Technical Report

Department of Computer Science
and Engineering
University of Minnesota
4-192 EECS Building
200 Union Street SE
Minneapolis, MN 55455-0159 USA

TR 03-009

Average Position of User Clicks as an Automated and Non-Intrusive Way of Evaluating Ranking Methods

B. Uygar Oztekin, George Karypis, and Vipin Kumar

February 05, 2003

Average Position of User Clicks as an Automated and Non-Intrusive Way of Evaluating Ranking Methods *

B. Uygar Oztekin
University of Minnesota,
Dept. of Computer Science,
Army HPC Research Center
oztekin@cs.umn.edu

George Karypis
University of Minnesota,
Dept. of Computer Science,
Army HPC Research Center
karypis@cs.umn.edu

Vipin Kumar
University of Minnesota,
Dept. of Computer Science,
Army HPC Research Center
kumar@cs.umn.edu

ABSTRACT

The need for an objective and automated way of evaluating the performance of different ranking/reranking methods is becoming increasingly important in the web search/meta search domain. There are various methods for ranking search results ranging from traditional information retrieval approaches to more recent methods based on link analysis and other quality measures that can be derived from the documents. There are also a number of strategies for combining different heuristics and answers from multiple experts. With all of these possibilities it is becoming increasingly difficult to find the best parameters, the best method, or the best mixture of methods that will maximize the quality for a particular query type or domain. This paper addresses the problem of automatically comparing the quality of the ordering of documents that are presented to the user as a sorted list according to believed relevance for a given topic or query. We introduce the average position of user clicks metric as an implicit, automated, and non-intrusive way of evaluating ranking methods. We also discuss under which situations and assumptions this metric can be used objectively by addressing various bias sources. Experiments performed in our meta search engine suggests that, this approach has the potential to sample a wide range of query types and users with greater statistical significance compared to methods that rely on explicit user judgements.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*search process, information retrieval, information filtering*

*This work was supported by the Army High Performance Computing Research Center (AHPCRC) contract number DAAD19-01-2-0014. The content of this work does not necessarily reflect the position or policy of the government and no official endorsement should be inferred. Access to computing facilities was provided by AHPCRC and the Minnesota Supercomputing Institute.

General Terms

Algorithms, performance, measurement

Keywords

Ranking evaluation metric, web search, implicit relevance feedback, average position of clicks, non-intrusive feedback, sampling

1. INTRODUCTION

With increasing availability of search engines, meta search engines, digital libraries, and information retrieval systems, there is a growing interest in objectively and automatically comparing the quality of the ordering of the items (documents) presented to the user. We propose an implicit relevance feedback approach to compare different ranking/reranking methods that uses the location of documents selected by users from a ranked list presented. This approach is applicable for evaluating the performance of different ranking methods in traditional search engines and information retrieval systems, as well as for comparing the quality of different merging and reranking methods used in meta search engines that combine and reorder results coming from different sources.

Various strategies have been proposed and used for evaluating the performance of different ranking methods. At one end of the spectrum, evaluation criteria is based on human experts' explicit relevance judgements. Although this approach can be used in evaluating the performance of different methods for the selected subset of queries and users, it is very difficult to have human experts to evaluate a sufficiently large sample that will span majority of queries and users, since the manual evaluation process for each sample is tedious and time consuming. At the other end of the spectrum, there are approaches for obtaining the relevance judgements by analyzing the documents using automated methods (e.g. using similarity to a query or an expanded query). Such an approach allows automatic evaluation of ranking methods, but it is not evident that these methods can simulate human judgements with reasonable accuracy. In our previous work [8], we introduced average position of clicks as an automated way of comparing rankings produced by different merging strategies. This

metric combines the strengths of both ends of the spectrum. Users are still in the loop, but we look at their implicit relevance judgements seen through the user logs. This method also allows evaluation of rankings for a large number of samples, since all users and all queries may contribute to the evaluation. Unlike automated methods that replace human judgements, this approach has the potential for providing better accuracy. Furthermore, it offers a non-intrusive way of obtaining implicit user judgements. For a query, we only need to have information about which ranking method is used and the location of the documents that the user has clicked on. We neither need to identify users, nor to have the contents of the query and the documents selected.

This paper formally introduces the average position of clicks metric, discusses its strengths and shortcomings, describes assumptions on user behavior for applicability of the metric, and provides guidelines for researchers interested in using the metric in web search or other domains. The proposed metric is also compared to average uninterpolated precision, which is widely used in the presence of explicit user judgements. Key differences between the two metrics are pointed out and insight on why the proposed metric, perhaps in conjunction with other statistics, may be preferred in the presence of implicit relevance feedback is provided. We make an effort to identify various bias sources that may affect the proposed metric, suggest extensions and modifications that could be used in related domains, and give insight on learning and exploration potentials that may be enabled by automated and implicit ways of obtaining relevance judgements. As an example application, we provide a summary of the results of our previous study indicating that the approach is promising in meta search domain, and that it can potentially be extended to other domains. Note that this application inherently does not address a portion of the issues that will be discussed. Some of the bias sources does not appear in a meta search context, and the click characteristics may not be representative of other domains. Comprehensive test of bias sources and learning potentials for various domains requires a series of further studies, and is beyond the scope of this paper.

The remainder of the paper is organized as follows: Section 2 presents the related work including applications that may benefit from the metric, Section 3 describes the average position of clicks metric including assumptions about the user behavior and discussions about various bias sources, Section 4 introduces an example application that the metric was applied to, Section 5 discusses learning and data exploration potentials, and finally, Section 6 presents our conclusions.

2. RELATED WORK

Explicit as well as implicit relevance feedback has been used in various applications in information retrieval, web search, and meta search to evaluate the performance of different ranking, reranking, and fusion methods, as well as learning the best parameters in various applications. If explicit user feedback is used, various methods are evaluated according to the relevance judgements given by a number of human experts for a subset of selected cases such as a selection of queries. This approach has the benefit of giving fairly accurate results on the performance of the methods for the particular set of cases selected. However, due to practical reasons, only a small fraction of the possible cases or queries as well as users can be sampled. Evaluation methods that use implicit relevance information have been proposed as an alternative in the lack of explicit judgements. As an example in information retrieval domain, one such method uses automated ways to simulate user judgements, typically using measures such as cosine similarity between the query and the documents, and term frequencies and/or phrase frequencies of the query terms present in the text [5]. Even though this approach has the potential to sample a wider range of queries, since human judgements are no longer in the loop, relevance judgements highly depend on the particular method used in approximating the relevance information, which may introduce considerable bias to the evaluation process.

Somewhere in between the above examples, there are implicit relevance feedback approaches that examine user behavior such as time spent on a particular document and number of clicks a user makes for a particular query, to come up with implicit ways of judging the performance of different methods. For instance, Kim et al. [4] conducted a study using a number of undergraduate students, focusing on USENET articles as well as journal articles, and used time spent reading the documents and printing behavior as an implicit way of obtaining relevance judgements.

Schapira [10] used a reinforcement learning approach for reranking and filtering search results. For each query, the system recorded how many times a particular URL was clicked on by different users. When the same query is issued at a later time, past information was used to boost the scores of the frequently visited URLs while lowering the scores of unvisited ones. A similar approach was also used by a commercial search engine, Directhit [1]. Implicit relevance feedback has also been used in profile based systems to learn user interests in time, and to rerank the search results to reflect them (e.g. by changing the relative importance of the terms according to user profile learned by the system) [16].

Profusion [12], one of the early meta search engines, used an explicit relevance feedback approach to learn the best weights in combining the results coming from different search engines. It used a variation of linear combination of scores approach [15], also used in various documented meta search engines ([11, 7], [12, 9], [2, 3]). In linear combination of scores model, the relevance of a document to a query is computed by combining both a score that captures the quality of each source and a relevance score of each document that is provided by the source, capturing the quality of the document with respect to the query. After the scores of the documents are normalized across the sources, each document's final score is calculated by multiplying its normalized score in the original source by the overall score or weight of

the particular source it is coming from. Profusion's approach in learning the scores of each source (search engine) was to train the system using a user study based on a relatively small subset of selected queries.

SavvySearch [2, 3] focused on learning and identifying the right set of search engines to which the queries should be issued depending on the topic of the query. Its aim was to figure out the best set of search engines that will respond with the best set of documents given the query terms. The system learned scores associated with each search engine by looking at their recent performance for each stemmed query term present in the query under investigation. For each term, a score was associated with each search engine. Criteria used in evaluating the search engines included number of links selected by the users from a particular search engine and whether or not the search engine responded with no results for the query term of interest. Although they looked at how many documents were selected per query, they did not focus on the positions of these documents. SavvySearch used a relatively simple implicit relevance feedback approach to learn the best set of engines for each stemmed query term.

In text retrieval domain, once the relevant and non-relevant documents are identified, provided that we have indication about how many relevant and non-relevant items are present in the whole collection, results produced by different methods are often compared using standard precision and recall analysis if the order of the documents is not important. If the order of documents is important, and if we have extensive information about relevant as well as non-relevant documents, modified methods that also take ranks into account such as average uninterpolated precision, which is widely used in TREC [14] benchmarks, can be considered. When we have partial information about relevant documents, and if we are interested more in the distribution of the user selections and roughly where they are occurring using different methods, we will discuss that the metric we are proposing, the average location of user selections or clicks can be more suitable. In our previous study [8], we considered average uninterpolated precision and average position of user clicks as possible metrics to evaluate the performance of different merging and reranking approaches implemented in our meta search engine, Mearf [6]. We experienced that for this particular domain, considering the pros and cons of each metric, average position of clicks seemed to be the beter choice, and that, the differences between methods under observation were more visible using this metric.

3. AVERAGE POSITION OF CLICKS

If extensive relevance information is available i.e., if we have indication about all the relevant and non-relevant documents as well as statistics for the whole collection, traditional information retrieval metrics such as precision and recall can be reliably applied to compare the quality of different approaches. If rank information is also important, then related measures such as average uninterpolated precision are proposed and widely used. These approaches work well with explicit user judgements in which a selected set of users extensively identify relevant as well as non-relevant items typically in the whole collection, but in the absence of explicit user judgements, application of these metrics becomes increasingly problematic.

In domains such as web search, implicit relevance judgement information obtained by observing the user behavior can probably suggest that some documents were found interesting, which may be considered as relevant documents, but in most of the cases we do not have reliable information on most of the items that are presented. Statistics on the whole collection such as how many documents are relevant and non-relevant to the query may also be unavailable. Assuming that we have implicit indication about a subset of relevant documents, and assuming that the average user typically starts to examine the results from the beginning most of the time, we can look at the locations where the user has found interesting documents using different methods. Average locations of these documents may be used for comparing the relative performance of the methods under consideration. Note that unlike explicit user judgements that are carried out typically for a large number of items returned, a typical user does not examine each link one by one to investigate all interesting documents. The user may obtain the information of interest by examining a few items, and completely skip other possibly relevant items without any feedback. Moreover, unlike explicit relevance judgements in which negative feedback is also provided, using implicit feedback approaches, there may not be a clear distinction between negative feedback (bad items) and no feedback at all (user did not investigate a given portion). Under these scenarios with partial relevance information, provided that all other parameters are fixed, we found that the average position of user clicks as a metric is easier to interpret and more intuitive in evaluating the performance of different approaches.

In the case of a search engine or a meta search engine, if we have different methods of reordering the results, we can evaluate their performance by looking at around which positions on the average, the users are able to find relevant or interesting information with each of these methods. If a particular method places high quality links in significantly better positions (towards the beginning of the list) compared to other methods, then we argue about this particular method being superior to others for the queries that we examined.

Average position of clicks or selections for a single query is given by $(1/k) \times \sum_{i=1}^{k} p_i$, where k is the total number of selections, and p_i is the position of the i^{th} selected item. For example, assume that we have two methods for reranking the same set of documents in different ways, and the task is to determine which of these methods are better regarding the quality of the ordering they produce (i.e., relevance of items are highest in earlier positions and drops down as we go down in the list). Assume that the user is looking for two particular items or documents. If the user can find interesting items at the 5^{th} and the 7^{th} positions with the first

method, and at the 9^{th} and the 15^{th} positions with the second method, using the average position of clicks metric for this particular example, in the first case, the user is able to find interesting items at the 6^{th} position on the average using the first method vs. the 12^{th} position on the average using the second method. We can argue that the first method is superior to the second method. If we look at a larger number of samples and look at the average position of clicks on all of the queries using two different methods, the method having the smaller average is intuitively better than the other method, provided that all other parameters are the same.

Note that there are two natural ways of obtaining the average position of clicks for multiple queries. One may take the average position of clicks for each query and take the mean of averages for each query that used a particular method. The other alternative is to treat each click individually and take the average of the positions of all collected clicks for a particular method. Although both of these approaches is expected to converge to the same mean, standard deviation for the first case is expected to be smaller than the second case.

In order to check if we have enough statistical significance, we propose to use the number of samples and the standard deviation to check whether or not we have enough samples by using confidence tests such as chi square. If average position of clicks is taken for each query, and then averaged for all queries that used a particular method, we propose to use the number of queries as the number of samples. If clicks are treated individually, and the mean and standard deviation is obtained accordingly, number of clicks should be used as the total number of samples. Both of the choices seem reasonable as long as the correct number of samples is used with the associated standard deviation.

In the following subsections we will discuss our assumptions about the user behavior, address possible bias sources and possible solutions, and compare average uninterpolated precision against average position of clicks, highlighting some of the key differences.

3.1 Assumptions on user behavior

The proposed metric is applicable in cases where users are given an ordered list from which they can select a number of interesting items or documents. This could be a search engine's response to an issued query, results of a meta search engine, or results of a query issued to an information retrieval system. In all of these cases, we assume that the results are presented to the user as an ordered list, sorted according to the relevance judgements of the particular engine and the method used. Our metric examines the positions of the documents that the users have selected using different methods, and compares methods against each other by looking at how good they are in placing interesting links in top positions on the average.

We also have a number of assumptions about the average user of the system. We assume that the average user is intelligent in making selections. For instance in the web search domain, if a user issues a query and gets a set of results back as an ordered list, sorted according to the relevance belief of the engine, we assume that the user is able to judge the relevance of the documents by looking at the summaries of the documents (snippets, titles, URLs). We also assume that the user starts scanning the list from the beginning, skip uninteresting links, and click on the ones that seems interesting or relevant to the query. We do not necessarily assume that the user scans the list exclusively linearly. For instance, the user may scan the list in blocks, look for interesting keywords or query terms (highlighted by most search engines), investigate a selection of links in a particular block, then move its window to lower positions to include documents lower in ranks and ignore documents that he/she already investigated in earlier positions.

Under the above assumptions, and assuming that other parameters are equivalent for different methods under comparison, we can argue that a method that has a lower average position of clicks being a better method compared to the ones having a higher average.

3.2 Bias sources

Average position of clicks metric can be biased in various ways. In the following subsections we address major bias sources and propose solutions or partial solutions for each.

3.2.1 *Average total number of links returned*

The most obvious bias source in using the proposed metric is probably the total number of links that are presented to the user and how easy for the user to navigate in the set of links that are returned. Provided that the user interface is the same for two methods in terms of look and feel as well as in terms of interactivity (response time, network delay etc.), total number of links presented on the average for different methods is a biasing factor in using average position of clicks metric. For instance, if we are considering two methods, the first one retrieving on the average 50 links per query, and the second one retrieving on the average 100 links per query. We can expect the average position of clicks to be lower for the first method simply because the users are not able to go past the 50^{th} position, whereas they can go up to 100 in the case of the second method. If other parameters are fixed, number of links retrieved on the average for different methods should roughly be the same in order to be able to have reliable comparisons, except in the case in which the method that presents the larger set of results has also the smaller average.

If total number of links presented by the methods under comparison are different, we suggest dividing the data into bins according to number of links retrieved. This should produce comparable number of links presented for each method under each bin, making it possible to alleviate the bias and to compare the methods better under these subdivisions.

3.2.2 *Different sets of results returned*

In comparing two reranking methods for the same query i.e., same set of results presented, we can directly look at the average position of clicks, since on the average, comparable number of links would be returned and

these links would be the same set of links but ordered differently according to the reranking method used.

We believe that if the different methods present different sets of results, average position of clicks metric should be more carefully applied. We can think of various scenarios: Assume that one of the methods presents relatively poor results compared to another method, but the distribution of relative relevance of the documents are similar in both methods. The user may still select roughly the equivalent positions, and select the same amount of links to be investigated, but the quality of the documents presented by the two methods can be quite different. In another scenario, assume that one of the methods has only one relevant document in top ranks, for instance at position 2, and the user clicks on that document. Assume that there are no other relevant documents in the set. Let us now consider another method which has 4 relevant documents in top 20 positions, for instance, at positions 1, 3, 6, and 9, and the user happens to select the first three of them. According to the average position of clicks metric, the first method is a better method if we do not consider the difference in total number of clicks, but one may argue that the second method is better than the first one in terms of number of relevant links that are presented.

To partially address the above issues, we suggest to look at the ratio of queries with clicks vs. the total number of queries. For instance, if a particular method performs much worse than other methods i.e., the quality of the documents presented using that method is significantly worse than others, one might expect that this method will also have a higher percentage of queries for which the user simply does not like the results and abandons without clicking on any of the documents. Significant difference in the ratio of queries with clicks vs. the total number of queries can be an indication for that type of situations. One may also look at the average number of clicks per query for different methods. If there is a significant difference in the average number of clicks per query among different methods, this might give an indication that the overall quality of the results returned by the different methods under investigation varies from method to method.

Another possibility is to divide the data according to number of clicks per query and compare the methods under each bin. The difference in this, combined with the above statistics, can be an important clue in situations in which there is significant difference in the quality of the documents produced by different methods.

3.2.3 Focused crawlers, robots, meta search engines

Automated agents that query the system and follow links can behave differently compared to our assumptions about the human users. If one is not careful in identifying them, they may introduce bias into the measurements. Possible bias sources we can think of are focused crawlers, robots, spiders, and meta search engines. Meta search engines may use the search engine we are focusing on, introduce results from other engines, and present them to the users with different orderings. Focused crawlers may start from a given query and download, for instance, all first few hundreds of results. Clustering and indexing interfaces as well as other possible interfaces that reorder the results or divide them in different views may also introduce significant bias if they are used on top of the search engine without being noticed. It is also possible that a competitor allocates resources to automatically issue bogus queries and arbitrarily simulate clicks on documents. It may be possible to reduce that kind of spamming by identifying and ignoring the hosts that issue more than a predetermined number of queries per day or identifying the cases in which the requests come much faster than a human surfer can possibly handle. It may also be possible to do a variant of log analysis to automatically identify robots (e.g. approaches suggested by Tan et al. [13] can possibly be modified and applied to this domain).

3.2.4 Changes in user behavior

Current assumptions about the behavior of users seems reasonable for the typical user we analyzed in logs as well as selected users we examined in real life. However, changes in user practices, for instance due to changes in user interfaces, browsers, and other factors, may affect the overall behavior of the average user. Before relying on this metric, one must make sure no bias is introduced into the system and the user model one has in mind still applies to a majority of the users to make sure that the statistics are not significantly affected by minorities or marginal users. If the results still come in ordered format, but different views and navigation facilities are supplied along with it, such as indices for related keywords or other filtering and navigation approaches, the validity of the assumptions about the user behavior should be reevaluated.

3.3 What are we measuring?

It is possible to use the average position of user clicks or selections metric in different domains. Before applying the metric, one must think about the meaning of clicks or selections for the particular domain of interest. If there are different meanings associated with the clicks or selections other than the fact that the user finds them interesting, then the results may be biased for a subset of methods against other methods.

In the web search domain, we believe that the user model we discussed is reasonable for the average human user. However, if we base our study on the user clicks, and if the users only have the snippets and titles to judge the quality of the document, we do not directly measure the relevance of the documents. It is quite possible that a document may have an interesting snippet, but the actual document can be less relevant to the query. The reverse situation is also possible. Additionally, there are all sorts of possibilities such as the link selected referring to a document which is no longer online. However, if we consider a large number of samples, and if one method is not favored or disfavored against others, one might expect that the fluctuations will be evened out.

By using the average position of user clicks metric in

the web search domain we basically measure how good a method is in placing interesting snippets and titles in higher positions in the eyes of the users. If the quality of the summaries i.e., snippets and titles, are highly correlated to the quality of the actual documents, then we are indirectly measuring the quality of the actual documents too. If the correlation for a particular domain is not sufficiently high, one must use judgements about the actual documents. One possibility is to infer this information by looking at how much time the user has spent examining each document and other statistics, but all of the new assumptions may introduce additional interpretations and possible bias.

3.4 Average position of clicks vs. average uninterpolated precision

Once a set of interesting or relevant documents are identified, an evaluation metric should be selected depending on the domain of interest. Different metrics have different objective functions, strengths, and sensitivities to different types of bias sources.

If we have k documents selected, average position of clicks or average position of selections is given by

$$\frac{\sum_{i=1}^{k} p_i}{k} \tag{1}$$

where p_i is the position of the i^{th} clicked or selected document.

For a given ordered list, if k documents are deemed relevant, the average uninterpolated precision is given by

$$\frac{\sum_{i=1}^{k} i/p_i}{k} \tag{2}$$

in which, p_i is the position of the i^{th} relevant document sorted according to positions i.e., according to their order of appearance in the ordered list, p_1 being the position of the first relevant document from the top.

The second metric produces a measure of precision from 0 to 1, 1 being the best case, happening only if all relevant documents are selected one after another from the beginning of the list and no non-relevant documents are present in between the relevant ones. Due to inverse of positions term on the numerator, this metric puts the highest importance to the first positions, and contributions of lower ranks decreases asymptotically to zero as we go down to lower ranks. In other words, if the number of relevant documents are fixed, and if we slightly move the relevant documents up and down by a few ranks, changing the position of the documents in first ranks makes the highest changes in the measurement. A change of one rank in these positions, for instance, from 3^{rd} position to 4^{th} position may have more effect than doubling or tripling the position of a relevant document in lower ranks, for instance, around 15^{th} position. This behavior may be desirable or undesirable depending on the domain. Note that, using the average position of clicks metric, the contribution of each position is the same; a change in one rank in the position of a relevant document in earlier positions is the same as a change in one rank in later positions.

As an exercise, we compared these two methods by simulating all possible cases for a collection of 20 and 50 documents and two simulated users. We fixed the number of clicks that the users would select out of the collection, applied both metrics, and noted the cases in which the two methods contradict each other i.e., one suggests that one set of clicks are better than the other, while the other metric suggests the reverse. We then sorted these cases according to the highest difference in average position of clicks and the highest difference in average uninterpolated precision. As an example with two simulated clicks, if the first set of clicks are in positions 3 and 8, and the second set of clicks are in positions 2 and 18, the average position of clicks are 5.5 and 10 respectively, clearly favoring the first set. However, in terms of average uninterpolated precision (0.29 and 0.31 respectively) the second set is better than the first one. As another counter-intuitive example, again fixing the number of clicks to two, assume that the first method has an average uninterpolated precision around 0.5, for instance the user might have selected the 2^{nd} and the 4^{th} links. For the second method, if the user selects the first link, no matter what the position of the second link selected is, even if it is the very last link in the list, say 100^{th} or 1000^{th} link, the average uninterpolated precision is higher than 0.5.

We observed that, given two methods having the same average position of clicks, if the clicks mostly occur towards the top ranks, the average uninterpolated precision tends to favor the method having the higher standard deviation, since the contribution of items towards the top positions can easily overcome the penalties incurred by higher positions and the scaling factor of i in the numerator. On the other hand, if the clicks occur in relatively higher positions, far from top ranks, benefits of lower positions may not be enough to overcome the penalties incurred by higher positions. In this case, average uninterpolated precision tends to favor tighter distributions having the same average. We believe that the click distributions and ranges in most search applications, especially in web search and meta search where a typical user rarely investigates more than a few tens of links, falls heavily in the first case.

Another key difference between these two metrics is the sensitivity to number of samples or clicks in the query. Assume that relevant documents are uniformly distributed in a particular range for a given method. Further assume that we have two users viewing the results, the first user selects x relevant documents out of z total relevant documents, and the second user selects a smaller subset, y documents, where y is smaller than x (e.g. y is half of x). If the selection of the documents by each user is done in a random manner out of all relevant documents, using the average position of user selections metric, the two results are expected to be similar, both approximating the mean of the distribution. Whereas, average uninterpolated precision metric may produce significantly different results for the two users. The key difference between these two methods is

that the average uninterpolated precision effectively assumes that the documents that are not selected are non-relevant to the query, whereas average position of clicks metric does not directly penalize the documents that are not selected. This may be desirable if we have partial information about the relevance of the documents, which is typically the case if implicit relevance judgements is used. In some sense, average position of clicks metric is less sensitive to number of samples i.e., number of user clicks for a particular query. We must also point out that as discussed in Section 3.2.2, the average position of clicks metric may be unable to distinguish between methods that have the same distribution of relevant documents, but with different percentages. For instance, if two methods have the same mean and standard deviation of relevant documents, but one of them has more relevant documents than the other, or the overall quality of the relevant documents in one method is significantly better than the other, average position of clicks metric may not be able to suggest this difference unless we introduce other measures such as average number of clicks per query or ratio of queries with clicks vs. total number of queries. On the other hand, if average uninterpolated precision metric is used with implicit and incomplete information, the number of clicks becomes a bias factor. In this case, we suggest to divide the data set at least according to number of clicks.

As an illustration for sensitivity to number of samples per query, assume that the same set of links, for instance 20, are returned for both cases, and that 3 of them are relevant to the query. Assume that the first method places the relevant documents in positions 4, 5, and 7 and the second method places them at positions 8, 9, and 10. Finally, assume that the first user selects only the 5^{th} link from the first method, while the second user examines 8^{th}, 9^{th}, and 10^{th} links from the second method. The average position of clicks are 5 and 9, respectively, which makes the first method superior to the second using this metric. The uninterpolated average precision, in contrast, is 0.2 for the first method, and about 0.216 for the second method, favoring the latter. However, if all 3 relevant links were chosen by the first user, average uninterpolated precision too, would consider the first method significantly better than the second one. In evaluating the performance of different merging methods, we observed that the two metrics suggested slightly different results in various bins i.e., one method was better than another method using one metric, but the reverse was suggested by the other metric. However once we divided the data set according to number of clicks in the query, and examined the methods under each subdivision, the two metrics showed less contradictions.

No single metric is best for all cases and applications. In comparing reranking methods in which the documents produced by different methods are the same but reordered, and if we have implicit and incomplete relevance information, considering the pros and cons of the two metrics, we selected the average position of user clicks as the primary metric to be used in our evaluations. Note that the selection of metric is orthogonal to implicit way of evaluating the results, other metrics can also be used, some may also be suitable for this domain, but no matter what the selection is, all possible sources for bias should be examined and addressed, for instance by dividing the data set into different subsets.

4. EXAMPLE APPLICATION IN META SEARCH DOMAIN: MEARF

We used the average position of clicks metric to evaluate the performance of various meta search approaches in our meta search engine, Mearf, which was online at http://mearf.cs.umn.edu/, and advertised in our department's home page since Dec 2000. In time, Mearf gathered a relatively small but steady international user base that issue a wide range of queries. In our previous paper [8], we presented the results of the user logs from Dec 2000 to Dec 2001, including the statistics from all queries and all users who issued these queries. Our main aim was to compare different merging and reranking approaches that can be used in meta search engines. We introduced 4 new methods (Centroid, WCentroid, BestSim, and BestMSim), and compared them against 2 variations of existing methods that we also implemented (Interleave and Agreement).

When a new query is issued to Mearf, one of the methods is randomly selected and used in merging/reranking. The user has no control on the selection of the method, nor on any related parameters. In order not to introduce psychological bias, we did not let the users know which particular method is selected and we were careful about not to give any visible clues. During the majority of the logs used in the study, Mearf used 4 search engines. For a regular user, we asked for 20 documents to be retrieved per engine, and after removing the duplicates, we ended up with about 60 to 65 unique links for a general query. Unlike most search engines, Mearf does not report the results incrementally like 10 or 20 documents per page. All of the retrieved results are presented at once in a compact manner, fitting about 20 links/snippets in a typical browser page. The user can easily scroll up and down in the whole set of results and click on the links that he/she finds interesting.

Queries are identified via a unique session id incorporating current time and process number, and the positions of the clicked documents are recorded and associated with the query they correspond to. We chose not to record IP numbers and not to identify the user in any way due to privacy issues. For a given query we have the query text and associated statistics such as number of terms in the query, we know how many documents are presented, and we have the locations of the documents that the user have clicked on.

Table 1.a summarizes the overall characteristics of the data set obtained from the logs. Table 1.b shows the characteristics of the data for different fusion methods. The column labeled "avg results per query" is the average number of documents returned by Mearf for each query, the column labeled "number of queries" is the number of times a particular method was selected to rerank the results, the one labeled "number of clicks"

1.a High level statistics	
total number of queries	17055
number of queries with clicks	10855
number of clicks	34498
average clicks per query	2.02
avg clicks per query ignoring queries without clicks	3.18
click ratio (queries with clicks / total number of queries)	0.64
average retrieval time	1.99 sec
average processing time	0.29 sec
average total time per query	2.28 sec

1.b Statistics for each method				
method	avg results per query	number of queries	number of clicks	click ratio
Interleave	62.64	1530	3015	0.64
Agreement	62.09	655	1241	0.60
Centroid	61.74	3381	6702	0.64
WCentroid	61.70	2403	5018	0.65
BestSim	61.93	3443	6817	0.62
BestMSim	61.45	3220	6671	0.65
Other	48.25	2423	5034	0.64

Table 1: Overall characteristics of the dataset

shows the total number of documents that were clicked using the corresponding method, and the column labeled "click ratio" is the number of times a particular method is used which resulted in at least one user click, divided by total number of times the method is used in reranking.

Table 2 summarizes the overall performance of the six fusion methods implemented. The column labeled "AvgPos" shows the average position of the documents that the user deemed as relevant by clicking on them, the column labeled "StdevPos" shows the standard deviation of the positions of the relevant documents, the column labelled "AvgPrec" shows the mean average uninterpolated precision. We also added two columns, "AvgFirst", and "AvgLast" showing average position of first clicks and last clicks of the queries respectively to give an idea about the ranges of positions of clicks for each method. By looking at the overall results, average position of clicks metric was able to offer better distinction compared to average uninterpolated precision. In terms of average position of clicks, the best methods were Centroid and WCentroid, and then BestSim and BestMSim methods, while focusing on average uninterpolated precision, all of the methods except the Interleave method was pretty much comparable.

An interesting observation follows from these results when comparing the Agreement method against other methods. This particular method boosts the rank of the URLs that commonly occur in multiple search engines. It assigns a score inversely proportional to the position of the document in the source search engine, and while merging the lists, it sums these scores if the same URL is observed in multiple search engines. As a result, for general queries, top few positions are very likely to contain high quality documents that is common in top positions in different search engines. Although it is one of the best methods according to average uninterpolated precision (in fact in smaller time windows or subdivisions, it outperformed others by one or two percents), it has a quite high average position of clicks compared to other methods except the Interleave method. Note that it has the highest standard deviation of position of clicks among the methods examined, and it can be used as an example showing that, with this and similar click characteristics, average uninterpolated precision tends to favor the methods with higher standard deviation against the ones having similar, sometimes significantly better average position of clicks. In some sense, unlike other methods implemented, this method puts the highest emphasis on optimizing the quality of the documents in top positions.

We divided the dataset into different bins according to number of links returned on the average in order to be able to focus on different methods for queries that returned different number of links, and compared each method against others in these bins separately. A subset of these results are summarized in Table 3. We also divided the dataset according to number of user clicks in the queries in order to alleviate any bias that may have been introduced by number of selections per query. A summary of these results are shown in Table 4. In each entry, the first number is the average position of clicks, and the second number is the mean average uninterpolated precision. Note that this is one of the tables in which the two metrics tend to agree in majority of the cases. This can be perhaps associated with average uninterpolated precision metric's sensitivity to number of clicks per query. Once the dataset is divided according to number of clicks per query, this possible bias factor is removed.

We then divided the dataset according to the number of terms in the query which may give an indication about how general or specific the query is. If there are few terms in the query, one may argue that the query is general or broad, and as more and more terms are added, one may argue that the query is getting more and more focused or specific. We found out interesting patterns that we did not predict in the beginning. One of the methods, BestSim, was performing quite good compared to others with smaller number of terms in the query, and it was getting worse and worse as the number of terms in the query was increasing. Another method, BestMSim, showed the opposite trend. Moreover, for at least two bins (2 terms and 5+ terms queries) one of these two methods outperformed all other methods investigated with respect to average position of clicks metric while both of these methods was outperformed by Centroid and WCentroid in the overall case. A subset of these results is given in Table 5. In each entry, first number is the average position of clicks, and the second number is the mean average uninterpolated precision. We encourage the interested reader to refer to our previous paper for detailed description of the methods and detailed analysis of results [8].

method	AvgPos	StdevPos	AvgPrec	AvgFirst	AvgLast
Interleave	17.4	18.7	0.33	5.9	18.7
Agreement	17.4	19.7	0.39	5.1	18.9
Centroid	12.6	14.4	0.39	4.6	15.2
WCentroid	12.9	14.2	0.38	4.8	16.1
BestSim	13.6	14.9	0.39	5.2	16.4
BestMSim	13.5	15.2	0.39	4.8	15.6

Table 2: Overall performance of methods

	25-49 links		50-74 links		75+ links	
method	AvgPos	AvgPrec	AvgPos	AvgPrec	AvgPos	AvgPrec
Interleave	11.3	0.34	16.5	0.34	21.3	0.30
Agreement	12.1	0.34	15.6	0.42	21.6	0.34
Centroid	9.8	0.42	12.8	0.39	14.1	0.37
WCentroid	10.3	0.42	12.6	0.38	15.4	0.33
BestSim	11.0	0.34	13.4	0.40	15.6	0.37
BestMSim	10.2	0.38	13.4	0.39	15.3	0.35

Table 3: Varying number of links returned per query

In summary, our experience with Mearf showed that

- Average position of user clicks is indeed biased by total number of links retrieved. By dividing the dataset into bins according to total number of links returned, for all methods, average position of user clicks heavily depended on average total number of links returned in the bin. This trend is clearly visible in Table 3. Although the numbers in different bins were significantly different from each other, relative performance of methods and their ordering from best to worst across different bins remained fairly consistent.
- The metric was capable of differentiating between methods implemented in Mearf. Our study suggests that using the metric as the comparison criterion, there were significant and consistent differences between methods implemented. Under each subdivision of the dataset (according to number of links returned, according to number of terms in the query text, as well as according to number of user clicks per query) some methods had consistently better average position of clicks than others i.e. users were more likely to click the links occurring in earlier positions using some methods compared to others. This trend was also visible for subdivisions in different time windows of a few months (not shown).
- Different metrics tell the story from different angles. We compared average position of clicks and average uninterpolated precision metrics under each subdivision. Although they mostly agreed in general, methods favored by the two metrics in some subdivisions were quite different. This can be mainly attributed to how much emphasis is given to different ranks by each method, which is addressed in Section 3.4. In comparing the performance of different methods, selection of the right metric is crucial. It may also be beneficial to examine the results using multiple metrics, each emphasizing different aspects.

In our experiments, we addressed a few of the bias sources and their proposed solutions, others still remain to be verified thoroughly. Performance of the metric and applicability of implicit relevance judgements to various domains should also be tested against human experts' explicit judgements. Although we have promising results with a small number of users and their subjective judgements, due to practical limitations, it is very difficult to obtain a large enough and objective comparison set to offer statistical significance.

4.1 Extending the analysis to incremental results

In applying average position of user clicks metric to traditional search engines that return links in predetermined increments, we propose to take a slightly different approach. Since the number of links returned on the average can be a biasing factor, we suggest to use the number of documents in the increments the user has investigated as the total number of links returned. For instance if the search engine reports the results in increments of 20, and if the user has decided to look at the first and the next page in which both pages contained 20 links, it may be reasonable to assume that the total number of links presented to the user to be 40 and compare this with other methods in which the user has also investigated the first two full pages. If the number of links on the last page is less than the increment limit, it may be reasonable to assume that the total number of links retrieved is the total number of links shown up to now, but there might be psychological issues. For instance, if there are only a few links on the last page, the user may be tempted to investigate seemingly the most relevant link(s) in the last page even though if the same link(s) was presented in the previous page which contained more interesting links, it could have been easily skipped.

In order to limit such biases, we suggest to divide the data set into different subdivisions (e.g. queries in which the user investigated the first full set of results, queries in which the user have investigated the first two

method	1 click		2 clicks		3 clicks		4 clicks		5+ clicks	
Interleave	7.3	0.45	9.4	0.32	13.0	0.25	14.8	0.24	22.9	0.17
Agreement	6.6	0.54	8.3	0.41	15.1	0.27	12.9	0.28	23.3	0.18
Centroid	6.3	0.52	8.2	0.40	10.0	0.32	11.2	0.27	16.2	0.21
WCentroid	7.0	0.49	8.4	0.39	10.0	0.30	11.5	0.28	16.0	0.21
BestSim	7.3	0.52	8.4	0.41	11.5	0.30	12.8	0.27	16.7	0.21
BestMSim	6.5	0.50	8.9	0.38	9.9	0.32	12.4	0.28	17.3	0.21

Table 4: AvgPos/AvgPrec, slicing by number of user clicks per query

method	1 term		2 terms		3 terms		4 terms		5+ terms	
Centroid	12.6	0.41	13.2	0.39	12.5	0.39	12.3	0.38	12.1	0.40
WCentroid	11.4	0.41	13.7	0.35	13.5	0.37	11.9	0.40	12.3	0.37
BestSim	12.7	0.43	14.8	0.38	13.1	0.39	13.8	0.38	12.0	0.36
BestMSim	12.5	0.43	12.5	0.38	13.2	0.37	15.1	0.37	16.2	0.38

Table 5: AvgPos/AvgPrec, varying number of terms in the query

full pages, queries which returned k number of results, where k is less than the increment value used in the queries, etc.). If different methods are compared in each of these bins, it may be safer to assume that at least some portion of the possible bias is eliminated.

By investigating the position of clicks under various bins and under various types of queries, examining the statistics from different angles may show interesting trends and psychological effects. We have seen a few examples in Mearf's user logs. For a widely used general purpose search engine, we believe that there are various possible ways to look at the data and explore interesting trends and patterns.

5. LEARNING AND EXPLORATION POTENTIAL

An automated evaluation method to test the performance of different approaches under various scenarios can enable learning of optimal parameters for a ranking/reranking method. Parameter space controlling ranking or reranking strategies can be sampled and tested under various query types and domains. For instance, in source selection or collection fusion problems, weights associated with different search engines can be learned to maximize performance and can be periodically adjusted to changes in the search engines used. In the next subsections we will discuss a few of these areas that we think may benefit from using automated ways of obtaining and using relevance information. Note that these methods require large number of samples and are unpractical using explicit relevance judgements.

5.1 Learning the optimal parameters for a particular method

By slightly changing the different parameters in the methods used and obtaining automated feedback from the users, a search engine or an information retrieval system, can collect various measurements about the performance of different methods with varying parameters. In time, a gradient descent like approach can be used by slightly changing the parameters and by modifying them so that the performance of the methods is improved and adapted to the current status of the web and the users.

Another approach is to divide the parameter space into larger bins in the beginning, and to evaluate the performance of different methods under these subdivisions. Once sufficient samples are collected, one can select the best bins, sample them in finer detail, and continue doing so iteratively in a pyramidal fashion, assuming that the problem is suitable for a greedy approach. As a reality check, one can constantly sample larger bins for changes in behavior. If the previously selected best bins are different than current ones, then the new best bins can be sampled in finer detail to calculate the current optimal parameters.

For instance, in a meta search engine that uses a linear combination of scores approach, which assigns different weights to different search engines, these weights can be sampled and divided into different bins. A small fraction of the queries can be used to sample these bins and to see if there are significant differences between the performance of current set of weights vs. the weights suggested by the bins sampled. When a search engine that is used by the meta search engine makes a change in its ranking method, such as introducing a new heuristic, one may expect that the optimal weights in linear combination of scores approach may significantly deviate from the previous optimal weights. By constantly sampling the parameter space and refining the best bins by sampling them in finer details, a meta search engine can quickly anticipate the changes made in the search engines and modify its model to stay up to date.

5.2 Selection of methods and parameters given a priori knowledge

A related topic of interest is how to select the best set of parameters or methods that should be used under different types of queries. If the performance of different methods can be estimated by looking at the type of the query and other a priori information about the query, and if we have an expectation about which methods are best suited with which set of parameters for the particular query type of topic, then the best method and its parameters can be selected with respect to that belief. Our experience with different merging and reranking approaches used in our meta search engine suggested that

no single method is best for all types of queries. If we have a priori knowledge such as number of terms in the query or the topic of the query, it may be possible to select the method that is expected to be the best given a particular query type. Automated evaluation methods may enable sampling of various query types in finer detail and learning the selection of best methods and their parameters for various cases.

A related application is the source selection problem in meta search engines, which can be summarized as the problem of selecting the best set of search engines that we expect will respond with high quality results for the current query. Again, a priori knowledge such as the number of terms in the query and context of the query that may be guessed by examining the query terms, can be treated as the a priori knowledge. The system can learn in time which sources have a higher chance of returning high quality documents for a number of concepts or terms that can be selected. The expectancies can also be automatically updated in time, as the databases of different engines as well as the ranking methods they use change.

6. CONCLUSION

In this paper we introduced average position of clicks metric, which can be used in conjunction with implicit and partial relevance feedback information. Various bias sources that may affect the metric are discussed and the metric is compared to average uninterpolated precision, a widely used metric when explicit user judgements are available. We highlighted key differences between these two metrics, and provided insight on why the proposed metric may be preferred with implicit and partial relevance information.

Average position of clicks can be used to compare the performance of different ranking/reranking methods. This approach does not put extra burden on the users and can be used to gather very large amount of samples to be processed automatically. Search engines and meta search engines can use the proposed metric to learn the best method or the best parameters for different types of queries. Furthermore, unlike methods that rely on analyzing the documents via machine computable measurements to come up with the relevance judgements, the proposed approach uses implicit relevance judgements of real users issuing real queries.

The proposed method is also quite non-intrusive. Identifying users or query topics is not required. It only needs information about the selected method, the parameters used for different queries, and the positions of the documents selected. Majority of the search engines already have the infrastructure to collect these statistics, and we believe that most users can provide this information without feeling loss of privacy.

Finally, an objective and automated way of evaluating the performance of different methods makes it possible to compare different ranking/reranking methods, to evaluate the choice of different parameters, and to enable new learning and exploration approaches that were previously unpractical using explicit relevance information. In this paper we introduced one such method that could be used with implicit relevance feedback information. Although not perfect, it has the potential to combine best of both worlds: we still use user judgements but in an implicit way, and we can collect a large number of samples that will offer greater statistical significance.

7. REFERENCES

[1] Directhit. http://www.directhit.com/.

[2] Daniel Dreilinger and Adele E. Howe. Experiences with selecting search engines using metasearch. *ACM Transactions on Information Systems*, 15(3):195–222, 1997.

[3] Adele E. Howe and Daniel Dreilinger. SAVVYSEARCH: A metasearch engine that learns which search engines to query. *AI Magazine*, 18(2):19–25, 1997.

[4] Jinmook Kim, Douglas W. Oard, and Kathleen Romanik. User modeling for information filtering based on implicit feedback. In *Proceedings of ISKO-France 2001, July 5–6, Nanterre, France.*

[5] Longzhuang Li and Li Shang. Statistical performance evaluation of search engines. In *WWW10 conference posters, May 2–5, 2001, Hong Kong.*

[6] Mearf. http://mearf.cs.umn.edu/.

[7] Metacrawler. http://www.metacrawler.com/.

[8] B. Uygar Oztekin, George Karypis, and Vipin Kumar. Expert agreement and content based reranking in a meta search environment using Mearf. In *procedings of the Eleventh International World Wide Web Conference, May 7–11, 2002, Honolulu, Hawaii*, pages 333–344. ACM Press.

[9] Profusion. http://www.profusion.com/.

[10] Agustin Schapira. Collaboratively searching the web – an initial study. Master's thesis, 1999.

[11] E. Selberg and O. Etzioni. Multi-service search and comparison using the MetaCrawler. In *Proceedings of the 4th International World-Wide Web Conference*, Darmstadt, Germany, Dec 1995.

[12] Mario Gomez Susan Gauch, Guijun Wang. Profusion: Intelligent fusion from multiple, distributed search engines. *Journal of Universal Computer Science*, 2(9):637–649, 1996.

[13] Pang-Ning Tan and Vipin Kumar. Discovery of web robot sessions based on their navigational patterns. In *Data Mining and Knowledge Discovery*, volume 6, pages 9–35, 2002.

[14] Text REtrieval Conference, TREC. http://trec.nist.gov/.

[15] Christopher C. Vogt and Garrison W. Cottrell. Fusion via a linear combination of scores. *Information Retrieval*, 1(3):151–173, 1999.

[16] Byoung-Tak Zhang and Young-Woo Seo. Personalized web-document filtering using reinforcement learning. *Applied Artificial Intelligence*, 15(7):665–685, 2001.