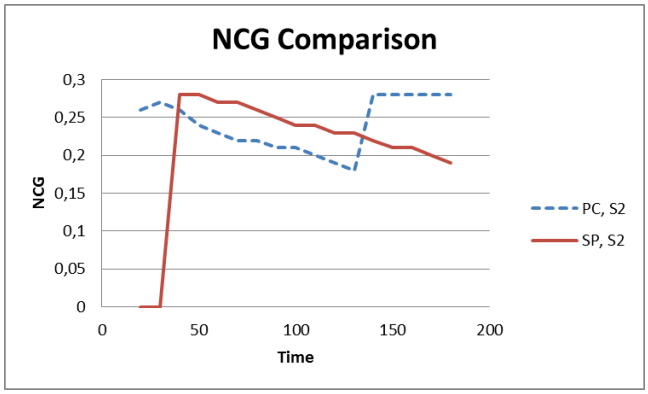

Fig 7. NCG vs. time comparison of PC and SP for S2 (41 Topics).

Due to normalization (division by the ideal cumulated gain vector) the SP scenario seems to have better performance in the time frame from 40 to 135 seconds. This is due to (a) ranking being somewhat effective, and (b) the number of documents seen in each session: in the PC case the user sees 15 to 35 documents, but in the SP case only 5 to 20 documents in the indicated time frame. Figure 8 plots CG with the corresponding data and makes the difference clear.

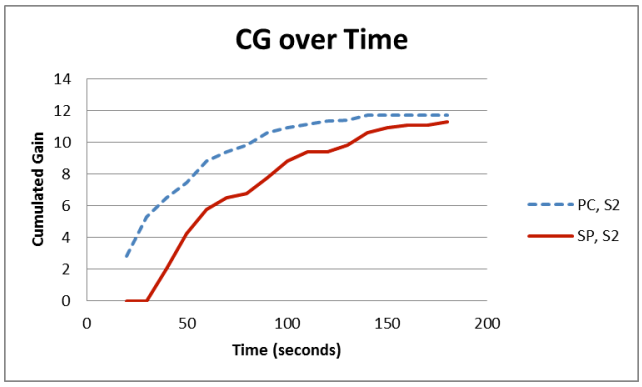

Fig 8. CG over time for S2 in scenarios PC and SP (41 Topics).

Similar pitfalls also plague the most classic metric, MAP. Consider the following two rankings observed for a given topic in two scenarios and/or strategies under the same time constraint (say, one minute; queries omitted and binary relevance for simplicity):

r1: 0 0 0 0 0 0 0 0 1 1

r2: 1 0 0 0 0

Further, assume that there are three relevant documents for the topic. The MAP for the ranking r1 is (1/9 + 2/10 + 0)/3 = 0.103 and for r2 (1 + 0 + 0)/3 = 0.333. Arguably, r2 is the better ranking, but if both require one minute, what is the user's opinion? The first session collected twice as many relevant documents.

Even within the un-normalized metric, such as CG, incorporating time in session-based evaluation has profound effects. Consider Figures 9 and 10. The former gives traditional cumulated gain over ranks for strategies S1 and S3 for the 41 topics. The latter gives CG over time in the two scenarios.

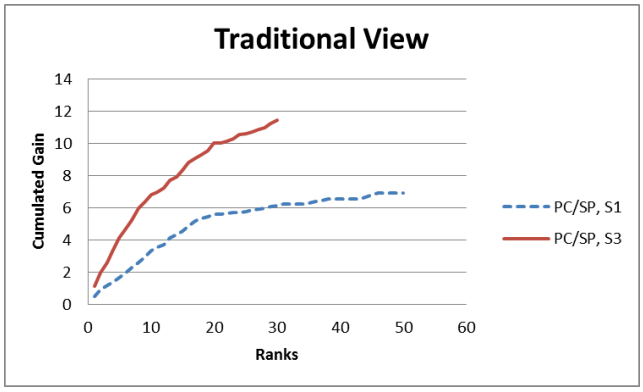

Fig 9. Traditional View, CGs over ranks for 41 topics, scenarios PC and SP for strategies S1 (allowing 5 queries) and S3 (allowing only 3 queries).

In Figure 9, both scenarios PC and SP have the same observed effectiveness, because the evaluation focuses on the gain (CG) over the result ranks, no matter how long it takes to retrieve the documents. The two strategies S1 and S3 differ in effectiveness, S3 providing far better effectiveness than S1. However, when time is taken into account (Fig. 10), the scenarios and strategies differ greatly from each other. Up to 60 seconds, S3 in the SP case is the worst strategy and this is entirely due to the high input cost of the long query. With enough time (180 sec.), S3 in SP catches up S3 in PC case. Also, PC and SP do not much differ for S1 due to the relatively low input cost and weak result quality. Comparing Figures 9 and 10, it is easy to see that time drives interaction and profoundly affects both user experience and effectiveness in sessions in different scenarios.

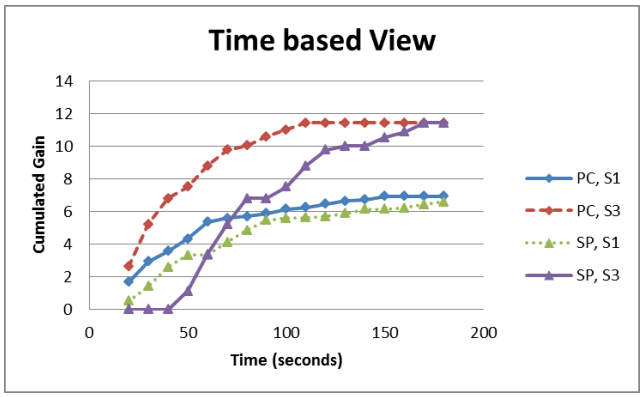

Fig 10. Time based View, CGs over time for 41 topics, scenarios PC and SP for strategies S1 and S3.

Limitations. In our study we did not take into account the time, which users spend for pondering about possible query words. One might argue that the more words one needs to identify, the harder (and slower) per word it comes. However, the thinking time is the same between sessions using the same number of words. In addition, this could be taken into account by revising subsequent query costs (Table 1). We have chosen to short-cut here in order to avoid too much complexity at this stage. Furthermore, we do not consider the time users spend in examining documents. This may depend on the device used. This can be seen as an artificial limitation. Tackling it would, however, complicate analysis, and this is therefore left for later study. We did not simulate user's learning during a session. Admittedly, learning from snippets and seen documents take place.

This is not impossible to simulate but some challenges remain to be solved.

We employed in the evaluation relatively limited query vocabularies, simple bag-of-word queries, and relatively short time frames. The query vocabularies and structure are justified by query length statistics in many search environments [14], [29], and the time frames by our simulation capabilities. However, the time frames are for *effective search time in sessions,* excluding thinking and document examination time. While the query vocabularies are short, they are human-generated for this collection, and therefore more realistic than words mined, e.g., from known relevant documents (in qrels).

We did not cover all the imaginable complex sessions. However we employed idealized and literature-based sessions, which shed the light on the peculiar evaluation problems beyond the traditional rank-based evaluation. This is a step forward while we are not suggesting that anyone follows a single strategy consistently in real life.

Our initial results are promising. First, the scenario, and to a large extent the device itself, dictate what kind of interactive behavior can be successful. Because real users do have limited resources and they use various devices having different properties, our methodology has unquestionable user relevance and potential pragmatic value for the industry. Measuring the effectiveness of systems from the pragmatic point of view may increase the validity of the results achieved. This may lead to greater user satisfaction. Secondly, our experimental results suggest that strict time constraints determine some session strategies as the best strategies as they maximize CG. The strengths of our approach are:

- The QM strategies S1-S5 have an empirical real life grounding
- The query vocabularies were generated by real test persons, and only thereafter used in automatic simulation
- We were able to evaluate over 20M sessions in each scenario; this is clearly intractable both physically, intellectually and economically with human test persons.

We have only taken the first steps. In future, we will study the dimensions of variation related to users, systems, information sources and sessions to construct more fine-grained scenarios explicating hypotheses about user goals, learning, and behaviors to validate evaluation measures used. [19]

6. CONCLUSIONS

In this study, we have shown the necessity of a pragmatic evaluation approach based on scenarios with explicit subtask costs under an overall time constraint. Effectiveness of various query modification and scanning strategies for two scenarios, namely, PC and SP is analyzed. Furthermore, the characteristics of the best and the worst interactive search sessions are examined. Expensive input favors scanning at length, cheap input favors better queries. The more time is available the less it matters how one searches – there will be time to identify the relevant documents. We have shown that the effort required by searching devices and the overall search time allocation drive interaction and profoundly affect both user experience and effectiveness in sessions in different scenarios. Moreover, we have also pointed out the inapt use of all normalized rank-based measures. Thus, we hope we could instigate new evaluation metrics for time-based comparisons.

7. ACKNOWLEDGMENT

This research was funded by Academy of Finland grant number 133021.

8. REFERENCES

[1] Azzopardi, L. 2007. Position Paper: Towards Evaluating the User Experience of Interactive Information Access Systems. In *SIGIR'07 Web Information-Seeking and Interaction Workshop*, 5 p.

[2] Azzopardi, L. 2011. The economics of interactive information retrieval. In *Proceedings of the 34th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 15-24.

[3] Bates, M. J. 1979. Information search tactics. *Journal of the American Society for Information Science,* 30(4), 205-214.

[4] Bates, M. J. 1989. The Design of Browsing and Berrypicking Techniques for the Online Search Interface. *Online Review*, 13(5), 407-424.

[5] Beaulieu, M. 2000. Interaction in Information Searching and Retrieval. *Journal of Documentation*, 56(4), 431-439.

[6] Belkin, N. L. 1980. Anomalous States of Knowledge as a Basis for Information Retrieval. *Canadian Journal of Information and Library Science*, 5, 133-143.

[7] Card, S. K., Moran, T. P., and Newell, A. 1983. *The Psychology of Human-Computer Interaction*. L. Erlbaum Assoc. Inc., Hillsdale, NJ, USA.

[8] Cleverdon, C.W., Mills, L., and Keen, M. 1966. Factors determining the performance of indexing systems, vol. 1-design. In *Aslib Cranfield Research Project*, Cranfield.

[9] Dunlop, M. D. 1997. "Time Relevance and Interaction Modeling for Information Retrieval". In *Proceedings of the 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Philadelphia,206-213.

[10] Fidel, R. 1985. Moves in online searching. *Online Review*, 9 (1), 62-74.

[11] Hearst, M. A. 2011. "Natural" Search User Interfaces. *Communications of the ACM*, vol. 54, 60-67.

[12] Hersh, W., Turpin, A., Price, S., Chan, B., Kraemer, D., Sacherek, L., and Olson, D. 2000. Do Batch and user Evaluations Give the Same Results? In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 17-24.

[13] Ingwersen, P. and Järvelin, K. 2005. *The Turn: Integration of Information Seeking and Retrieval in Context.* Heidelberg, Springer.

[14] Jansen, M. B. M., Spink, A., and Saracevic, T. 2000. Real Life, Real Users, and Real Needs: A Study and Analysis of User Queries on the Web. *Information Processing & Management*, 36(2), 207-227.

[15] Järvelin, K. and Kekäläinen, J. 2002. Cumulated Gain-Based Evaluation of IR Techniques. *ACM Transactions on Information Systems,* 20(4), 422-446.

[16] Järvelin, K. and Kekäläinen, J. 2000. IR evaluation methods for retrieving highly relevant documents. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Athens, Greece, 41-48.

[17] Kamvar, M. and Baluja, S. 2007. Deciphering Trends in Mobile Search. *Computer,* 40(8), 58-62.

[18] Karat, C-M., Halverson, C., Horn, D., and Karat, J. 1999. Patterns of entry and correction in large vocabulary continuous

speech recognition systems. In *ACM Conference on Human Factors in Computing Systems*, 568-575.

[19] Karlgren, J., Järvelin, A., Eriksson, G, and Hansen, P. 2011. Use cases as a component of information access evaluation. In *DESIRE'11 workshop, October* 28, 2011, Glasgow, Scotland, UK.

[20] Keskustalo, H., Järvelin, K., Pirkola, A., Sharma, T. and Lykke, M. 2009. Test Collection-Based IR Evaluation Needs Extension Toward Sessions – A Case of Extremely Short Queries. In *Proceedings of the 5th Asia Information Retrieval Symposium* (AIRS'09), 63-74.

[21] Kuhlthau, C. C. 1991. Inside the Search Process. *Journal of the American Society for Information Science*, 42(5), 361-371.

[22] Price, S.L., Nielsen, M.L., Delcambre, L.M.L., and Vedsted, P. 2007. Semantic Components Enhance Retrieval of Domain-specific Documents. In *Proceedings of the 16th ACM CIKM*, 429-438.

[23] Ruthven, I. 2008. Interactive Information Retrieval. In *Annual Review of Information Science and Technology*, vol. 42, 2008. 43-91.

[24] Salton, G. 1970. Evaluation Problems in Interactive Information Retrieval. *Information Storage and Retrieval,* 6, 29-44.

[25] Smith, C. L. and Kantor, P. B. 2008. User Adaptation: Good Results from Poor Systems. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 147-154.

[26] Smucker, M. D. 2009. Towards Timed Predictions of Human Performance for Interactive Information Retrieval Evaluation. In *Third Workshop on Human-Computer Interaction and Information Retrieval* (HCIR'09), October 23, 2009, Washington DC, USA.

[27] Sormunen, E. 2002. Liberal Relevance Criteria of TREC – Counting on Negligible Documents? In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Tampere, 324-330.

[28] Spink, A. 1997. Study of Interactive Feedback during Mediated Information Retrieval. *Journal of the American Society for Information Science*, 48(5), 382-394.

[29] Stenmark, D. 2008. Identifying Clusters of User Behavior in Intranet Search Engine Log Files. *Journal of the American Society for Information Science*, 59(14), 2232-2243.

[30] Su, L.T. 1992. Evaluations Measures for Interactive Information Retrieval. *Information Processing & Management* 28(4), 503-516.

[31] Turpin, A. and Hersh, W. 2001. 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*, 225-231.

[32] Turpin, A. and Scholer, F. 2006. 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*, 11-18.

[33] Vakkari, P. 2000. Cognition and changes of search terms and tactics during task performance. In *Proceedings of RIAO 2000 Conference*, Paris: C.I.D., 894-907.