

Towards Timed Predictions of Human Performance for Interactive Information Retrieval Evaluation

Mark D. Smucker
Department of Management Sciences
University of Waterloo
msmucker@uwaterloo.ca

ABSTRACT

Today's popular retrieval metrics are largely divorced from any notion of a user interface or a user model. These metrics such as mean average precision produce measures of ranked results quality rather than predictions of human performance. Using GOMS, we modify the Cranfield-style of evaluation to create a new evaluation method that makes testable predictions of human performance. While not yet validated by user studies, we demonstrate using our evaluation method that such an evaluation technique gives information retrieval researchers the ability to understand how changes in the interface or in the underlying retrieval algorithm impact user performance. Future work should be directed to the creation and validation of evaluation methods that predict user performance and incorporate explicit user interfaces and user models.

1. INTRODUCTION

While the information retrieval (IR) community has known since the work of Dunlop [8] that IR evaluation could be improved with automated usability methods from the field of human computer interaction (HCI) [9], retrieval metrics devoid of explicit user interfaces and user models continue to dominate IR evaluation.

As a step towards answering our call to move Cranfield-style evaluation towards a more realistic evaluation [14], we use GOMS, an automated usability method, to create an evaluation method that makes testable predictions of human performance. Cranfield-style evaluation measures the ranking quality of a retrieval algorithm given a test collection of documents, search topics, and relevance judgments. An example of a commonly used metric in Cranfield-style evaluation is precision at rank 10 (P10). The precision at rank 10 is equal to number of relevant documents found within the first 10 documents returned by a retrieval algorithm divided by 10. While these metrics can be somewhat correlated with user performance [1, 2, 15] they do not make testable predictions of user performance.

GOMS estimates the time for expert users to complete a task given a certain interface [4, 10]. The acronym GOMS stands for Goals, Operators, Methods, and Selections. In simple terms, GOMS is about finding the sequence of operations on a user interface that allows the user to achieve the user's goal in the shortest amount of time. GOMS allows an

interface designer to obtain predicted task times for different interfaces before more expensive user testing.

In our case, the IR user has a goal of finding as many relevant documents as possible. The operations are the actions possible with a hypothetical user interface. We embody "methods and selections" in what we refer to as a *user model*. The user model we create in this paper is a simple, first step towards better models. For example, our model lacks the ability to perform query reformulation. Considerable research effort will be required to create user models based on observed user behavior. Even with a better user model, the overall evaluation methodology will need to be validated with user studies to determine the accuracy of the performance predictions [5, 6].

By combining GOMS with the Cranfield-style of evaluation, we obtain a simulation of user behavior for which all user actions have associated times. For example, from GOMS we know that moving the mouse to a button will take on average 1.1 seconds [11]. From this simulation many testable measures of human performance are computable. In this paper, we compute the number of relevant documents read by the simulated user within 10 minutes.

Evaluation methods, such as ours, that explicitly incorporate a user interface and a user model allow IR researchers to investigate the impact of interface changes on user performance before turning to more expensive user studies for confirmation. In other words, IR researchers can simulate user behavior over a hypothetical user interface to generate testable hypotheses. For example, based on our experimental results, we hypothesize that the user interface determines the relationship between ranked retrieval quality and user performance.

Next we describe our method in more detail and then follow with our experiments and preliminary results.

2. METHODS AND MATERIALS

Our evaluation methodology consists of a hypothetical user interface and a user model defined over that interface. Our hypothetical interface is a simplified version of today's common web search interface. The interface provides a text box that allows the user to enter and submit a keyword-like query. On submission of the query, the user is presented with 10 query-biased summaries of the top ranked results produced by an underlying retrieval algorithm in response to the query. Each result summary provides a hyperlink or button that when clicked on will take the user to the full document. The interface provides the means for the user to hit a "back button" and return to the search results. The

Let t be the total search time.
Enter query and hit return. ($t \leftarrow t + K(\text{length}(\text{query}) + 1)$)
Wait for results & move hands to mouse. ($t \leftarrow t + W$)
for $i \leftarrow 1$ to Number of Results **do**
 Read and evaluate summary. ($t \leftarrow t + SE$)
 $D \leftarrow$ document at result i
 judgment \leftarrow qrels judgment of D
 if judgment is non-relevant **then**
 With probability $P0$ decide to read D .
 else if judgment is relevant **then**
 With probability $P1$ decide to read D .
 else // judgment is highly relevant
 With probability $P2$ decide to read D .
 end if
 if Decided to read D **then**
 Point mouse to link/button. ($t \leftarrow t + P$)
 Click mouse button. ($t \leftarrow t + BB$)
 Wait for result page to load. ($t \leftarrow t + W$)
 Read and evaluate D . ($t \leftarrow t + DE$)
 if judgment is relevant or judgment is highly relevant
 then
 numRelevantRead \leftarrow numRelevantRead + 1
 end if
 Point mouse to back button. ($t \leftarrow t + P$)
 Click mouse button. ($t \leftarrow t + BB$)
 end if
 if $((i+1) \bmod 10) = 1$ **then** // Only 10 results per page
 Point mouse to next page link/button. ($t \leftarrow t + P$)
 Click mouse button. ($t \leftarrow t + BB$)
 Wait for next page of results. ($t \leftarrow t + W$)
 end if
end for

Figure 1: User model. The time each action takes is shown in parentheses. Table 1 lists the model parameters and their values.

search results interface also provides a link or button to take the user to a new page with the next 10 ranked results.

Figure 1 shows our user model. First, the simulated user enters the query by typing and then waits for the first 10 search results. The user then proceeds to read and evaluate the result summaries one after the other. With some probability conditional on the relevance of the underlying document, the user will decide to click on a summary and read the document. After reading the document, the user hits a “back button” and continues reading and evaluating the search result summaries. When the user reaches the end of the summaries on a page, the user clicks on a link or button to request the next 10 results. All actions have associated times.

Our user model is simple, for demonstration purposes, and not an attempt to capture the complex process of search. For example, while query reformulation could be made possible with our hypothetical interface, our user model is incapable of reformulating queries. Eye tracking research has clearly shown that users quickly reformulate queries that don’t produce top ranked relevant documents [12].

Table 1 lists the parameter settings of our user model. These settings come primarily from two places. For GOMS, we utilize the keystroke level model (KLM) [3]. In this model, the operators are defined at the level of keystrokes

Keystroke (average non-secretarial typist 40 wpm) [11]	$K = 0.28$ s
Type a sequence of n keys [11]	$n \times K$ s
Point the mouse to a target on the display [11]	$P = 1.1$ s
Press or release the mouse button [11]	$B = 0.1$ s
Click mouse button (press and release) [11]	$BB = 0.2$ s
Move hands to keyboard or mouse [11]	$H = 0.4$ s
Mental act of routine thinking or perception [11]	$M = 1.2$ s
Wait for search results or web page to load	$W = 1$ s
Time to evaluate a search result summary [16]	$SE = 19$ s
Time to evaluate a document for relevance [16]	$DE = 88$ s
Probability of clicking on non-relevant summary [16]	$P0 = 0.25$
Probability of clicking on relevant summary [16]	$P1 = 0.53$
Probability of clicking on highly relevant summary [16]	$P2 = 0.77$

Table 1: User model parameters. All times are in seconds. Figure 1 shows the user model.

and mouse movements. Timings for these operators are averages obtained from various user studies [11]. In our use of GOMS, we inadvertently omitted use of the “mental” operator. Even so, most mental actions in our model are involved in the evaluation of the search result summaries and documents and are captured by the SE and DE parameters.

Our other source for parameter settings comes from the work of Turpin, Scholer, Järvelin, Wi and Culpepper [16] who created a methodology to include search result summaries into standard list quality metrics such as precision at 10 (P10) and mean average precision (MAP). As part of their work, they asked users to determine whether or not to click on a summary and view the corresponding document. If the user felt the summary would lead to a relevant document, the user would decide to click on the summary. Users then judged the relevance of documents on a 4 point graded scale. On average, users took 19 seconds to evaluate a summary and 88 seconds to evaluate a document. While we know that eye tracking results show that users usually spend much less than 19 seconds reading a summary [7], we utilize Turpin et al.’s timings to be consistent with their measures of summary evaluation accuracy.

In a simulation analysis of TREC 9 and 10 submitted runs, Turpin et al. mapped their two highest relevance categories to TREC’s “highly relevant” and their least relevant category to “relevant” and finally mapped non-relevant to non-relevant. With this mapping, the probability that a user would click on a summary was 0.77 for highly relevant documents, 0.53 for relevant, and 0.25 for non-relevant. These summary evaluation accuracies are in line with the 75% accuracy found by Sanderson [13]. We use these probabilities in our experiments with the same TREC 9 runs.

3. EXPERIMENTS

For our experiments we use the 40 automatic, title only ad-hoc web retrieval runs from TREC 9. For each run we compute the precision at 10 (P10) as well as the number

Average Improvement over Normal Model	
Condition	Percent Imp.
Perfect Summaries	80%
Read Documents Twice as Fast	38%
Better Summaries	23%
Read Summaries Twice as Fast	17%

Table 2: Results for the 4 interface improvements described in Section 3. For each of the 40 TREC-9 runs, user performance is measured as the number of relevant documents read within 10 minutes.

of relevant documents read by our simulated user within 10 minutes. Because there is inherent randomness in our user model caused by the different probabilities of clicking on a result summary, we simulate usage 1000 times for each topic of each run and average the predicted performance.

In addition, we examine 4 possible interface improvements:

1. We modify the result summaries so that users can evaluate them twice as fast (9.5 s rather than 19 s).
2. We improve the evaluation accuracy of summaries. For relevant and highly relevant documents, the summary evaluation accuracy increases by 25% (0.53 to 0.663 and 0.77 to 0.963) and for non-relevant documents the error rate decreases by 25% (0.25 to 0.188).
3. We provide some means for the users to evaluate documents twice as fast (44 s rather than 88 s).
4. We make summaries perfect. All relevant and highly relevant documents are viewed, and users waste no time reading non-relevant documents. While likely an impossible interface improvement if evaluation time remains unchanged, this change allows us to see the maximum possible gain for improvements in summary evaluation accuracy.

We naively assume all interface improvements do not affect other aspects of the search process. For example, for improvement 1 above, users can evaluate summaries faster with no decrease in evaluation accuracy.

4. RESULTS AND DISCUSSION

Table 2 shows that our evaluation method predicts that each of the interface improvements would increase the number of relevant documents evaluated by the user within 10 minutes.

While our interface improvements are all “what-if” experiments, we can see in Figure 2 that under the assumptions of our evaluation method, the user interface determines the relationship between ranked retrieval quality and user performance. What good is a 20% improvement in P10? The answer depends on the quality of the user interface. Better interfaces better translate retrieval gains into user performance gains.

Based on Figure 2, should we conclude that P10 is a metric that mirrors user performance when performance is defined to be the number of relevant documents examined within 10 minutes? No. We’ve replaced one evaluation method of retrieval quality with another but neither have been validated against actual human performance.

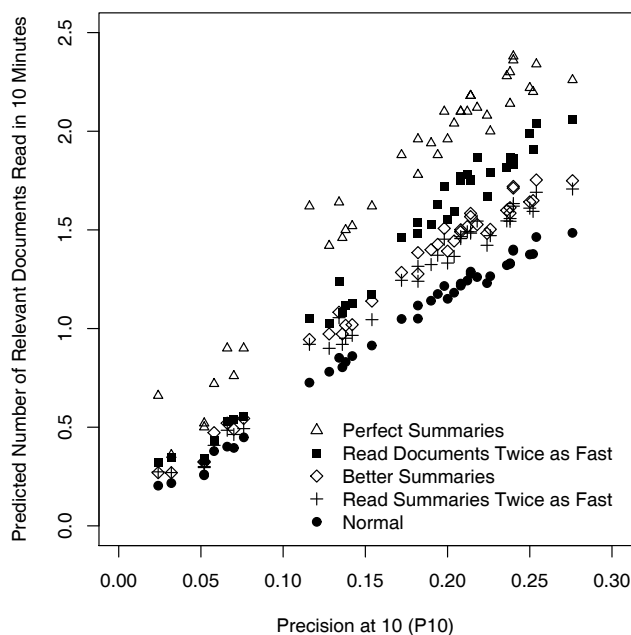


Figure 2: This figure shows the precision at 10 (P10) vs. the predicted number of relevant documents read within 10 minutes for each of the 40 TREC 9 runs and the 5 interface conditions described in Sections 2 and 3.

What we have with our new evaluation method is a method that aims to be directly predictive of the variable of concern: human performance. Precision at 10 or MAP does not attempt to predict human performance. P10 and MAP and metrics like them output a measure of list quality that is loosely coupled with user performance.

What a method like our simple example does is that it marries together retrieval quality, a user model, and the hypothetical user interface and makes a prediction concerning user performance. All of these 4 important parts of an evaluation of retrieval performance are explicit in our evaluation.

The significant shift in thinking that our evaluation method brings about is that when an evaluation method contains all of these components, we gain the ability to start asking questions about what will most improve human performance. In other words, we can look to see where the user is spending time. Is the most time spent manipulating the interface? Or is it spent wading through non-relevant documents? Or is it spent reading documents? Our evaluation method allows the IR researcher to gain insight to these questions.

5. CONCLUSION

We combined an automated usability method, GOMS, with the Cranfield-style of evaluation to produce a new evaluation method that produces testable predictions of human performance. This evaluation method allows IR researchers to investigate the impact of various interface improvements and also to see the degree to which changes in retrieval quality affect user performance. Future work remains to create accurate, predictive evaluation methods that explicitly incorporate both the user interface and a model of the user’s search behavior.

6. ACKNOWLEDGMENTS

This work was supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC), and in part by the University of Waterloo. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect those of the sponsors.

7. REFERENCES

- [1] A. Al-Maskari, M. Sanderson, P. Clough, and E. Airio. The good and the bad system: does the test collection predict users' effectiveness? In *SIGIR'08*, pages 59–66. ACM, 2008.
- [2] J. Allan, B. Carterette, and J. Lewis. When will information retrieval be “good enough”? In *SIGIR'05*, pages 433–440. ACM, 2005.
- [3] S. K. Card, T. P. Moran, and A. Newell. The keystroke-level model for user performance time with interactive systems. *CACM*, 23(7):396–410, 1980.
- [4] S. K. Card, A. Newell, and T. P. Moran. *The Psychology of Human-Computer Interaction*. Lawrence Erlbaum Associates, Inc., Mahwah, NJ, USA, 1983.
- [5] W. S. Cooper. On selecting a measure of retrieval effectiveness. *JASIS*, 24(2):87–100, Mar/Apr 1973.
- [6] W. S. Cooper. On selecting a measure of retrieval effectiveness: Part ii. implementation of the philosophy. *JASIS*, 24(6):413–424, Nov/Dec 1973.
- [7] E. Cutrell and Z. Guan. What are you looking for?: an eye-tracking study of information usage in web search. In *CHI'07*, pages 407–416. ACM, 2007.
- [8] M. D. Dunlop. Time, relevance and interaction modelling for information retrieval. In *SIGIR'97*, pages 206–213. ACM, 1997.
- [9] M. Y. Ivory and M. A. Hearst. The state of the art in automating usability evaluation of user interfaces. *ACM Computing Surveys*, 33(4):470–516, 2001.
- [10] B. E. John and D. E. Kieras. Using GOMS for user interface design and evaluation: which technique? *ACM Transactions on Computer-Human Interaction*, 3(4):287–319, 1996.
- [11] D. Kieras. Using the keystroke-level model to estimate execution times. <ftp://ftp.eecs.umich.edu/people/kieras/GOMS/KLM.pdf>, copy obtained via Google, <http://74.125.95.132/search?q=cache:wvKGAm5KIIJ:ftp://ftp.eecs.umich.edu/people/kieras/GOMS/KLM.pdf,2001>.
- [12] L. Lorigo, M. Haridasan, H. Brynjarsdóttir, L. Xia, T. Joachims, G. Gay, L. Granka, F. Pellacini, and B. Pan. Eye tracking and online search: Lessons learned and challenges ahead. *JASIS*, 59(7):1041–1052, 2008.
- [13] M. Sanderson. Accurate user directed summarization from existing tools. In *CIKM'98*, pages 45–51. ACM, 1998.
- [14] M. D. Smucker. A plan for making information retrieval evaluation synonymous with human performance prediction. In *Proceedings of the SIGIR 2009 Workshop on the Future of IR Evaluation*, pages 11–12, July 2009.
- [15] A. Turpin and F. Scholer. User performance versus precision measures for simple search tasks. In *SIGIR'06*, pages 11–18. ACM, 2006.
- [16] A. Turpin, F. Scholer, K. Jarvelin, M. Wu, and J. S. Culpepper. Including summaries in system evaluation. In *SIGIR'09*, pages 508–515. ACM, 2009.