Improving Passage Ranking with User Behavior Information

Weize Kong, Elif Aktolga and James Allan
Center for Intelligent Information Retrieval
School of Computer Science
University of Massachusetts Amherst
Amherst, Massachusetts
{wkong, elif, allan}@cs.umass.edu

ABSTRACT

User behavior information has proved valuable for inferring document relevance, but its role in deducing relevance at the passage/section level is not well explored. In this paper, we study how user behavior information implies section relevance, and use this information to improve section ranking. More specifically, we focus on four types of user search behavior that occur while browsing a document – dwell time, highlighting, copying and clicks at the section level. Experimental results based on a commercial query log show that user behavior information can significantly improve section ranking. While section-level click information is a very powerful signal of relevance, it depends on an interface supporting section-level links. We find comparable levels of gain using other behavior information that does not depend upon such an interface.

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

General Terms: Experimentation, Human Factors

Keywords: Passage Retrieval, User Behavior, Dwell Time

1. INTRODUCTION

It is widely known that user interaction with information retrieval systems can provide valuable information to improve effectiveness – for example, queries and their click patterns can be mined to associate web pages with queries for which they are likely to be relevant.

In this study we explore what happens when those ideas are extended from full document retrieval to section or passage retrieval. We consider the case when a section or paragraph of a document may better address a user's information need than its containing document. What happens if we combine document-level click information with section-level ranking? What additional value is there in click details at the *section* level? In cases where section-level clicking is cumbersome or inappropriate, is there other information that could be used to the same effect?

CIKM'13, Oct. 27–Nov. 1, 2013, San Francisco, CA, USA.

ACM 978-1-4503-2263-8/13/10 ...$15.00.
http://dx.doi.org/10.1145/2505515.2505719.

This issue has received little attention to date, partly because not all data is organized such that section retrieval makes sense and partly because few retrieval systems are instrumented to collect user interaction data at that level of granularity. We are fortunate to have access to a collection of hierarchically organized documents with browsing and click information recorded at the document and section level, allowing us to explore and evaluate the utility of that information.

In the first part of this study (Section 4), we use the browsing and click information to analyze user behavior at the section level. Our analysis focuses on four types of behaviors occurring while browsing a document: the time during which a section is displayed ("dwell time"), highlighting or copying a portion of text, clicks on links between and within documents, and clicks on section links directly in response to the query. The analysis gives insight into passage level user behavior, reveals existing biases, and testifies its potential in estimating passage relevance. We find that many section dwell times (50% according to our dataset) are shorter than 2 seconds, suggesting users skim many sections instead of reading them. We also discover that a bias exists for section dwell time, which is due to non-relevant sections being adjacent to relevant sections that are displayed together. For clicks, we find a position bias due to the structure of the documents.

In the second part of the study (Sections 5 and 6) we evaluate the value of logged behavior to improve the ranking of sections in response to a query. Based on the earlier analysis, we select a set of features that are indicative of relevance and use them to rank sections. We consider each user behavior feature individually as well as in a group. We confirm our hypothesis that direct query-to-section clicks are most effective among the four types of user behaviors. Given the value of click information at the document level, that result is not at all surprising. However, we also show that if section-level click information cannot be included – e.g., if there is no appropriate way to provide links to sections – we can achieve half of the gain of section level clicks using browsing-based features alone.

2. RELATED WORK

Passage retrieval has a long history within Information Retrieval: improving document retrieval [6, 26, 27, 31, 36], focusing query expansion [41], extracting explanatory snippets, locating responses for question answering [4, 7, 18, 37], and retrieving appropriate passages [5, 36, 38, 40]. Our

study focuses on retrieving passages that are marked as sections of documents and employing user behavior information to improve accuracy.

There is a large body of work studying user search behavior or "implicit measures" at the document-level: Guo and Agichtein [23] estimate document relevance from cursor movements and scrolls in addition to dwell time and other previously studied user behavior features. Features from user behavior analysis particularly targeted towards web search were also investigated [1, 2].

The question of how to employ user search behavior information for estimating *passage* relevance and improving *passage* ranking is largely unexplored. The closest work that we find is done by Buscher et al., in which they studied segment-level display time and segment-level feedback from an eye tracker [10, 11, 12, 13]. They hypothesized which parts of documents being read are likely to be relevant, and used those for query expansion and re-ranking of documents. While their results showed potential for inferring passage relevance by means of user behavior, they did not investigate that problem. One other related work at the segment level focuses on text highlighting as a form of relevance feedback [22].

Several user studies have been performed for analyzing the link between relevance and dwell-time [16, 28, 29, 35, 39]: in particular, these studies focus on the correlation between reading time and explicit feedback. Fox et al. [20] analyze the relationship between explicit measures (such as relevance judgments) and implicit measures (user behavior information). For this, they also looked at a sequence of characteristic user behavior patterns. Some earlier work also utilizes dwell distributions for collaborative filtering to predict user ratings [34]. All of that work was done at the document level, whereas we focus here on section-level dwell time.

Learning to rank is a well-known supervised technique from machine learning and information retrieval for learning a function that tries to optimally rank a list of documents or passages. There are three major approaches: point-wise, pair-wise, and list-wise [14, 15]. As a point-wise approach we use RandomForests [8], a decision tree approach for which a fully grown unpruned tree is built during the training phase. For pair-wise approaches, where training instances are learned together in pairs, we use Ranking SVMs [24] and RankBoost [21]. Finally, list-wise approaches directly learn a ranked list by optimizing an evaluation measure. We use AdaRank [42] and Coordinate Ascent [33].

3. MOTIVATION

The motivation for this research arose during discussions with a medical informatics company, UpToDate. UpToDate hosts medical information that is searched on a daily basis by a large number of physicians. UpToDate's search engine is based on the open source Apache Lucene system[1]. It has been extensively tuned over several years, using parameter sweeps, evaluations on subsets of users, careful editing of the hosted information, and human intuition. A user's query is converted into a complex weighted combination of the original query words, synonyms from a controlled vocabulary, and field references.

[1]http://lucene.apache.org/java/docs/index.html

The collection of documents is small from the perspective of IR research or Web search: it comprises just under 16,000 high quality documents and is constantly being updated by several experts in the field (physicians) according to the most recent medical findings. These documents are hierarchically arranged into sections and subsections. Figure 1 illustrates a document and its hierarchically arranged sections: section H1.2.2 is contained in section H1.2, which is in turn contained in H1.

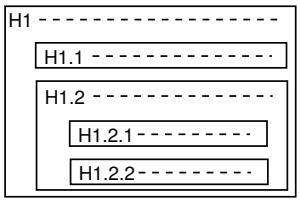

Figure 1: Hierarchically arranged sections of a document in our collection.

Like most search engines, UpToDate collects clickthrough information from its users; this work relies on an anonymized subset of this data.[2] It is well known that such information can be used to improve *document* retrieval effectiveness. However, UpToDate collects *section*-level click information as well as document-level clicks. From click behavior analysis (Section 4), it is evident that certain sections are heavily favored over others. These are particularly sections summarizing the content of a document or dosage information about a certain drug. In a non-medical domain, these could be sections summarizing a product or describing specifics about a technique, recipe, or recommendation. Since UpToDate's current system only ranks documents, it seems that it would be beneficial if the results could be displayed at the section level so that scrolling through the document becomes unnecessary when appropriate.

Currently, the *section*-level clicks are collected in several ways. In response to a query, a user may click on a document in the ranked list of results, or may hover over the title, causing an outline listing of all the headings with the document to be displayed. A *rollover click* is triggered when the user clicks a section heading in that preview rollover panel. When a document is displayed – whether because it was selected directly or a section was chosen – the outline is displayed to the left of the document. An *outline click*, then, is triggered through a click on that outline. Finally, a *see link* is logged when a user clicks on a link within a section's text leading to another section (e.g., "see also. . . "). This type of click is the only one that does not require an outline structure in documents.

In addition to click information, our anonymized log includes other search behavior information: every half second the log captures which text is highlighted by the mouse and whether the user has "copied" the highlighted text. From this we can determine the time a user spends with a single section displayed, or the *dwell time* for that section. We stress that these actions do not require an outline structure

[2]Although it receives a substantial number of queries each day, the volume is small compared to major search engines: our collection contains around 18 million queries per month compared to the more than 13 billion queries Google is estimated to have handled this past March [17].

in the documents; they do not necessitate changes to the interface, and can be logged in the background by the browser using JavaScript or similar technologies.

Our goal is to explore the impact of section level features from four different classes of user search behavior information: dwell time on sections, highlighting or copying of text within sections, "see link" clicks, and outline or rollover clicks. The baseline search system we use is a carefully crafted Lucene implementation incorporating various scores such as tf-idf and document-level clickthrough information. This means we are constrained to combining a carefully crafted Lucene score with hierarchical content and user behavior information from sections – in the same way that much learning-to-rank research includes system-level scores such as BM25 and tf·idf. We leave for future work approaches that expose the inner workings of the Lucene black-box system and integrate specific content information more elaborately.

Before diving into the details of learning better section retrieval, we first present an analysis of search behavior on our anonymized query log to understand the underlying data and to be able to do better feature engineering.

4. USER BEHAVIOR ANALYSIS FOR SECTION RETRIEVAL

In this section we present an analysis of the different types of section level behaviors. First we explore dwell time, and then we move on to highlight and copy operations. Finally, we analyze the three types of section-level clicks. The analysis is based on the query log with the training query set, described in Section 6.1.

4.1 Dwell

4.1.1 Dwell distribution

Every half second, our log notes which sections are displayed on the screen. Given this information, we extract section dwell times, which capture the time when a section starts being displayed until the section is not rendered in the user's screen anymore. We compare the section dwell time distribution to the document dwell time distribution in Figure 2. Similar to Liu et al. [32], document dwell times are extracted as the time difference between the opening and closing actions to a document. The closing action is performed either by closing the page, issuing a new query, or by clicking and being directed to other documents.

We can see from Figure 2 that the document dwell distribution is skewed in our data set like in other studies [32, 30]. Moreover, we find that the section dwell distribution is even more skewed. The cumulative probability achieves 50% when the dwell time is 2 seconds. This indicates that many section dwell times are extremely short, which are more likely to be generated by skimming sections than actually reading them. 96% of section dwells are short – less than 100 seconds – while there are only 58.5% of document dwells that are less than 100 seconds.

The right part of Figure 2 shows the dwell distribution using frequencies with dwell times binned by 1 second. Sections have many short dwells, and starting from around 145 seconds, documents have more long dwells than sections. The sudden drop for section dwell frequency in the Figure is at 960 seconds, from frequency 2256 to 1858, which is

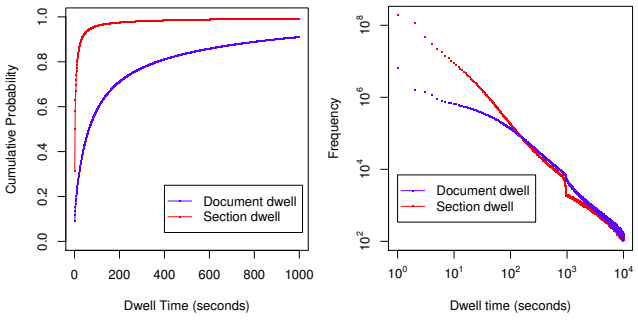

Figure 2: Document and section dwell time distribution. Left: cumulative distribution. Right: frequency distribution.

actually exaggerated by the log-scale. A difference of 398 in frequency is actually small for this 4 month data set. It could have been generated by one single user, who used the system in around 16-minute intervals, which is likely to be the average meeting time for a patient.

4.1.2 Dwell and Section Relevance

Similar to the assumption that long dwell times indicate (document) relevance [16, 20, 35], we hypothesize that long section dwell times indicate section relevance. We test this by using two ways of aggregating section dwell times across different users when searching for the given query - **cumulative dwell time** and **average dwell time**. Cumulative dwell time sums up all the dwells for a section given the query across different users, while average dwell normalizes cumulative dwell time by the number of dwells.

In Figure 3, we show the cumulative dwell time and average dwell time for sections of different relevance ratings. Clearly, sections of higher relevance ratings have higher dwell times according to the mean value, for both cumulative dwell time and average dwell time, from ratings 1 to 4. But the trend is reverse from ratings 4 to 5, which indicates a disagreement between dwell times and our human judgment for "perfect relevance".

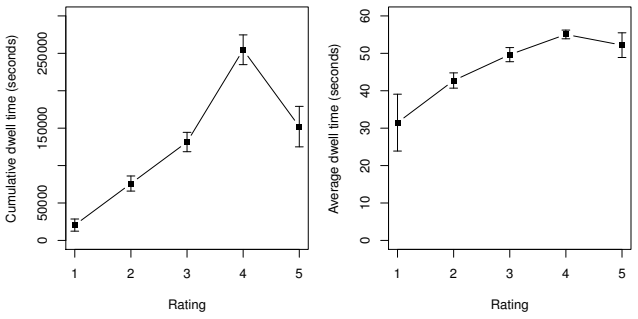

Figure 3: Dwell time for sections of different relevance ratings. Error bars show 95% confidence intervals. Left: cumulative dwell time. Right: average dwell time.

In the right part of Figure 3 we can see that users dwell around 48 to 55 seconds on average for relevant sections, suggesting that this is likely to be the average time users spend on reading relevant sections in our data set. Inspired

by this, we define a relevant section view to be a section dwell that is longer than a certain threshold. We tune the threshold to optimize MAP of ranking results based on the relevant view frequency using our train queries. As shown in Figure 4, the optimal threshold is 49 seconds, consistent with the average dwell time for relevant sections in Figure 3.

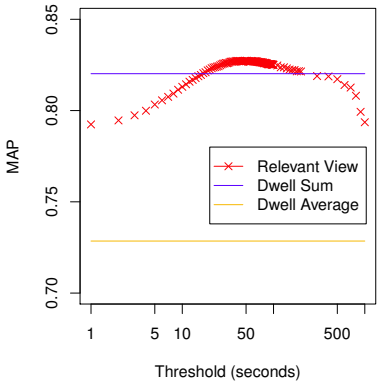

Figure 4: MAP for relevant view frequency using different thresholds, cumulative dwell time, and average dwell time.

In Figure 4, we also compare the effectiveness of ranking sections using the three ways of aggregating dwells. As we can see, cumulative dwell time and relevant view frequency are much better than average dwell time in ranking sections. Relevant view frequency is slightly better than cumulative dwell time when using the optimal threshold.

4.1.3 Dwell Bias

Even though a correlation exists between dwell time and section relevance, we also find a bias for section dwells. When users read a relevant section, often adjacent sections are also being displayed on the screen. Therefore, dwell times at a relevant section also affect the dwell times for its adjacent sections. We test this adjacent section dwell bias in Figure 5, which shows the cumulative dwell time for non-relevant sections at difference distances from their closest relevant section in the document. We measure distance between two sections by the number of sections/subsections between them, e.g. sections right adjacent to each other have distance 1. Clearly, non-relevant sections right next to a relevant section have a much higher dwell time than sections at further distance, proving the existence of an adjacent section dwell bias.

4.2 Highlight and Copy

While reading a document, users sometimes highlight or copy parts of the content in the document, which is very likely to be relevant to users' information needs as shown by Golovchinsky et al. [22] in a controlled laboratory setting. Following this idea, we study the frequency of a section being highlighted/copied when searching for a given query.

Figure 6 shows section copy and highlight frequencies of documents while searching for a query in our training set, where each point represents a section and a query. The x-axis shows the copy frequency of this section when users search for the query, and the y-axis stands for the highlighting frequency. We can see that highlights happen much

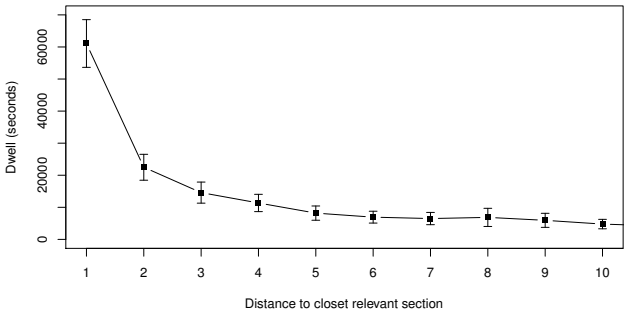

Figure 5: Dwell time of non-relevant sections at distance 1 to 10, from closet relevant section. Error bars show 95% confidence intervals.

more often than copies. For sections of copy frequency 50, the range of highlight frequency is 151 to 841. The figure also suggests a strong positive correlation between highlight and copy actions. The Pearson's correlation coefficient between copy and highlight frequencies is 0.79.

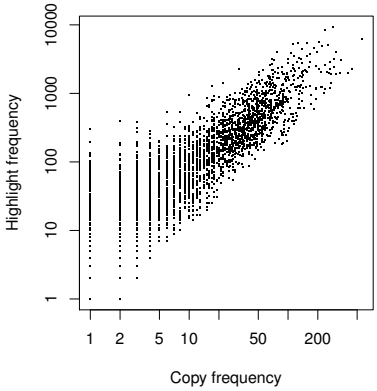

Figure 6: Copy and highlight frequencies of documents given a query in the training set.

We test if copy and highlight frequency information implies section relevance in Figure 7, which shows highlight/copy frequencies for sections of different relevance ratings. The figures look much like Figure 3, in which we see a clear increase in highlight/copy frequencies from ratings 1 to 4, but a drop from 4 to 5.

4.3 Clicks

Section clicks and document clicks are different in their sources. Document clicks happen when users are browsing document ranking, while sections clicks happen when users are browsing within a document (for outline click and seelink click), or reading the rollover panel (for rollover clicks in our system). Despite the difference in sources, both document clicks and section clicks indicate to some level that users are interested in the document/section.

We test if the three types of clicks – outline, seelink and rollover – indicate section relevance in Figure 8. The Figure shows the click frequency of the three types of clicks for sections of different ratings. The trend for rollover clicks is similar as for dwell time in Figure 3 and highlight/copy in Figure 7 – we can see an increase from ratings 1 to 4,

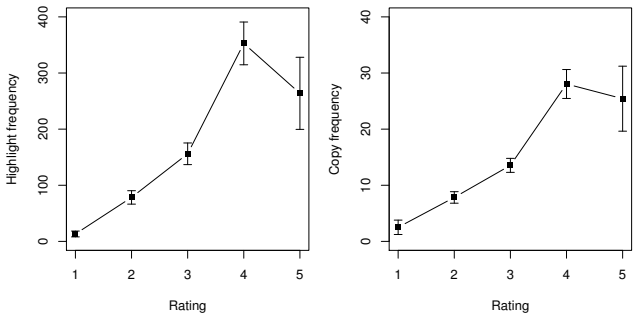

Figure 7: Highlight/copy frequencies versus ratings. Error bars show 95% confidence intervals. Left: highlight frequency. Right: copy frequency.

and a drop from ratings 4 to 5. Differently, outline clicks increase from ratings 1 to 5 according to the mean value, although the variation of outline clicks for sections of rating 5 is very high. Seelink clicks do not clearly correlate with the relevance ratings.

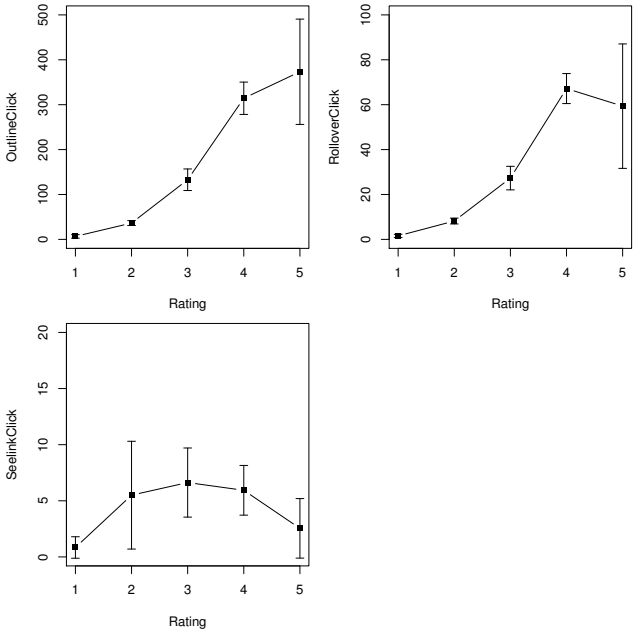

Figure 8: Outline/rollover/seelink Click and rating. Error bars show 95% confidence intervals. Top left: outline click. Top right: rollover click. Bottom left: seelink click.

Similar to the document click bias [3], we find that clicks can be biased by the section position in the document. In Figure 9, we plot the outline and rollover click distribution over section positions, as well as the section distribution over positions, which represents the click distribution when the likelihood of a section being clicked is equal for all sections. We can see that except for the sections at the top of the document, outline and rollover click distributions are close to the section distribution. However, outline clicks at the top 2 sections are far less than the distribution of sections. This is probably because when opening a document, the top part of the document is already displayed, and therefore there is no need for users to click on the outline to get to top sections.

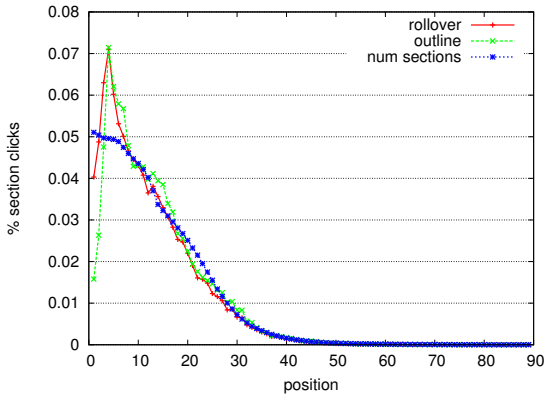

Figure 9: Percentage of sections clicks whose sections have a certain position in the document.

5. FEATURES FOR SECTION RANKING

In this section we summarize the features used for the experiments.

5.1 Section Content-based Features

Section content-based features include features that only depend on the section content (and the query), listed in the first part of Table 1. The most obvious content feature is luceneScore, which is the raw Lucene score for the section as obtained by the search engine. Further, we measure the percentage of overlapping title 1-5-grams between the section and the document titles with the feature titleOverlap. referenceProb utilizes reference or link counts between sections: it refers to the probability of the section being linked to by other sections with respect to all the sections in all the documents in the corpus. This is very similar to PageRank scores[9].

In our corpus, a section can be at three levels; level 1 indicates that it is a section at the top level; level 2 means it is a subsection with 1 parent section at level 1; and level 3 means that it is a subsubsection having 1 parent section at level 2 and another one at level 1 (see Figure 1). The feature level refers to the raw level numbers 1, 2, or 3.

Further, a section has a certain position within the document, which describes the order of its placement with respect to the other sections in the document. The feature position denotes a raw position number, which in our corpus is between 1 and 89 (see Figure 9).

5.2 Document-based Features

Sections from relevant documents are more likely to be relevant. Based on this intuition, we use two document-based features for ranking sections, as described at the bottom of Table 1. docLuceneScore is the raw Lucene score for the document containing the section, obtained from the search engine. docClickProb is the likelihood for that document to be clicked given a query.

Table 1: Section content-based and document-based features.

Section content	
luceneScore	section Lucene Score
titleOverlap	overlap between section and document title
referenceProb	section reference probability
level	section level
position	section position
Document content	
docLuceneScore	document Lucene Score
docClickProb	probability of clicking the document for this query

5.3 User Behavior Features for Ranking Sections

Following our analysis on Section 4, we employ user behavior features based on dwell, highlight, copy and click actions for ranking sections. First, we describe the notation used for features. Let $f(\cdot)$ denote a type of user behavior features, q denote an issued query, s denote a section and d denote the document which s belongs to. Then, we can use $f(s,q)$ to denote the value of feature f for section s when searching with query q. $f(d,q) = \sum_{s \in d} f(s,q)$ denotes the sum of all the feature values for sections in document d, and $f(q) = \sum_d f(d,q)$ denotes the sum of all feature values for query q. As an example, let the copy frequency (described in Section 4) be expressed as $copy(\cdot)$. Then $copy(s,q)$ is the copy frequency of section s while searching with query q, $copy(d,q)$ is the total copy frequency of this document given the query, which is the sum of copy frequencies of all the sections in document d given that query, and $copy(q)$ is the total copy frequency when searching with the query, summing up all the copy frequencies for sections in search sessions for that query.

Generally, we have three ways of aggregating/normalizing a type of user behavior feature f. 1) The probability given the query, $P_f(s|q) = \frac{f(s,q)}{f(q)}$. This is simply a way of normalizing to avoid using the absolute value. For example, copy probability (copyProb) is the copy frequency for the section and query, normalized by the total copy frequency for the query, or more formally $P_{copy}(s|q)$; 2) To capture the relevance of sections compared to other sections in the same document, we use the probability given the query and document as another way of normalization, which is denoted as $P_f(s|d,q) = \frac{f(s,q)}{f(d,q)}$. For example, copy probability given the document (copyDocProb) can be denoted as $P_{copy}(s|d,q)$. It is the copy frequency for the section and query normalized by the total copy frequency for the query and the document which the section belongs to. 3) To cope with the variation of the feature value over different queries, we use the deviation from the average value for the query, $\sigma_f(s,q) = f(s,q) - \frac{f(q)}{|S|}$, where S is the set of sections that are used to sum up $f(s,q)$ for $f(q)$. For example, copy deviation (copyDevi) is the deviation of copy frequency from the average copy frequency for the query.

We summarize the section level user behavior features employed in Table 2. Feature types cumuDwell and avgDwell are cumulative and average dwell times described in Section 4.1.1. For relevant views (relView), we used 49 seconds as a threshold, which is tuned on our training queries (see Section 4). hilite and copy stand for highlight and copy frequencies for sections. OutlineClick, rolloverClick and seelinkClick are click frequencies for the three corresponding click types. allClick is the click frequency for all section clicks, including clicks of all the three types.

6. EXPERIMENTS

6.1 Data set

We use four months of (anonymized) query log data for feature extraction and analysis. We sample 100 queries from the query log, and randomly split them into 50 queries for training and 50 unseen queries for test. To avoid biasing parameters and training toward certain queries, the training queries are distinct from the test queries for which the results are reported, and only train queries are used for our analysis in Section 4. All training and test queries are popular, i.e., each query has at least 500 section clicks. We did not include tail queries since there was a considerable disagreement between the judgments of our medical expert and the clicks of such queries.

To have a wide spectrum of sections for the evaluation, we pooled them from five ranking lists – sections ranked by Lucene, section clicks, cumulative dwell time, highlight frequency and copy frequency. The top ranked unique sections were pooled from each of these five lists until 100 unique sections were collected.

Our aim in ranking the Lucene section results is to better approximate how a medical expert would rank them. Hence, as truth data we use judgments provided by such an expert. For training, there are 5012 sections judged in total, whereas for test there are 5001 judged sections. So on average each query has around 100 judged sections. Table 3 shows a breakdown of the judgments for each label. We follow the PEGFB scale for these judgments labels, i.e., label 5 means 'perfect', 4 means 'excellent', 3 means 'good', 2 means 'fair', and 1 means 'bad', which the medical expert used for judging the sections.

Table 3: Human judgments for train and test.

Label	#judgments for train	#judgments for test
1	206	227
2	1541	1560
3	1855	1520
4	1269	1572
5	141	122
sum	5012	5001

6.2 Evaluation

We tried various point-wise, pair-wise and list-wise learning to rank algorithms for our experiments, such as RandomForests, RankBoost, AdaRank and Coordinate Ascent from RankLib [19]. Many of the models yield the same kind of results, so in Section 6.3 we present the results for two models only, RankBoost and RandomForests.

Given a query Q, in order to evaluate its ranked list of n sections R, we use NDCG and Precision. These are well-known measures for learning to rank. NDCG is defined as follows [25]:

$$\text{NDCG(Q, R)} = \frac{1}{Z_Q} \sum_{i=1}^{n} \frac{2^{rel_i} - 1}{\log_2(1+i)} \quad (1)$$

Table 2: User behavior features for ranking sections

Dwell	
cumuDwellProb	cumulative dwell probability, $P_{cumuDwell}(s\|q)$
cumuDwellDocProb	cumulative dwell probability given the document, $P_{cumuDwell}(s\|d,q)$
cumuDwellDevi	deviation from the average cumulative dwell time for the query, $\sigma_{cumuDwell}(s,q)$
avgDwell	average dwell time, $avgDwell(s,q)$
avgDwellDevi	deviation from the mean of average dwell time for the query, $\sigma_{avgDwell}(s,q)$
relViewProb	relevant view probability, $P_{relView}(s\|q)$
relViewDocProb	relevant view probability given the document, $P_{relView}(s\|d,q)$
relViewDevi	deviation from the average relevant view frequency for the query, $\sigma_{relView}(s,q)$
Highlight	
hiliteProb	highlight probability, $P_{hilite}(s\|q)$
hiliteDocProb	highlight probability given the document, $P_{hilite}(s\|d,q)$
hiliteDevi	deviation from the average highlight frequency for the query, $\sigma_{hilite}(s,q)$
Copy	
copyProb	copy probability, $P_{copy}(s\|q)$
copyDocProb	copy probability given the document, $P_{copy}(s\|d,q)$
copyDevi	deviation from the average copy frequency for the query, $\sigma_{copy}(s,q)$
Click	
outlineClickProb	outline click probability, $P_{outlineClick}(s\|q)$
outlineClickDocProb	outline click probability given the document, $P_{outlineClick}(s\|d,q)$
outlineClickDevi	deviation from the average outline click frequency for the query, $\sigma_{outlineClick}(s,q)$
rolloverClickProb	rollover click probability, $P_{rollover}(s\|q)$
rolloverClickDocProb	rollover click probability given the document, $P_{rollover}(s\|d,q)$
rolloverClickDevi	deviation from the average rollover click frequency for the query, $\sigma_{rollover}(s,q)$
seelinkClickProb	seelink click probability, $P_{seelink}(s\|q)$
seelinkClickDocProb	seelink click probability given the document, $P_{seelink}(s\|d,q)$
seelinkClickDevi	deviation from the average outline seelink frequency for the query, $\sigma_{seelink}(s,q)$
allClickProb	click probability for all types of section clicks, $P_{allClick}(s\|q)$
allClickDocProb	click probability given the document for all types of section clicks, $P_{allClick}(s\|d,q)$
allClickDevi	deviation from the average section click frequency, $\sigma_{allClick}(s,q)$

where i is the i^{th} section in R and rel_i is the relevance grade of section i. This measure is normalized with Z_Q so that NDCG=1.0 when ranking is perfect. Precision is measured as follows:

$$\text{P@n} = \frac{1}{n}\sum_{i=1}^{n} rel(s_i) \tag{2}$$

where $rel(s_i)$ is a binary relevance judgment for section s_i. Since our relevance judgments are graded, we set $rel(s_i) = 1$ if the judgment label is ≥ 3 and $rel(s_i) = 0$ otherwise.

In the experimental results following in the next section we typically evaluate NDCG and Mean Average Precision at rank 100.

6.3 Results

In this section, our experimental results on the test set are first compared using individual features that were described in Section 5. Then, we use incremental feature sets to analyze the gains.

6.3.1 Using Individual Features

Table 4 shows the results ranked in increasing order of MAP. Deviation features are excluded here because they are rank-equivalent to the corresponding probability features (i.e., for instance ranking based on copyProb is the same as ranking based on copyDevi) although they may have a different effect in learning to rank models. Feature luceneScore at the top represents the raw section Lucene scores for sections. Not surprisingly, it also performs the poorest when compared to all the other features with a MAP of 0.5941 and an NDCG of 0.7610. At the other end – the bottom of Table 4 – we have the strongest features with a maximum MAP of 0.8928 and NDCG of 0.9096, which are related to clicks collected through an outline interface. Certainly, if such an interface is set up, clicks yield the strongest signal and should be employed. But if this is not the case, which user search behavior features are the strongest? hiliteProb, relViewProb, and copyProb are among the best scoring single features. When comparing the amount of click actions on sections in our training data set to copy and highlight actions, it forms the majority among the three with about 54%. Highlighting is the next common action representing about 40% of all actions. Copying a portion of text is rare – only about 6% of the three. Given this, it is nice to see that copyProb is among the strongest features, confirming that it is a sparse but trustworthy feature. Highlighting a portion of text is a noisier action since the user may be moving the mouse and marking text as a reading aid.

6.3.2 Combining Feature Sets

As presented in Section 5, there are six types of feature sets: **(section) content**, **document**, **dwell time**, **highlight**, **copy**, and **clicks**. We start with the section Lucene score from the **content** features and incrementally add interesting feature sets. We divide the **click** feature set into further subsets, **outline**, **rollover**, **seelink**, and **all**. Abbreviations for feature set names are shown in Table 5.

The results are presented in Table 6. While training, we specifically optimize for NDCG. Statistical significance tests are done with the paired t-test (p-value < 0.05) comparing other feature sets to **c+doc+dw+cop+h+sl**, which uses (section) content, document and user search behavior features without clicks depending on an outline (seelink clicks are within documents pointing from one section to another). Looking at the results we see that without any machine learning the Lucene retrieved results perform poorest (sig-

Table 4: Some single features ranked in increasing order of MAP.

Feature	MAP	NDCG
luceneScore	0.5941	0.7610
cumuDwellDocProb	0.6426	0.7729
relViewDocProb	0.6528	0.7771
titleOverlap	0.6723	0.8005
seelinkClickDocProb	0.6868	0.8006
level	0.6900	0.7822
seelinkClickProb	0.6971	0.8113
referenceProb	0.7089	0.8129
hiliteDocProb	0.7161	0.8014
position	0.7172	0.8335
allClickDocProb	0.7188	0.7964
avgDwell	0.7285	0.8045
copyDocProb	0.7330	0.8135
outlineClickDocProb	0.7545	0.8117
rolloverClickDocProb	0.7632	0.8146
docLucene	0.8111	0.8396
docClickProb	0.8180	0.8595
cumuDwellProb	0.8202	0.8641
copyProb	0.8230	0.8831
relViewProb	0.8272	0.8677
hiliteProb	0.8382	0.8846
rolloverClickProb	0.8729	0.8940
allClickProb	0.8902	0.9070
outlineClickProb	0.8928	0.9096

Table 5: Feature Set Abbreviations that are used for the experiments.

Abbreviation	Feature Set
c	**content**
doc	**document**
dw	**dwell**
cop	**copy**
h	**highlight**
sl	**seelink**
o	**outline**
r	**rollover**

nificantly worse) with a MAP of 0.5941 and an NDCG of 0.761. We get a huge 24% absolute gain for MAP using section content features **c** with RandomForests while the gain for NDCG is 11%. Adding document features (**c+doc**) yields another major boost for both models. Next, we add user search behavior features. We found that when including dwell time, highlight or copy individually to **c+doc**, the differences are very small and insignificant. Therefore the next combinations we analyze are **c+doc+dw+h** and **c+doc+dw+cop**. While both yield similar gains across the two models, **c+doc+dw+cop** performs a little better according to the MAP results. There is no significant difference between the two and **c+doc+dw+cop+h+sl** according to RandomForests but with RankBoost **c+doc+dw+h** is significantly worse according to both measures – MAP and NDCG. Next, we combine dwell, highlight, and copy with the content and document features to yield **c+doc+dw+cop+h**. The only feature set that separates it from **c+doc+dw+cop+h+sl** are the seelink features. According to RandomForests, this difference is insignificant, but with RankBoost we get an absolute improvement of about 1%. Finally, we compare this result to using clicks from the outline interface (**c+doc+o**). Outline and rollover clicks are among the strongest single features according to Table 4 and it is not surprising that we get a significant gain by using them. But when comparing this to our purely user search behavior features combination **c+doc+dw+cop+h+sl**, the difference in the gain of about 1% is very small. This shows that when an outline interface is not available, we can achieve a comparable performance by using only user search behavior related features. Using all features combined we get another small boost for RandomForests while this slightly hurts NDCG for RankBoost.

Table 6: Results with combined feature sets. Results marked with ↑ (better) and ↓ (worse) are statistically significant over c+doc+dw+cop+h+sl within the same model using the two-paired t-test (p-value < 0.05).

Model	Feature Set	MAP	NDCG
Lucene	–	0.5941↓	0.7610↓
RandomForests	**c**	0.8317↓	0.8753↓
RandomForests	**c+doc**	0.9054↓	0.9282
RandomForests	**c+doc+dw+h**	0.9196	0.9397
RandomForests	**c+doc+dw+cop**	0.9214	0.9352
RandomForests	**c+doc+dw+cop+h**	0.9189	0.9389
RandomForests	**c+doc+dw+cop+h+sl**	0.9196	0.9389
RandomForests	**c+doc+o**	0.9306↑	0.9521↑
RandomForests	**c+doc+o+r**	0.9329↑	0.9501↑
RandomForests	**all**	0.9331↑	0.9509↑
RankBoost	**c**	0.7176↓	0.8265↓
RankBoost	**c+doc**	0.8642	0.8917
RankBoost	**c+doc+dw+h**	0.8661↓	0.8954↓
RankBoost	**c+doc+dw+cop**	0.8683	0.8958↓
RankBoost	**c+doc+dw+cop+h**	0.8684↓	0.8988
RankBoost	**c+doc+dw+cop+h+sl**	0.8701	0.9001
RankBoost	**c+doc+o**	0.8808↑	0.9105↑
RankBoost	**c+doc+o+r**	0.8849↑	0.9084
RankBoost	**all**	0.8855↑	0.9077↑

7. CONCLUSIONS

In this paper we explored the problem of improving section ranking with four types of section level user behaviors: section dwell time, section-level highlighting and copying of text, and section level clicks (including see link, outline and rollover click actions)

In our experiments exploring section ranking, we demonstrated that if section-level clicks are available, they provide a 3% gain in effectiveness over a highly tuned baseline of Lucene scores blended with document-level click information. However, recognizing that outline-style lists of section headings may not always be appropriate, we showed that non-intrusive logging of dwell time, text highlighting and copying, and "see also" links can provide around half of that gain. Of course, we note that these gains are not large, which also support the conclusion that section-level logging is currently of little value: using document-level clicks and excellent passage retrieval is almost as good as the approaches we explored for incorporating section-level logging information into ranking.

Our analysis revealed several interesting findings about user behaviors at the section level. According to our data

set, 50% of section dwells are shorter than 2 seconds, suggesting users skim many sections instead of reading them. We discovered an adjacent bias for section dwell times: non-relevant sections adjacent to relevant sections tend to get more dwell time than non-relevant sections that are further away, because when users read a relevant section, adjacent sections are also being displayed. For highlighting and copying, we find strong correlations between highlight frequencies and copying frequencies with a Pearson's correlation coefficient of 0.79. For clicks, we find that users tend not to click sections in the top part of the pages from the outline because they are already being displayed, which results in a position bias for section clicks.

8. ACKNOWLEDGMENTS

This work was supported in part by the Center for Intelligent Information Retrieval, in part by NSF grant #IIS-0910884, and in part by UpToDate. We are of course grateful to UpToDate for providing us with anonymized data for these experiments, but are particularly grateful to Jerry Greene's help in the time-consuming process of tagging data for our experiments. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the sponsor.

9. REFERENCES

[1] E. Agichtein, E. Brill, and S. Dumais. Improving web search ranking by incorporating user behavior information. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '06, pages 19–26, New York, NY, USA, 2006. ACM.

[2] E. Agichtein, E. Brill, S. Dumais, and R. Ragno. Learning user interaction models for predicting web search result preferences. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '06, pages 3–10, New York, NY, USA, 2006. ACM.

[3] E. Agichtein, E. Brill, S. Dumais, and R. Ragno. Learning user interaction models for predicting web search result preferences. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '06, pages 3–10, New York, NY, USA, 2006. ACM.

[4] E. Aktolga, J. Allan, and D. A. Smith. Passage reranking for question answering using syntactic structures and answer types. In *Proceedings of the 33rd European conference on Advances in information retrieval*, ECIR'11, pages 617–628, Berlin, Heidelberg, 2011. Springer-Verlag.

[5] P. Arvola, S. Geva, J. Kamps, R. Schenkel, A. Trotman, and J. Vainio. Overview of the inex 2010 ad hoc track. In *Proceedings of the 9th international conference on Initiative for the evaluation of XML retrieval: comparative evaluation of focused retrieval*, INEX'10, pages 1–32, Berlin, Heidelberg, 2011. Springer-Verlag.

[6] M. Bendersky and O. Kurland. Re-ranking search results using document-passage graphs. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '08, pages 853–854, New York, NY, USA, 2008. ACM.

[7] M. W. Bilotti, J. Elsas, J. Carbonell, and E. Nyberg. Rank learning for factoid question answering with linguistic and semantic constraints. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, CIKM '10, pages 459–468, New York, NY, USA, 2010. ACM.

[8] L. Breiman. Random Forests. *Machine Learning*, 45:5–32, 2001.

[9] S. Brin and L. Page. The Anatomy of a Large-scale Hypertextual Web Search Engine. *Comput. Netw. ISDN Syst.*, 30:107–117, April 1998.

[10] G. Buscher, A. Dengel, and L. van Elst. Eye movements as implicit relevance feedback. In *CHI '08 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '08, pages 2991–2996, New York, NY, USA, 2008. ACM.

[11] G. Buscher, A. Dengel, and L. van Elst. Query expansion using gaze-based feedback on the subdocument level. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '08, pages 387–394, New York, NY, USA, 2008. ACM.

[12] G. Buscher, A. Dengel, L. van Elst, and F. Mittag. Generating and using gaze-based document annotations. In *CHI '08 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '08, pages 3045–3050, New York, NY, USA, 2008. ACM.

[13] G. Buscher, L. van Elst, and A. Dengel. Segment-level display time as implicit feedback: a comparison to eye tracking. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '09, pages 67–74, New York, NY, USA, 2009. ACM.

[14] Y. Cao, J. Xu, T.-Y. Liu, H. Li, Y. Huang, and H.-W. Hon. Adapting ranking svm to document retrieval. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '06, pages 186–193, New York, NY, USA, 2006. ACM.

[15] Z. Cao, T. Qin, T.-Y. Liu, M.-F. Tsai, and H. Li. Learning to rank: from pairwise approach to listwise approach. In *Proceedings of the 24th international conference on Machine learning*, ICML '07, pages 129–136, New York, NY, USA, 2007. ACM.

[16] M. Claypool, P. Le, M. Wased, and D. Brown. Implicit interest indicators. In *Proceedings of the 6th international conference on Intelligent user interfaces*, IUI '01, pages 33–40, New York, NY, USA, 2001. ACM.

[17] comScore. comScore Releases April 2013 U.S. Search Engine Rankings. `http://www.comscore.com`.

[18] A. Corrada-Emmanuel, W. B. Croft, and V. Murdock. Answer Passage Retrieval for Question Answering. Technical Report IR-283, University of Massachusetts, Center for Intelligent Information Retrieval, 2003.

[19] V. Dang. RankLib. 2011. `http://www.cs.umass.edu/~vdang/ranklib.html`.

[20] S. Fox, K. Karnawat, M. Mydland, S. Dumais, and T. White. Evaluating implicit measures to improve web search. *ACM Trans. Inf. Syst.*, 23(2):147–168, Apr. 2005.

[21] Y. Freund, R. Iyer, R. E. Schapire, and Y. Singer. An Efficient Boosting Algorithm for Combining Preferences. *J. Mach. Learn. Res.*, 4:933–969, December 2003.

[22] G. Golovchinsky, M. N. Price, and B. N. Schilit. From reading to retrieval: freeform ink annotations as queries. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '99, pages 19–25, New York, NY, USA, 1999. ACM.

[23] Q. Guo and E. Agichtein. Beyond dwell time: estimating document relevance from cursor movements and other post-click searcher behavior. In *Proceedings of the 21st international conference on World Wide Web*, WWW '12, pages 569–578, New York, NY, USA, 2012. ACM.

[24] R. Herbrich, T. Graepel, and K. Obermayer. Large Margin Rank Boundaries for Ordinal Regression. In P. J. Bartlett, B. Schölkopf, D. Schuurmans, and A. J. Smola, editors, *Advances in Large Margin Classifiers*, pages 115–132. MIT Press, 2000.

[25] K. Järvelin and J. Kekäläinen. Cumulated Gain-based Evaluation of IR Techniques. *ACM Trans. Inf. Syst.*, 20:422–446, October 2002.

[26] M. Kaszkiel and J. Zobel. Passage retrieval revisited. In *Proceedings of the 20th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '97, pages 178–185, New York, NY, USA, 1997. ACM.

[27] M. Kaszkiel, J. Zobel, and R. Sacks-Davis. Efficient passage ranking for document databases. *ACM Trans. Inf. Syst.*, 17(4):406–439, Oct. 1999.

[28] D. Kelly and N. J. Belkin. Reading time, scrolling and interaction: exploring implicit sources of user preferences for relevance feedback. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '01, pages 408–409, New York, NY, USA, 2001. ACM.

[29] D. Kelly and N. J. Belkin. Display time as implicit feedback: understanding task effects. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '04, pages 377–384, New York, NY, USA, 2004. ACM.

[30] D. Kelly and N. J. Belkin. Display time as implicit feedback: understanding task effects. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '04, pages 377–384, New York, NY, USA, 2004. ACM.

[31] E. Krikon, O. Kurland, and M. Bendersky. Utilizing inter-passage and inter-document similarities for reranking search results. *ACM Trans. Inf. Syst.*, 29(1):3:1–3:28, Dec. 2010.

[32] Y. Liu, B. Gao, T.-Y. Liu, Y. Zhang, Z. Ma, S. He, and H. Li. Browserank: letting web users vote for page importance. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '08, pages 451–458, 2008.

[33] D. Metzler and W. Bruce Croft. Linear Feature-based Models for Information Retrieval. *Inf. Retr.*, 10:257–274, June 2007.

[34] B. N. Miller, J. T. Riedl, and J. A. Konstan. Grouplens for usenet: Experiences in applying collaborative filtering to a social information system. In C. Lueg and D. Fisher, editors, *From Usenet to CoWebs - Interacting with Social Information Spaces*, pages 206–231. Springer, London, 2003. Computer-Supported Cooperative Work.

[35] M. Morita and Y. Shinoda. Information filtering based on user behavior analysis and best match text retrieval. In *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '94, pages 272–281, New York, NY, USA, 1994. Springer-Verlag New York, Inc.

[36] P. Ogilvie and J. Callan. Language Models and Structured Document Retrieval. In *Proc. 1st INEX workshop*, pages 33–40, 2003.

[37] S. Tellex, B. Katz, J. Lin, A. Fernandes, and G. Marton. Quantitative evaluation of passage retrieval algorithms for question answering. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, SIGIR '03, pages 41–47, New York, NY, USA, 2003. ACM.

[38] C. Wade and J. Allan. Passage Retrieval and Evaluation. Technical report, University of Massachusetts, 2005.

[39] R. W. White and D. Kelly. A study on the effects of personalization and task information on implicit feedback performance. In *Proceedings of the 15th ACM international conference on Information and knowledge management*, CIKM '06, pages 297–306, New York, NY, USA, 2006. ACM.

[40] R. Wilkinson. Effective retrieval of structured documents. In *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '94, pages 311–317, New York, NY, USA, 1994. Springer-Verlag New York, Inc.

[41] J. Xu and W. B. Croft. Improving the Effectiveness of Informational Retrieval with Local Context Analysis. *Transactions on Information Systems*, 18(1):79–112, 1998.

[42] J. Xu and H. Li. Adarank: a boosting algorithm for information retrieval. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '07, pages 391–398, New York, NY, USA, 2007. ACM.