Segment-Level Display Time as Implicit Feedback: A Comparison to Eye Tracking

Georg Buscher[1,2], Ludger van Elst[1], and Andreas Dengel[1,2]
[1]Knowledge Management Dept., DFKI
[2]Dept. for Knowledge-Based Systems, University of Kaiserslautern
Kaiserslautern, Germany
{georg.buscher, ludger.van_elst, andreas.dengel}@dfki.de

ABSTRACT

We examine two basic sources for implicit relevance feedback on the segment level for search personalization: eye tracking and display time. A controlled study has been conducted where 32 participants had to view documents in front of an eye tracker, query a search engine, and give explicit relevance ratings for the results. We examined the performance of the basic implicit feedback methods with respect to improved ranking and compared their performance to a pseudo relevance feedback baseline on the segment level and the original ranking of a Web search engine.

Our results show that feedback based on display time on the segment level is much coarser than feedback from eye tracking. But surprisingly, for re-ranking and query expansion it did work as well as eye-tracking-based feedback. All behavior-based methods performed significantly better than our non-behavior-based baseline and especially improved poor initial rankings of the Web search engine.

The study shows that segment-level display time yields comparable results as eye-tracking-based feedback. Thus, it should be considered in future personalization systems as an inexpensive but precise method for implicit feedback.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Relevance Feedback*

General Terms

Experimentation, Measurement

Keywords

Personalization, implicit feedback, eye tracking, display time

1. INTRODUCTION

Personalization has quite some history in information retrieval research. It aims at incorporating user-specific information like context and interaction data and adapting the search process according to individual user needs and preferences. Yet, this is a very hard problem and it has been identified as one of the major challenges in information retrieval research lately [4].

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
SIGIR'09, July 19–23, 2009, Boston, Massachusetts, USA.
Copyright 2009 ACM 978-1-60558-483-6/09/07 ...$5.00.

The goal of interpreting user interaction data is to infer user-perceived relevance of documents or content. It has been shown that explicitly asking the user for relevance feedback is very useful for individually improving the accuracy of search [20]. However, asking for explicit relevance feedback is an additional burden to the user and demands extra cognitive effort. Therefore, automatically inferring relevance by interpreting user behavior is a highly promising approach.

Until now, research in this area has primarily focused on user interaction data that is easy to get, i.e., coming from keyboard and mouse input allowing for measures such as click-through, display time, scrolling, mouse movements, etc. (compare [15]). In most cases, these measures have been used to estimate object-level relevance, thus relevance for entire documents. Recently, research has been conducted to shed light on how additional measures like eye movements [6, 5, 21] or emotions [3] might relate to user intent and user assessment of individually perceived relevance. Especially feedback generated from eye tracking can be applied on the segment level, e.g., for specific passages of a document. This kind of precise feedback has been proven to be particularly effective for improving search accuracy [6].

But eye trackers with sufficient precision are too expensive nowadays so that such feedback data will not be available in a common workplace in the near future. Scrolling behavior, however, is simple to observe so that display times of different document segments are easy to compute. Inexpensive, yet accurate feedback from display time would be highly valuable since it can be simply logged by browser toolbars.

In this paper we examine the relationship between segment-level display time and segment-level feedback from an eye tracker in the context of information retrieval tasks. How do they compare to each other and to a pseudo relevance feedback baseline on the segment level? How much can they improve the ranking of a large commercial Web search engine through re-ranking and query expansion? To answer those questions after providing some background information, we first explain the technical basis of the applied implicit feedback methods. Then, the study design is described, followed by a detailed analysis of the results and a conclusion.

2. BACKGROUND AND RELATED WORK

A good deal of research has been conducted in the area of interpreting user interaction data for implicit feedback. In this section, we will divide the works into those aiming at object-level feedback and those targeting the segment level. With reference to Oard & Kim [18] feedback on the object level means interpreting user interaction data on the level

of entire documents. Thus, object-level feedback matches the kind of feedback that is needed in a classical relevance feedback scenario as per Rocchio [20]. In contrast, segment-level feedback is more finer grained, for example on the level of headlines or paragraphs.

2.1 Object-Level Implicit Feedback

Most existing research about implicit relevance feedback focuses on the object level and is based on mouse- and keyboard-related interaction data (compare also [15]).

Some of the first research done in the area of implicit feedback in information retrieval was that of Morita & Shinoda [17]. They conducted a study where participants had to provide explicit relevance feedback for news articles after having read them. The study focused on the correlation between reading time and explicit feedback while considering document length and additional textual features. They noted that there is a strong tendency to spend more time on interesting articles rather than on uninteresting ones. This is a finding which has also been reported in [7] and [8]. Furthermore, Morita & Shinoda found only a very weak correlation between the lengths of articles and associated reading times, indicating that most articles are only read in parts, not in their entirety.

However, Kelly & Belkin [13] tried to reproduce the results of Morita & Shinoda in a different, more complex information retrieval scenario but found no correlations between display time and explicit relevance feedback for documents. In a later naturalistic study Kelly & Belkin [14] found again no general relationship between display time and the users' explicit ratings of the documents' usefulness. They rather observed a high variation of display time with respect to different users and different tasks.

Later, White & Kelly [22] reported that adjusting display time thresholds for implicit feedback according to task type leads to improved retrieval performance. On the contrary, adjusting the thresholds according to individual users worsened performance. This stands in contrast to findings of a prior study by Rafter & Smyth [19] who showed for one specific task type that display time is correlated with user interest, especially after individually adjusting the measure.

Additional implicit measures have been examined on the object level as well. On the one hand, it has been found that good indicators of interest are for example the amount of scrolling on a page [7], click-through [8, 12], and exit type for a Web page [8]. On the other hand, mouse movements and mouse clicks while viewing a document are rather noisy with regard to user interest [7]. Furthermore, when incorporating a variety of different implicit measures at the same time (page dwell time among them) considerable improvements of search result ranking can be achieved as reported by Agichtein et al. [1]. Individually personalizing the measures leads to further improvements as shown by Melucci & White [16].

2.2 Feedback on the Segment Level

Segment-level feedback is less thoroughly studied than object-level feedback. However, there certainly are user actions suggesting that specific document parts are more interesting or relevant than others. This is important to know especially for longer documents containing multiple topics. For example, Golovchinsky et al. [9] focused on user-created annotations on documents such as highlightings, underlinings, circles, and notes in margin. They used this kind of feedback to infer relevance of document passages. In a document search scenario utilizing query expansion, they reported a significant improvement of the annotation-based feedback technique over explicit relevance feedback on the document level.

Ahn et al. [2] followed a similar idea but used the concept of a personal notebook where users could paste text passages worth remembering. On the basis of the text passages they built up term-based task profiles which were then used for re-ranking search result lists. Compared to a baseline ranking function not considering any feedback, the task-profile-based ranking performed significantly better.

The previous two approaches both need more or less explicit and therefore rare user interactions (i.e., annotating, copying&pasting) to work properly. Buscher et al. [6] only rely on implicit data and determine which parts of a document have been read, skimmed, or skipped by interpreting eye movements. Read and skimmed parts were taken as relevant while skipped document parts were ignored. They report considerable improvements concerning re-ranking of result lists when including gaze-based feedback on the segment level compared to relevance feedback on the document level.

Gyllstrom & Soules [10] follow a similar idea, but consider all text that has been visible on the screen for building up term-based task profiles. They use such profiles for task-based indexing of documents on the desktop and show that re-finding documents that way is more effective compared to simple desktop search.

In general, it has been shown that display time as implicit relevance feedback works relatively well on the object level. However, there seems to be great variation due to user type and task type. Further, it has been shown that segment-level feedback can outperform object-level feedback due to its higher precision.

3. IMPLICIT FEEDBACK METHODS

In the conducted study we applied two basic input sources for generating implicit feedback: segment-level display time based on scrolling behavior and reading detection based on eye tracking. Furthermore, we wanted to compare such behavior-based evidence to non behavior-based pseudo relevance feedback from the query itself on the level of paragraphs. In all cases, the aim was to determine parts of previously seen documents that are more relevant than others within the context of the user's current search session. The most characteristic terms from those document parts should then be used for query expansion and re-ranking purposes within the same search session.

3.1 Segment-Level Display Time

Preprocessing. Display time on the segment level, i.e., the duration a passage of a document has been visible on the screen, can very simply be measured by analyzing scrolling behavior. We implemented a javascript-based observation component for a Web browser analyzing every scrolling action from the user. The component measured the display time of every single line of text contained in the document. After a page view was finished, the display time was coarsened to the level of paragraphs by averaging the display times of the contained lines.

Method *DsplTime(t)*. We implemented two methods for extracting terms based on segment-level display time. The method *DsplTime(t)* first concatenates all documents viewed by the user within one search session in their entirety resulting in one large virtual context document d. Second, a filtered context document d_P is generated containing only those parts of d that got a display time greater than the threshold t in seconds (the optimal value for t is derived in the experiment, see section 5.1). Third, based on the filtered context document d_P, the most characteristic terms are extracted according to their $tf \times idf$ scores[1].

Method *DsplTimeNeg(t_1, t_2)*. This method basically works in the same way as *DsplTime(t)*, but incorporates negative feedback while computing term scores in the following way: It generates the same filtered context document d_P as above based on the time threshold t_2 in seconds. Additionally, it creates a filtered context document d_N only containing those parts of d that achieved a display time between t_1 and t_2 seconds (for optimal times t_1 and t_2 see section 5.1). Thus, d_N contains text parts that have only shortly been displayed. The score for a term w is computed based on formula (1)[1].

$$\frac{tf(w, d_P)}{tf(w, d_P) + tf(w, d_N)} \times idf(w) \quad (1)$$

$tf(w, d_X)$ corresponds to the term frequency of term w within document d_X. Thus, terms that are very specific for longer displayed text sections of d tend to get higher scores than terms that also appear in shortly displayed parts.

3.2 Eye-Tracking-Based Reading Detection

Preprocessing. During reading, eye movements show a very characteristic pattern composed of fixations and saccades. In short, during fixations the eyes are steadily gazing at one point; saccades are very fast, ballistic movements from one fixation to the next. While reading, the majority of saccades go from the left to the right, text line by text line – a pattern that can be identified as such. Figure 1 shows an example of typical eye movement paths while reading (circles = fixations, lines = saccades).

We applied an algorithm for reading detection as described in [5] coupled with an unobtrusive Tobii 1750 desk-mounted eye tracker and used it to find the lines of text that have been read by the user according to observed reading patterns. This algorithm provided character-offset-based intervals specifying which text snippets (on the granularity of single lines of text) have been read by the user (compare interval offsets noted in Figure 1). The intervals were coarsened to the paragraph level as follows: Two intervals were concatenated and merged into one larger interval if they belonged to the same paragraph and were close to each other, i.e., if they either overlapped, or if not more than 130 characters were in between them. (130 characters corresponded to approximately one and a half lines of text in our setting.) Otherwise, the intervals stayed unchanged.

Method *EyeTrack(l)*. Similar to display time, two virtual context documents d_P and d_N are generated based on the text parts read by the user. d_P contains the contents of coherently read text parts (as specified by the above-computed intervals) having a length at least l characters. Likewise, d_N contains only read text parts of a length less than l characters (for an optimal l see section 5.1). The most characteristic terms are determined according to formula (1) from above. The basic assumption of this method is that users stop reading if the information is irrelevant; otherwise they continue reading. Thus, terms contained in short coherently read parts of the text tend to get lower scores than terms in long coherently read parts.

[1]For the computation of *idf* values Wikipedia was used as a document corpus.

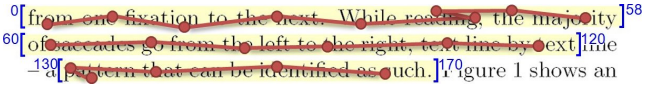

Figure 1: Eye movements while reading.

3.3 Query Focus

Pseudo relevance feedback on the segment level is a method to find relevant document parts without explicitly logging user behavior. Therefore, it is well-suited as a baseline for our behavior-based methods.

Method *QueryFocus*. Similar to the methods above, a virtual context document d is created by concatenating all documents viewed by the user within one search session. Next, d is split into a set of paragraphs. Using Lucene[2] in conjunction with a BM25 ranking function, the user query is applied to the set of paragraphs resulting in a ranking score for each paragraph. Out of the entire set, those paragraphs are selected that achieved a ranking score not worse than half of the ranking score achieved by the best matching paragraph. The selected paragraphs are merged into one virtual document d_P which forms the basis for $tf \times idf$-based term extraction, like in the method *DsplTime(t)* from above.

4. STUDY DESIGN

The main aim of this study was to derive optimal parameter settings for the implicit feedback methods described above and to compare their performance with respect to improved ranking of search results in an information retrieval scenario. The study comprised two independent experimental sessions with two independent groups of participants. For both groups the task protocol looked exactly alike, i.e., they were given an informational task, they had to read through preselected documents while being eye-tracked, query a Web search engine, and judge randomized result lists. However, two different procedures were applied for the two groups: *re-ranking* and *query expansion*. The first experimental session (17 participants) was used to derive optimal parameter settings for the implicit feedback methods and to test re-ranking of search results based on implicit feedback. In the second experimental session (15 participants), the best parameter settings were used to test and validate the effectiveness of the feedback methods using query expansion.

4.1 User Tasks

From the participants' viewpoint the task was as follows in both experimental sessions:

1. They were given a relatively broad information need.
2. They looked through 4 preselected documents containing information relevant to their information need.
3. They had to pose 3 different queries according to more specific (sub)information needs to a search engine.
4. For each query they had to judge all delivered result entries (up to 20 per query in randomized order).

[2]http://lucene.apache.org/

Each participant had to perform those four steps for two different information needs. We provide more details for all four steps in the following.

(1) One information need was about the topic *perception*, the other one about *thermoregulation*. In both cases, the participants were told to put themselves into the position of a journalist who's task is to write magazine articles about each topic. The exact task description for each topic was printed in an email that contained the 4 preselected documents as attachments. In both cases we selected the same 4 articles from the German version of Wikipedia about snakes, bees, dogs, and seals. We modified the articles to be of similar length (around 4200 words). Each article described many different aspects of the focused animal species. However, all of them contained relevant sections concerning both information needs.

(2) The participants were told that the 4 email attachments should help them in getting to know the corresponding task topic. They had moderate time pressure, i.e., around 15 minutes, to look through all 4 preselected documents for each task. This did not allow them to read thoroughly through the entire articles; they rather had to focus on their task topic. Only during this phase of the experiment the eye movements of the participants were recorded and display time was measured.

(3) After having looked through the 4 preselected documents, they had to pose three different queries to a search engine. They had to use a search interface created by us which looked similar to those of popular Web search engines. One query was to find more information about the task topic in general. The other two queries were to find information about specific subtopics. For example concerning the topic *perception* the two subtopic queries should retrieve information about *visual* and *magnetic perception* in particular. The topics for the queries were told orally to the participants and they were free in formulating them.

(4) For each issued query, the participants had to give explicit relevance feedback for all returned result entries (up to 20). We utilized a 6-point feedback scale ranging from "+++" to "– – –". The participants were told not just to look at the document summary on the result list, but to open and quickly look through each document before giving their relevance rating. In all cases, the results were presented in randomized order.

All tasks and preselected documents were in German. The Web search engine was parameterized to return Web pages only in German. Before starting the experiment, each participant was briefed and the eye tracker was calibrated. Half of the participants started with the topic *perception*, the other half with the topic *thermoregulation*.

4.2 Two Procedures: Re-Ranking and Query Expansion

We applied two procedures to test the effectiveness of the terms extracted by the different feedback methods: On the one hand, the procedure applied in the first experimental session comprised re-ranking of search results from the commercial Web search engine Live Search[3]: A user query from phase (3) of the task protocol was first submitted to the search engine whereupon the top X results were re-ranked based on the terms extracted by the different feedback methods. On the other hand, the procedure applied in the second experimental session employed query expansion: The top characteristic terms extracted by the different methods were first added to the original user query (task phase 3). Each expanded query was then submitted to Live Search. However, the task protocol in both experimental sessions was the same; they looked exactly alike for the participants.

There are mainly two reasons why we chose to examine the effects using both procedures. First, we conjectured that the impact of query expansion and re-ranking on the quality of the search results is different: Re-ranking of a result list cannot change the absolute quality of it since no new documents are retrieved. In contrast, submitting an expanded query to a search engine can lead to a considerable change of the overall result quality since additional documents might be retrieved or others might be excluded. Thus, one aim was to validate our results using two different procedures.

The second reason is related to finding concrete parameters that work well for the different implicit feedback methods. In this respect, the re-ranking procedure clearly has the advantage with respect to evaluation that the participant is needed only one time to provide explicit relevance feedback for the top X retrieved results per query. Since the set of documents does not change with re-ranking, an arbitrary number of re-ranking methods can be evaluated based on one fully rated result list afterwards. Thus, the re-ranking procedure lets us evaluate dozens of different term extraction methods posteriori, i.e., the 4 implicit feedback methods from above based on a variety of different parameter settings, in order to find the best parameters. This would not have been possible with the query expansion procedure because every expanded variant of a user query might produce a completely different result list for which we might not have explicit user ratings.

Re-Ranking Procedure. In this procedure, each user query was first submitted to the Live Search engine using the corresponding SDK[4]. The top 20 resulting Web pages were automatically downloaded and stored locally, either directly from their Web site or, if that took too long, then from the Live Search cache. The order of the top 20 results was randomized before user presentation. After a participant judged all of the top 20 results, we had everything needed for posteriori re-ranking:

- the user query,
- the resulting Web documents,
- relevance judgements for those documents, and
- the logged user behavior data concerning eye tracking and display time on the 4 preselected documents the participant viewed before issuing the query.

After an experimental run was finished, re-ranking was performed as follows: First, a document index was built for the top 20 result documents from Live Search using Lucene[2]. Second, each implicit feedback method was applied on the preselected documents providing a list of terms paired with their achieved scores. Third, since the Lucene framework allows for weighted query terms, an expanded query was built by adding the top x extracted terms weighted by their achieved scores. x was determined based on the user-given terms so that the expanded query, i.e., user-given plus expansion terms, had a total number of 19 terms. (This tech-

[3]http://www.live.com/

[4]http://dev.live.com/livesearch/sdk/

nique was applied to be comparable to the query expansion procedure below which did not work with more than 19 query terms.) Weights were also added to the original user query terms so that 40% of the total weight lied on the user-given terms and 60% on the expansion terms. Fourth, the weighted expanded query was used by Lucene to re-rank the top 20 documents from the original Live Search result list.

Query Expansion Procedure. With this procedure we aimed at comparing the four implicit feedback methods from above (i.e., *DsplTime*, *DsplTimeNeg*, *EyeTrack*, *QueryFocus*), each with their best parameter setting as determined by the re-ranking procedure. Each user query was first expanded with the best m terms extracted by the implicit feedback methods and then issued to Live Search via its SDK. Since Live Search did not support weighted terms we used the following format for expanded queries:

$term_{U1}$ ... $term_{Un}$ ($term_{E1}$ OR ... OR $term_{Em}$)

Thus, the user-given terms $term_{Ui}$ were written as a space-separated list, whereas the expansion terms $term_{Ei}$ were connected with OR operators. An expanded query had at most 19 terms, i.e., $n + m = 19$, because the Live Search SDK did not support longer queries.

In this way, we got 4 expanded queries according to the 4 applied implicit feedback methods for each original user query. Each of them as well as the unexpanded user query yielded a separate result list from the Web search engine. To avoid a too high burden on the participant, at most 20 results should be presented for acquiring explicit relevance feedback for each user query. Therefore, the first X results of each separate result list were included in a merged (distinct) result list so that the total number of results of the merged list was ≤ 20. Thus, each merged list contained at least the first X entries from every result list generated by the feedback methods (X turned out to be 8.8 during the experiment, so there was considerable overlap between the separate result lists). In the end, the order of the results in the merged result list was randomized before user presentation. We kept track of the origin of each entry in the merged list so that we could assign the judgements of entries from the merged list to entries from the separate lists later on.

4.3 Evaluation Metrics

To evaluate the quality of the different result lists and their rankings, we compute three different well-accepted information retrieval metrics, each of which focuses on a different aspect of system performance.

- **Precision at K**. $P(K)$ measures the fraction of relevant documents within the top K results. Since it needs binary relevance classification, we take all positive relevance labels in our setting (+++, ++, +) as positive feedback and all negative labels as negative feedback. The position of relevant documents within the top K results does not influence the value of $P(K)$.

- **DCG at K**. The Discounted Cumulative Gain [11] $DCG(K)$ is a different measure for the quality of the top K results which does consider their order. Furthermore, it is not bound to binary relevance judgments but rewards more relevant results. It is computed as:

$$DCG(K) = \sum_{j=1}^{K} (2^{r(j)} - 1)/log(1 + j) \qquad (2)$$

  $r(j)$ corresponds to the relevance score of the j^{th} result entry. In our case, the ratings +++, ++, + lead to relevance scores of 3, 2, 1, respectively, whereas all negative ratings get a relevance score of 0.

- **MAP (at K)**. Mean Average Precision is a single value metric for a set of result lists. It is computed as the mean of average precision values for all result lists in the set. Average precision concerning one result list is calculated as the mean of the $P(K)$ values after each relevant document was retrieved. We compute MAP at K by delimitating the considered result lists to the first K entries, i.e., computing $P(i)$ only for $i \leq K$.

5. RESULTS

Overall, 32 university graduate and undergraduate students took part in the study. They were native German speakers and volunteered based on call for participation notes in public areas of the university. Each student was rewarded with 10 Euros for around 1 hour of work.

The first 17 participants took part in the first experimental session comprising the re-ranking procedure. The remaining 15 students were employed for the second session including the query expansion procedure. Altogether, they issued 192 user queries and gave relevance ratings for 3497 result list entries.

We report the results in the following way: First, the best parameter settings for the previously described implicit feedback methods are presented and discussed. Second, we analyze the performance of those methods and compare them to our baselines for both the re-ranking and the query expansion procedure. Finally, we analyze the reasons for the differences in performance more closely by directly comparing the document segments used as relevance feedback by the different feedback methods.

5.1 Evaluating Parameter Settings

Based on the data of the re-ranking procedure, we tested several parameter settings for the implicit feedback methods. Table 1 shows MAP and DCG scores at $K = 10$ after re-ranking the top 20 search results from the Web search engine by the different feedback methods. The best-performing

Table 1: MAP and DCG at $K = 10$ for different methods and parameter settings.

Variant	MAP	DCG
DsplTime(10 sec)	0.733	9.10
DsplTime(20 sec)	0.745	9.11
DsplTime(30 sec)	0.754	9.25
DsplTime(40 sec)	0.753	9.17
DsplTime(50 sec)	0.741	9.16
DsplTimeNeg(1 sec, 30 sec)	0.769	9.39
DsplTimeNeg(5 sec, 30 sec)	0.744	8.97
DsplTimeNeg(10 sec, 30 sec)	0.736	8.96
EyeTrack(0 chars)	0.747	9.14
EyeTrack(50 chars)	0.749	9.18
EyeTrack(100 chars)	0.733	9.17
EyeTrack(150 chars)	0.736	9.15
EyeTrack(200 chars)	0.734	9.06

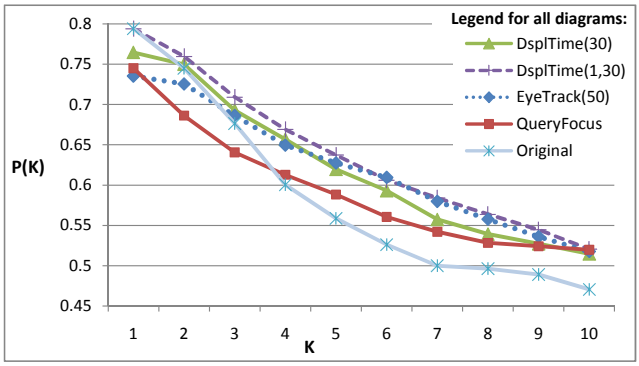

Figure 2: Precision at K with respect to re-ranking.

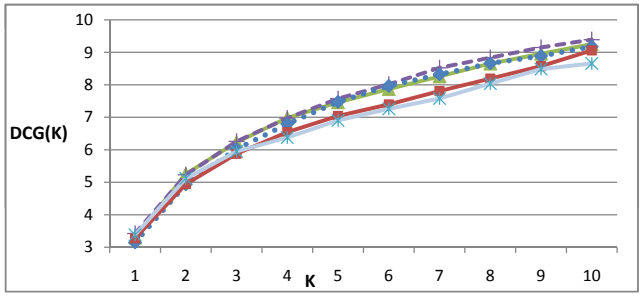

Figure 3: DCG at K with respect to re-ranking.

Table 2: MAP and DCG at $K = 10$ with respect to re-ranking and query expansion.

Variant	Re-Ranking MAP	Re-Ranking DCG	Query Expansion MAP	Query Expansion DCG
DsplTime(30)	0.754 *q	9.25 *q *o	0.771 *q	8.79 *q *o
DsplTimeNeg(1, 30)	0.769 *q	9.39 *q *o	0.758	8.89 *q *o
EyeTrack(50)	0.749 *q	9.18 *q	0.762	8.58 *o
QueryFocus	0.709	9.06	0.751	8.21
Original	0.729	8.66	0.751	8.05

parameter settings are marked in the table and are further used in the remainder of this paper.

Discussion. The best working threshold for display time in order to separate relevant from irrelevant parts of a viewed document was $t = 30$ seconds. This is in line with previous research. For example, White & Kelly [22] reported that a threshold of $t = 29$ seconds worked best for documents used in a comparable task type (at the object level, however).

Concerning the display-time-based method including negative feedback, i.e., *DsplTimeNeg*, it seems that the more of shortly viewed text parts it takes as negative feedback, the better gets the method. Our initial idea of not including too shortly viewed text parts for negative feedback was due to the higher risk for the user of missing an actually relevant passage within those parts: There is a higher chance of missing a relevant part in a text section the user has seen for just 2 seconds than in a section the user has seen for 10 seconds. However, as further analysis showed, there was no participant that completely missed reading any relevant part. Thus it was best to consider all parts of the documents as irrelevant that have been displayed for less than 30 seconds. Yet, this might not hold in real-world scenarios.

The *EyeTrack* method tends to work best with a threshold

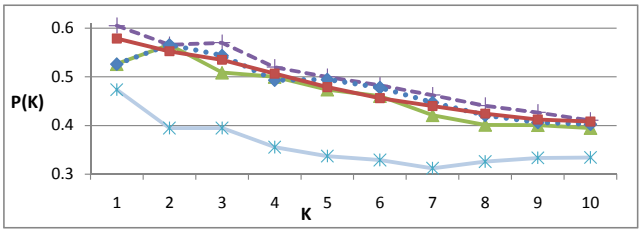

Figure 4: Precision for poorly performing queries (original MAP $\leq$ 0.7) with respect to re-ranking.

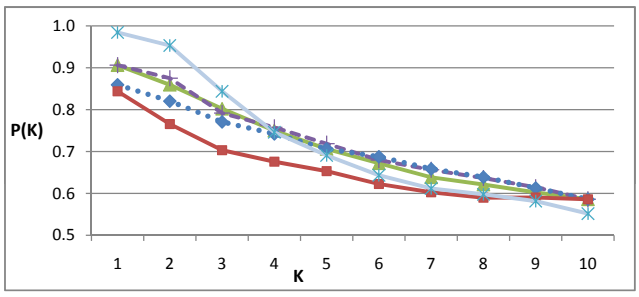

Figure 5: Precision for highly performing queries (original MAP $>$ 0.7) with respect to re-ranking.

Table 3: MAP and DCG with respect to queries with poor and high original performance.

Variant	MAP of original ranking <= 0.7 MAP	<= 0.7 DCG	> 0.7 MAP	> 0.7 DCG
DsplTime(30)	0.63 *o	6.48 *o	0.83	10.90
DsplTimeNeg(1, 30)	0.65 *o	7.11 *o	0.84	10.74
EyeTrack(50)	0.63 *o	6.63 *o	0.82	10.69
QueryFocus	0.62 *o	6.80 *o	0.76 *o	10.40
Original	0.50	5.18	0.86	10.73

of $l = 50$ characters. According to our initial assumption it seems to be an indicator of irrelevance if the user starts reading but stops within the first few words. (The content in these brackets is 50 characters long).

5.2 Comparison of Method Performance

All three methods based on user behavior, i.e., based on display time and eye tracking, perform considerably better than our *QueryFocus* baseline and could also improve the original ranking from the Web search engine. Figures 2 and 3 show precision and DCG diagrams at different recall levels K for the re-ranking procedure. Table 2 shows MAP and DCG scores at $K = 10$ for all methods, both for the re-ranking and for the query expansion procedure. Asterisks mark significance at the 0.05 level of the differences with respect to the *QueryFocus* baseline (*q) and the *Original* ranking of the user query from Live Search (*o). Significance was determined by a two-sided paired student's t-test. Differences between the behavior-based methods were not significant.

Concerning MAP, none of the behavior-based methods was significantly better than the original ranking generated by the Web search engine. Thus, to further explore the effect of our segment-level feedback methods on the original

ranking, we divided the user queries in two groups: *poorly performing queries*, i.e., user queries achieving a MAP of ≤ 0.7 in the original ranking from the Web search engine, and *highly performing queries* (MAP > 0.7). The appropriate precision diagrams are shown in Figures 4 and 5. An overview of the MAP and DCG scores at $K = 10$ is given in Table 3 with respect to re-ranking (incorporating 38 poorly and 64 highly performing queries). As before, an asterisk (*o) marks significance of the differences with respect to the original ranking from the Web search engine.

Discussion. Overall, the highest gains were achieved by the method *DsplTimeNeg*, i.e., 8.5% in MAP over the *QueryFocus* baseline using the re-ranking procedure and 10.5% in DCG over the original ranking using the query expansion procedure (both significant). When splitting the user queries in those with poor and those with high performance with respect to the original ranking from the Web search engine, the gains with respect to poor queries increase up to 31% for MAP and up to 37% for DCG at 10. With respect to well performing queries, the behavior-based methods do not seem to worsen the original ranking too much (no significant difference). However, the *QueryFocus* method does significantly decrease performance by 12%.

The last findings are somewhat in line with previous research by Agichtein et al. [1] who reported that incorporating implicit feedback is especially improving performance for poor queries while hurting queries with high original performance. However, in parts our results stand in contrast since we could not detect any significant impairment for originally highly performing queries when applying the best working behavior-based methods.

In general, display-time-based methods seem to work equally well as the eye-tracking-based one. This is surprising since the latter method uses much more precise feedback about which parts of viewed documents have been relevant to the user. It rather seems that a little more context as used by the display-time-based methods does not hurt performance. Using non relevant parts of viewed documents as negative feedback can further improve performance as demonstrated by the *DsplTimeNeg* method.

5.3 Comparison of Feedback Segments

All of our used implicit feedback methods focus on certain segments of the viewed documents. In order to understand their different performances noted above concerning re-ranking and query expansion we further analyzed the specific document parts they focused on for generating feedback.

Figure 6 depicts the segments focused by the different methods for one of the four viewed preselected documents used in the task about *thermoregulation*. On the left, a small thumbnail of the full document is shown. The dotted rectangle at the top illustrates the size of the browser viewport. The solid rectangle towards the end of the document marks the only directly relevant segment with respect to the task in which is was used. The reading and viewing behavior of three typical participants is shown on the right. The first column "e" in each box marks the parts of the document that have been detected as read and thus that were applied as implicit feedback by the *EyeTrack(50)* method. Column "d" displays the feedback segments used by the method *DsplTime(30)*. Column "t" represents the display times in continuous format, i.e., not applying a threshold of 30 seconds. Participant 1 did already look through the document before because *perception* was her first task topic while *thermoregulation* was her second (remember that both tasks came with the same set of preselected documents); participants 2 and 3 were first-time-viewers of the document. Accordingly, participant 1 views the document in a more goal-directed way whereas participants 2 and 3 are more involved looking through different sections.

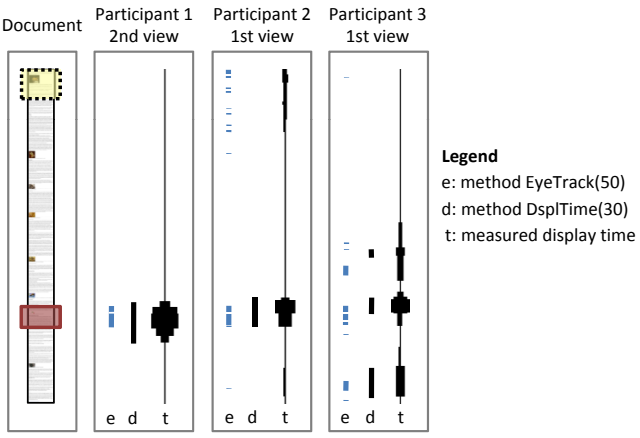

Figure 6: Example illustration of feedback segments.

Table 4: Overlapping between segments used by the different feedback methods.

w(e∩d) / w(e) = 81%	w(r∩e) / w(r) = 86%
w(e∩d) / w(d) = 55%	w(r∩d) / w(r) = 96%
w(r∩e) / w(e) = 69%	w(r∩f) / w(r) = 15%
w(r∩d) / w(d) = 52%	w(r∩f) / w(f) = 23%

Legend
e: EyeTrack(50)
d: DsplTime(30)
f: QueryFocus
r: relevant

Table 4 presents a more in-depth quantification of the overlapping between segments used by the different feedback methods. Here, "e", "d", and "f" stand for sets of document segments used as implicit feedback by the methods *EyeTrack(50)*, *DsplTime(30)*, and *QueryFocus*, respectively. "r" represents the set of segments that have been marked as being directly relevant to the task by the conductor of the study. The function $w(.)$ returns the number of words in a given set of segments. For example, the two entries in the upper left of Table 4 state that 81% of the document segments used as feedback by the method *EyeTrack(50)* were also used by the method *DsplTime(30)*. However, only 55% of the feedback parts used by *DsplTime(30)* were also used by *EyeTrack(50)*.

Discussion. The comparison of segments used for extracting characteristic terms by the different feedback methods provides some insights into why the methods perform as shown in the previous section. The segments focused by the *QueryFocus* method did not have a high overlap with the relevant sections in the document: on average only 15% of the relevant text was focused by that method, while only 23% of the focused text was relevant. The respective overlap concerning the behavior-based methods is considerably higher, which explains their improved performance with respect to re-ranking and query expansion. Comparing the methods *DsplTime(30)* and *EyeTrack(50)* the results show that the latter is more focused on the relevant parts of the document while the former is generally broader and uses larger text segments as basis for term extraction. However,

it is not the case that method *DsplTime(30)* completely incorporates all parts used by *EyeTrack(50)*. Yet, we believe that display time might be a well-working replacement for eye-tracking-based evidence, especially when creating more sophisticated display-time-based feedback mechanisms than used in this study. In general, the experiment shows that using rather coarse display time instead of fine-grained gaze-based feedback apparently but surprisingly does not hurt retrieval performance.

6. CONCLUSION

The results of this study show that segment-level display time can be as valuable as eye-tracking-based feedback, both concerning re-ranking of search results and query expansion. All implicit feedback methods based on display time and eye tracking data significantly outperform a baseline utilizing the user query for pseudo relevance feedback on the paragraph level. Precision can further be improved when interpreting document parts that were viewed very quickly by the user as negative feedback.

With respect to the original ranking from a major Web search engine, the best implicit feedback method achieved gains in MAP of up to 31% for initially poorly performing user queries. Popular commercial Web search engines can be assumed to be highly specialized for certain kinds queries containing numerous heuristics for their optimization, so that it is important not to worsen performance for initially highly performing queries. Nevertheless, for those highly performing queries, we could not detect any significant general impairment when applying implicit feedback.

As our more in-depth analysis shows, feedback based on display time is coarser than feedback based on eye tracking, meaning that the former method interprets more document segments as positive feedback than what the user has actually read. However surprisingly, this does not hurt performance with respect to re-ranking or query expansion.

Of course it has to be born in mind that we examined segment-level display time in a very controlled laboratory study for one specific task type. In real-world scenarios it can be assumed that there will be much more noise in the data, for example originating from time periods during which the user is not looking at the screen at all. Thus, it should be ensured that display time is only measured when the user is looking at the document. In this respect, inexpensive face detection by a Web cam might be a very promising addition in order to accurately record and interpret display time in real-world settings.

7. ACKNOWLEDGMENTS

We thank Susan Dumais for a nice idea concerning evaluation and the anonymous reviewers for their valuable comments. This work was supported by the German Federal Ministry of Education, Science, Research and Technology (bmb+f), (project Mymory, grant 01IWF01; project Perspecting, grant 01IW08002).

8. REFERENCES

[1] E. Agichtein, E. Brill, and S. Dumais. Improving web search ranking by incorporating user behavior information. In *SIGIR '06*, pp. 19–26, 2006.

[2] J. W. Ahn, P. Brusilovsky, D. He, J. Grady, and Q. Li. Personalized web exploration with task models. In *WWW '08*, pp. 1–10, 2008.

[3] I. Arapakis, J. M. Jose, and P. D. Gray. Affective feedback: an investigation into the role of emotions in the information seeking process. In *SIGIR '08*, pp. 395–402, 2008.

[4] N. J. Belkin. Some(what) grand challenges for infor- mation retrieval. *SIGIR Forum*, 42(1):47–54, 2008.

[5] G. Buscher, A. Dengel, and L. van Elst. Eye movements as implicit relevance feedback. In *CHI '08: CHI '08 extended abstracts on Human factors in computing systems*, pp. 2991–2996, 2008.

[6] G. Buscher, A. Dengel, and L. van Elst. Query expansion using gaze-based feedback on the sub- document level. In *SIGIR '08*, pp. 387–394, 2008.

[7] M. Claypool, P. Le, M. Wased, and D. Brown. Implicit interest indicators. In *IUI '01: Proc. of the 6th intl. conf. on Intelligent user interfaces*, pp. 33–40, 2001.

[8] S. Fox, K. Karnawat, M. Mydland, S. Dumais, and T. White. Evaluating implicit measures to improve web search. *ACM Transactions on Information Systems*, 23(2):147–168, 2005.

[9] G. Golovchinsky, M. N. Price, and B. N. Schilit. From reading to retrieval: freeform ink annotations as queries. In *SIGIR '99*, pp. 19–25, 1999.

[10] K. Gyllstrom and C. Soules. Seeing is retrieving: building information context from what the user sees. In *IUI '08: Proc. of the 13th intl. conf. on Intelligent user interfaces*, pp. 189–198, 2008.

[11] K. Järvelin and J. Kekäläinen. Ir evaluation methods for retrieving highly relevant documents. In *SIGIR '00*, pp. 41–48, 2000.

[12] T. Joachims, L. Granka, B. Pan, H. Hembrooke, F. Radlinski, and G. Gay. Evaluating the accuracy of implicit feedback from clicks and query reformulations in web search. *ACM Transactions on Information Systems (TOIS)*, 25(2), April 2007.

[13] D. Kelly and N. J. Belkin. Reading time, scrolling and interaction: exploring implicit sources of user preferences for relevance feedback. In *SIGIR '01*, pp. 408–409, 2001.

[14] D. Kelly and N. J. Belkin. Display time as implicit feedback: understanding task effects. In *SIGIR '04*, pp. 377–384, 2004.

[15] D. Kelly and J. Teevan. Implicit feedback for inferring user preference: a bibliography. *SIGIR Forum*, 37(2):18–28, 2003.

[16] M. Melucci and R. W. White. Discovering hidden contextual factors for implicit feedback. In *Proc. of the CIR'07 Workshop on Context-Based Information Retrieval in conjunction with CONTEXT-07*, 2007.

[17] M. Morita and Y. Shinoda. Information filtering based on user behavior analysis and best match text retrieval. In *SIGIR '94*, pp. 272–281, 1994. Springer.

[18] D. W. Oard and J. Kim. Modeling information content using observable behavior. In *Proc. of the 64 Annual Meeting of the American Society for Information Science and Technology*, pp. 38–45, 2001.

[19] R. Rafter and B. Smyth. Passive profiling from server logs in an online recruitment environment. In *ITWP 2001: Proc. of the IJCAI Workshop on Intelligent Techniques for Web Personalization*, 2001.

[20] J. Rocchio. *The SMART Retrieval System: Experiments in Automatic Document Processing*, chapter Relevance feedback in information retrieval, pp. 313–323. Prentice Hall, 1971.

[21] J. Salojärvi, K. Puolamäki, and S. Kaski. Implicit relevance feedback from eye movements. In *ICANN'05*, pp. 513–518, 2005.

[22] R. W. White and D. Kelly. A study on the effects of personalization and task information on implicit feedback performance. In *CIKM '06: Proc. of the 15th ACM intl. conf. on Information and knowledge management*, pp. 297–306, 2006.