Modeling User Variance in Time-Biased Gain

Mark D. Smucker
Department of Management Sciences
University of Waterloo, Canada
mark.smucker@uwaterloo.ca

Charles L. A. Clarke
School of Computer Science
University of Waterloo, Canada
claclark@plg.uwaterloo.ca

ABSTRACT

Cranfield-style information retrieval evaluation considers variance in user information needs by evaluating retrieval systems over a set of search topics. For each search topic, traditional metrics model all users searching ranked lists in exactly the same manner and thus have zero variance in their per-topic estimate of effectiveness. Metrics that fail to model user variance overestimate the effect size of differences between retrieval systems. The modeling of user variance is critical to understanding the impact of effectiveness differences on the actual user experience. If the variance of a difference is high, the effect on user experience will be low. Time-biased gain is an evaluation metric that models user interaction with ranked lists that are displayed using document surrogates. In this paper, we extend the stochastic simulation of time-biased gain to model the variation between users. We validate this new version of time-biased gain by showing that it produces distributions of gain that agree well with actual distributions produced by real users. With a per-topic variance in its effectiveness measure, time-biased gain allows for the measurement of the effect size of differences, which allows researchers to understand the extent to which predicted performance improvements matter to real users.

Author Keywords

Information retrieval; search evaluation

ACM Classification Keywords

H.3.4 Information Storage and Retrieval: Systems and Software—*Performance evaluation (efficiency and effectiveness)*

INTRODUCTION

User interfaces of major commercial search engines typically share a common interaction paradigm. In response to a query, these retrieval systems display their results as a ranked list of surrogates. In the case of document retrieval, these surrogates include a document title, a URL, and a query-biased summary (or snippet) which taken together we will simply call a *summary*. Clicking on a summary takes the user to the summary's corresponding full document. The user can easily navigate back and forth between the summaries and full documents, examining the results as they wish and stopping whenever they want. Colloquially, this interaction paradigm has come to be known as *ten blue links*. We are interested in understanding and modeling user behavior on these ten blue links. With accurate models of user behavior, we can predict user performance in an automated fashion and evaluate the utility of ranking algorithms.

In this paper, we use stochastic simulation to model user behavior on ten blue link interfaces. In particular, we use simulation to estimate the expected number of relevant documents that a user will find when processing a single ranked list. More generally, we can predict the *distribution* of the number of relevant documents found by a population of users.

When we use simulation to estimate the expected number of relevant documents found by a user, the simulation acts as a Cranfield-styled evaluation metric called *time-biased gain* (TBG) [23, 24]. Cranfield-style evaluation, named for the experiments conducted by Cleverdon at Cranfield University in the 1960s, forms the primary evaluation methodology employed at TREC and other evaluation efforts [28]. The research literature sometimes refers to Cranfield-style evaluation as "batch evaluation" in order to contrast it with "interactive evaluation" and other user-oriented methodologies.

Cranfield-style evaluation is characterized by the development of reusable test collections for evaluating ranking functions and other specific search engine components. A test collection consists of relevance judgments for a fixed set of documents with respect to a fixed set of queries. To evaluate a search engine, we execute the queries over the documents to generate a ranked result list, and then apply the judgments to compute traditional retrieval effectiveness measures, including mean average precision (MAP) [20] and normalized discounted cumulative gain (nDCG) [13]. The reusable nature of these test collections permits us to re-compute the measures as often as needed for tasks such as parameter tuning, intra-system comparisons, and learning to rank.

In the human computer information retrieval (HCIR) community, Cranfield-style evaluation is widely viewed as an inadequate, and possibly misleading, substitute for actual experiments with users. For example, in their classic paper Hersh et al. [12] ask "Do improvements in system performance demonstrated by batch evaluation confer the same benefits for real users?" System-oriented researchers working on improved ranking algorithms frequently publish results showing improvements in metrics such as MAP, while HCIR researchers generally feel that MAP does not predict user performance on interactive retrieval systems. What good is an improvement in MAP, or any other evaluation metric, if the metric does not reflect improved user performance?

HCIR'12, October 4–5, 2012, Cambridge, MA, USA.

Our work builds on that of Smucker and Jethani [25] who contend that the low predictive power of many Cranfield-style metrics comes from their lack of realism. Indeed, most metrics have little or no model of a user or of a user interface. Smucker and Jethani investigated precision as a metric given that the concept of precision is integral to many metrics. They found evidence that as the user interface became more complex, precision became less predictive of user performance. Precision as a metric assumes that users move at a constant rate down a ranked list and that all relevant documents are recognized as such. Neither of these assumptions are realistic, and they become less realistic as interfaces allow more interaction.

We designed time-biased gain to allow us to more realistically model users for the purpose of measuring retrieval system effectiveness in human terms. This paper's version of time-biased gain has a ten blue links styled hypothetical user interface and a user model that reflects that user actions take time and not all user decisions are error free. Different versions of time-biased gain can be created for different interfaces and usage scenarios.

Real users are variable. Different users have different information needs, and this aspect of user variation is commonly part of IR evaluation in the form of a set of search topics. Beyond the search topic, we know that users have different abilities and different strategies that they employ when conducting a search. In addition to variation across users, there is also variation in decisions and the times to make decisions by an individual user.

While real users are variable, most batch evaluation metrics purposely have no notion of user variability. Traditional Cranfield- and TREC-style tests adopt a "ruthless abstraction of the user" [27], as nothing more than a list of relevance judgments with respect to some query. According to Voorhees [27], this abstraction may be seen as an important strength of the Cranfield approach, providing "sufficient fidelity to real user tasks to be informative", while being "broadly applicable, feasible to implement, and comparatively inexpensive." In particular, this abstraction strips away all issues of user variability. For any given result list, this abstract user always behaves in exactly the same way, and thus, for a single result list, traditional effectiveness measures produce a single number, with no associated variance.

Of course, in reality different users do experience the same result list differently. In this work, in addition to variability of search topics, our stochastic simulation of time-biased gain models variation in both the actions of an individual user and variation across different users to produce per-topic variance in the effectiveness measure. As we shall see, if this variance accurately reflects the user population, it provides insights into system effectiveness not provided by traditional measures.

Modeling user variance does not limit our ability to detect small performance improvements, for our simulation allows us to take an unlimited number of samples and thus can produce a precise point estimate. Having a good estimate of the variance allows us to compute and understand the *effect size* of performance differences between systems. If the variance is large relative to the difference, the effect size is reduced, which means fewer users will experience the difference between the systems. Without the estimation of variance, such effect size computations are limited to the estimation of variance across topics, which underestimates the actual variance in performance.

The main contribution of this paper is an effectiveness measure that both predicts the number and distribution of relevant documents that a user will save when processing a ranked list with a ten blue links interface. By producing a per-topic distribution of the number of saved relevant documents, we can talk about whether or not differences in systems will matter to users by measuring the effect size of differences on a per topic basis.

We next describe time-biased gain, its simulation of user behavior, and how we predict the distribution of the number of relevant documents saved by a user processing a ranked list of documents. We then show how time-biased gain allows us to measure the impact of performance improvements on the user experience. Related work is reviewed before the conclusion of the paper.

TIME-BIASED GAIN

As described in the introduction, our goal is to predict the distribution of the number of relevant documents that a user will save when processing a ranked list using a ten blue links styled interface. We will refer to the number of relevant documents saved as simply the cumulative gain.

We consider a user to accumulate gain over time. Let us define a function $G(t)$ that represents the cumulative gain over time t. Given that $G(t)$ is a function of time, if we have a probability density function $f(t)$ that gives us the distribution of how long a user is likely to work, then we can compute the expected gain as follows:

$$E[G(t)] = \int_0^\infty G(t)f(t)dt, \tag{1}$$

which represents time-biased gain in its general form.

Equation 1 says nothing about the form of gain or how it is accumulated. In this paper we consider each relevant document to have a gain of 1, and thus Equation 1 represents the expected number of relevant documents saved by a user, but other assumptions and scenarios are possible. In a Web search context, if a user has a navigational need then $G(t)$ might have a single step, occurring when the target page is found. In some cases, $G(t)$ might be partially continuous, if gain is realized by viewing a video or listening to music. Under another approach, we might measure gain according to the number of informational nuggets [7, 19] encountered as documents are read. Likewise, time-biased gain as described by Equation 1 is not restricted to the processing of ranked lists of documents. Time-biased gain could be applied to the whole search process, from the moment a user starts seeking information to the moment the user stops. We leave these ideas for future work.

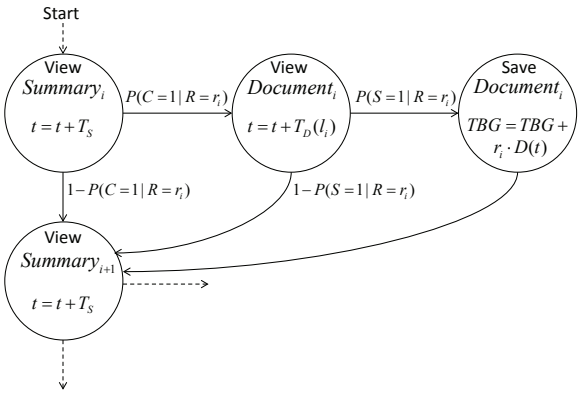

Figure 1. User model. Table 1 describes the parameters.

A user processing a ranked list does so according to some process. This process is complex, and our work can only start to scratch the surface of how to model user behavior. To simplify our model, at the cost of some accuracy, we choose to model users as proceeding down the rank list, one document at a time. On the whole, this simplification is a good approximation to reality. In the user data we use from the experiment of Smucker and Jethani [25], 94% of the users' moves through the list were to a lower ranked document.

With user motion fixed to be down a ranked list one document at a time, we can express Equation 1 as:

$$TBG(L,U) = \sum_{k=1}^{|L|} g_k D(T(k)), \tag{2}$$

where $TBG(L,U)$ is the time-biased gain for the ranked list L of length $|L|$ with the user model U, $T(k)$ is the time it takes for a user to finish processing the document at rank k and realize its gain g_k, and $D(t)$ is the survival probability, i.e. the probability that a user survives to time t. The gain of a document, g_k, depends on whether or not the document is saved as relevant by a user. In our model, a user only realizes gain when the user saves a NIST judged relevant document. A single saved relevant document has a gain of 1. We call $D(t)$ the *decay* function. Appendix A explains how we get from Equation 1 to 2.

To compute Equation 2, we need to model the gain achieved at rank k, g_k, the time it takes to achieve that gain, $T(k)$, and also the probability of surviving to rank k, $D(t)$. Rather than formulate an analytic expression for g_k and $T(k)$, we use a previously developed stochastic simulation [23] to determine when a simulated user saves a relevant at rank k.

Our simulation models the time it takes users to complete various actions as well as the probabilities with which they make various decisions. Figure 1 represents our model as a semi-Markov model. Now, the task of determining g_k and $T(k)$ are replaced with determining the probabilities of clicking on a summary and of saving a document, how long it takes to process a document summary, T_S, and how long it takes to process a full document of length l words, $T_D(l)$. We model the time to process a summary as a Weibull distribution. The time to process a full document is modeled as a log-normal linear distribution based on the document's length and whether or not it is a duplicate of a higher ranked document. In the next section, we explain how we calibrate these components of the simulation.

Parameter	Description
t	Accumulated time.
T_S	Time spent viewing a document summary. Modeled as a random deviate drawn from a Weibull distribution.
$P(C=1\|R=r_i)$	Probability of clicking on a summary given its NIST relevance.
$T_D(l_i)$	Time spent viewing full document at rank i, which contains l_i words. Modeled as a random deviate drawn from a log-normal linear distribution. Duplicates modeled as a random deviate drawn from a log-normal distribution.
$P(S=1\|R=r_i)$	Probability of saving a viewed document given its NIST relevance.
r_i	The NIST relevance for the document at rank i, where $r_i = 1$ if the document is relevant, $r_i = 0$ otherwise.
$D(t)$	Decay function. Probability of surviving to time t.
TBG	Time-biased gain. The expected number of saved relevant documents.

Table 1. User model parameters. Figure 1 shows the user model.

Running the simulation once produces a single sample of time-biased gain for the given list and user model. Repeatedly running the simulation will produce a distribution of $TBG(L,U)$ values for the list and user model. The mean of this distribution produces an estimate of the expected gain. If we only have a single user model, the distribution of TBG will be representative of the variance in the decisions of a user and the time it takes to make those decisions, which was the limit of the variance we previously modeled [23]. In this paper, our goal is to produce a distribution of the gain for a population of users.

The semi-Markov model of Figure 1 is too simple to correctly model a user population's variance. We can either choose to create a new, more complex model, or we can divide a population into homogeneous subsets and use the simple model for each subset [29]. We have chosen to do the latter.

To model a population of users, $\mathcal{U}$, we will create a set U of N user models, where each user model U_i represents a different result list processing strategy. To estimate TBG for a single result list L, we will take B samples of user behavior and then average these samples:

$$TBG(L) = \frac{1}{B}\sum^{B} TBG(L, U_i : i = Random(1,N)) \tag{3}$$

where $Random(1, N)$ returns a random integer in the range $[1, N]$, and $TBG(L, U_i)$ is a single random sample of TBG for list L with user model U_i. A single user model U_i can produce as many different random samples of TBG as we want. As such, our ability to precisely estimate TBG is only limited by the number of samples B that we produce. The B samples produce for us a per-topic distribution of retrieval effectiveness that is representative of the user population $\mathcal{U}$ modeled by the N models of U.

Calibration of Model Components

We use data from phase 2 of the user study of Smucker and Jethani [25] to calibrate the simulation. In phase 2 of that study, 48 participants searched ranked lists for relevant documents using a ten blue links styled interface. The summaries page displayed 10 summaries and provided links for navigating to the next and previous 10 summaries in the ranked list. Clicking on a summary took the participant to a page that displayed the full document. On this page, the participant could save the document as relevant if desired, but the participant was not required to make an explicit relevance judgment.

Each participant worked for 10 minutes on each of 4 TREC 2005 Robust track [26] search topics. Using language similar to that of Smith and Kantor [21], participants were instructed to "try to find and save as many relevant documents as possible in the 10 minutes while saving as few non-relevant documents as possible." The result lists contained duplicates, and participants were instructed to judge duplicate documents the same. Documents were from the AQUAINT newswire collection. The result lists had two precisions: 0.3 and 0.6. The lists were designed to mimic actual ranked lists in that documents likely to be ranked highly by search engines were ranked higher. A 0.3 precision list had 3 relevant documents for each 10 documents displayed, and, likewise, 0.6 precision had 6.

In total, the study used 8 search topics (310, 336, 362, 367, 383, 426, 427, 436) and the topics were balanced across search tasks, list precision, and participants. Participants searched each list for 10 minutes. In addition to the original report on the study [25], which includes full details of the user study, we have reported on the variety of list processing strategies [22] observed. We believe that the user behavior in the Smucker and Jethani study is representative of actual result list processing behavior given the similarity between the study's observed behavior and that reported by other researchers (cf. strategies in Aula [1] and Dumais et al. [9], probabilities of clicking on summaries in Yilmaz et al. [31]).

Our stochastic simulation needs to capture variance in behavior at both the individual user level as well as the population level, i.e. across users. For each participant in our study, we will create a single user model. In other words, we will have $N = 48$ user models in our population U, where each of the models corresponds to one of the participants in our study.

To calibrate each of the user models, we will use the same process as we have previously described [23], but we will restrict the calibration of each of the 48 models to a specific participant's data.

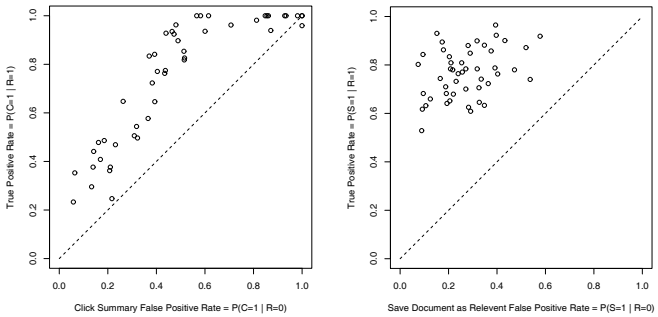

Figure 2. Probabilities of clicking on summaries and saving relevant documents for the 48 user models.

The model shown in Figure 1, has a probabilistic transition between the view-summary state and the view-document state and there is another probabilistic transition between the view-document state and the save-document state. We model the probability of clicking on a summary, $P(C|R)$, and the probability of saving a document, $P(S|R)$, as conditional on the document's NIST relevance. To estimate the participant's overall probabilities, we compute a weighted average of the per-topic estimates. For a search topic's viewed documents, we can compute the fraction of NIST relevant documents that are saved and this is $P(S = 1|R = 1)$. Likewise, we can easily compute $P(S = 1|R = 0)$. For the probability of clicking on a summary, we make an assumption that all summaries up to the last clicked summary are viewed by a participant. With this assumption, we can compute a search topic's $P(C|R)$ as we did for $P(S|R)$. In computing the final weighted averages of the probabilities, we weight a topic's probabilities by the number of viewed documents or summaries.

Figure 2 shows the computed values for the click and save probabilities for the 48 user models plotted in ROC space. Points above the dashed line represent an ability to discriminate between relevant and non-relevant summaries and documents. User models with $P(C = 1|R = 1) = 1$ and $P(C = 1|R = 0) = 1$ represent users that click on every summary. Points closer to the upper left corner show increasing abilities to discriminate between relevant and non-relevant documents.

The web-based system used in the user study recorded the time spent on each view of the summaries page and the full document page. To estimate the amount of time spent on a summary, T_S, we need to allocate the recorded times to all the viewed summaries, which we again assume includes all summaries up to the last clicked summary. To do this, we spread each recorded time on the summaries page across $M_t/|S_t|$ summaries, where M_t is the maximum rank reached by the participant on topic t and $|S_t|$ is the number of recorded times. We then fit a Weibull distribution to all of the summaries' times to allow us to simulate the variance in summary viewing times for a participant by drawing random deviates from the corresponding Weibull distribution.

For all of the fit Weibull distributions, we compared the mean of the user model fit with the mean of the participant and found good agreement in all cases. Figure 3 shows two exam-

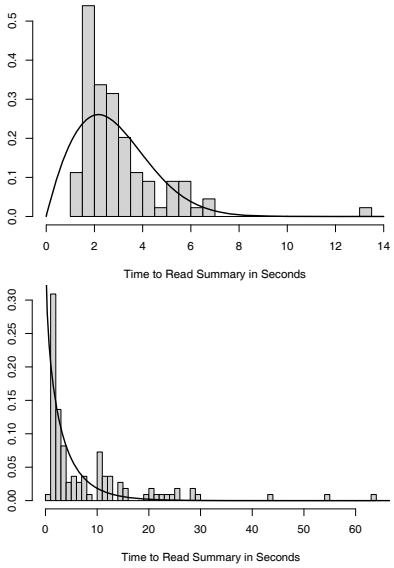

Figure 3. Examples of the distributions of summary times for single participants shown as histograms and the maximum likelihood fit Weibull probability density function for each of the corresponding user models.

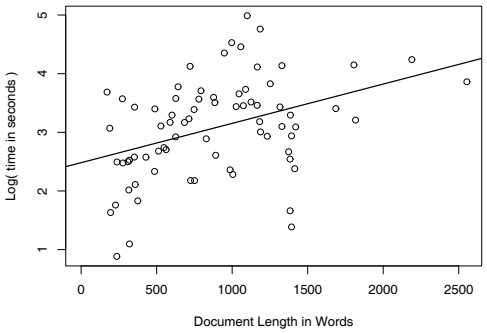

Figure 4. Example of a linear fit of time to judge a document for one user.

ples of the distribution of summary times and the corresponding Weibull probability density function. The top example was the most common style of fit, while in a limited number of cases the fit Weibull distribution took on the shape of an exponential as show in the bottom example. The exponential-like distribution has the unfortunate property of having us model a participant as sometimes taking zero time to process a summary. In the future, we may use smoothed empirical distributions rather than fit a known distribution.

In the case of the time spent viewing a document, T_D, we have found that participants spend more time on longer documents where the document length is measured in words. We have also found that the time to judge a duplicate document is independent of its length. As such, we separately model the time to judge a first viewed document and viewings of duplicates lower in the ranked list. For each participant, we fit the log of their times spent viewing "first viewed" documents as a linear function of the documents' lengths. Such a fit produces a model with slope a, intercept b, and standard error of the residuals σ_f. Individual participants in general have not judged enough duplicates for us to estimate individual distributions of the time spent viewing duplicates. As such, we fit a single log-normal distribution to all users' recorded times spent viewing duplicates. We have found that incorporating the document's NIST relevance or the precision of the ranked list into the model does not increase the variance explained when document length is available.

To produce a random deviate for the time spent viewing a document a document, we use the following function:

$$T_D(l) = \exp(al + b + \sigma_f u)F + \exp(\mu_d + \sigma_d u)(1 - F) \quad (4)$$

where u is a random deviate drawn from a normal distribution with a mean of zero and a variance of 1, l is the length of the document in words, and F is a binary indicator that is 1 if the document is a first viewed document or 0 if a duplicate. The parameters: a, b, and σ_f are for the log-normal linear fit, and the parameters: μ_d and σ_d are the parameters of the log-normal distribution for the duplicates. When we fit each of the 48 user models, in one case, a was slightly less than zero, and we set a to zero to represent that the participant's time to judge documents appeared to have no correlation with document length.

Figure 4 shows an example fit for one participant. As can be seen, there is considerable variance in the time to judge documents and our model only captures two of the many variables that affect the time a user will spend viewing a document.

Our decay function, $D(t)$, is based on data from a commercial search engine log provided by Microsoft to researchers in 2006 and 2007. Coming from a separate dataset, we do not have the ability to fit it separately to each of the 48 models, and we use the same fit as in Smucker and Clarke [24]:

$$D(t) = e^{-t\frac{\ln 2}{h}}, \quad (5)$$

where h is the half-life of users. The half-time corresponds to the the time at which half of the population of users has stopped processing the ranked lists. In our experiments, h equals 224 seconds.

VALIDATION OF SIMULATION

After the design of a simulation and its calibration, we must validate it to confirm that the simulation performs correctly and can make useful predictions. In the case of the simulation described in this paper, we want to determine if it produces a distribution of gain values that is a good fit to observed values from the user study that was used to calibrate it.

We need to validate the distribution of gain after a period of time. Each participant searched for 10 minutes on each search topic. We will compare the actual distribution of gain after 10 minutes to the predicted distribution after 10 minutes. To match the user study, we will not apply any decay to the computation of TBG.

In order to compare a distribution of gain values from the user study against a distribution from the simulation, we need a distribution from the user study with enough data for the

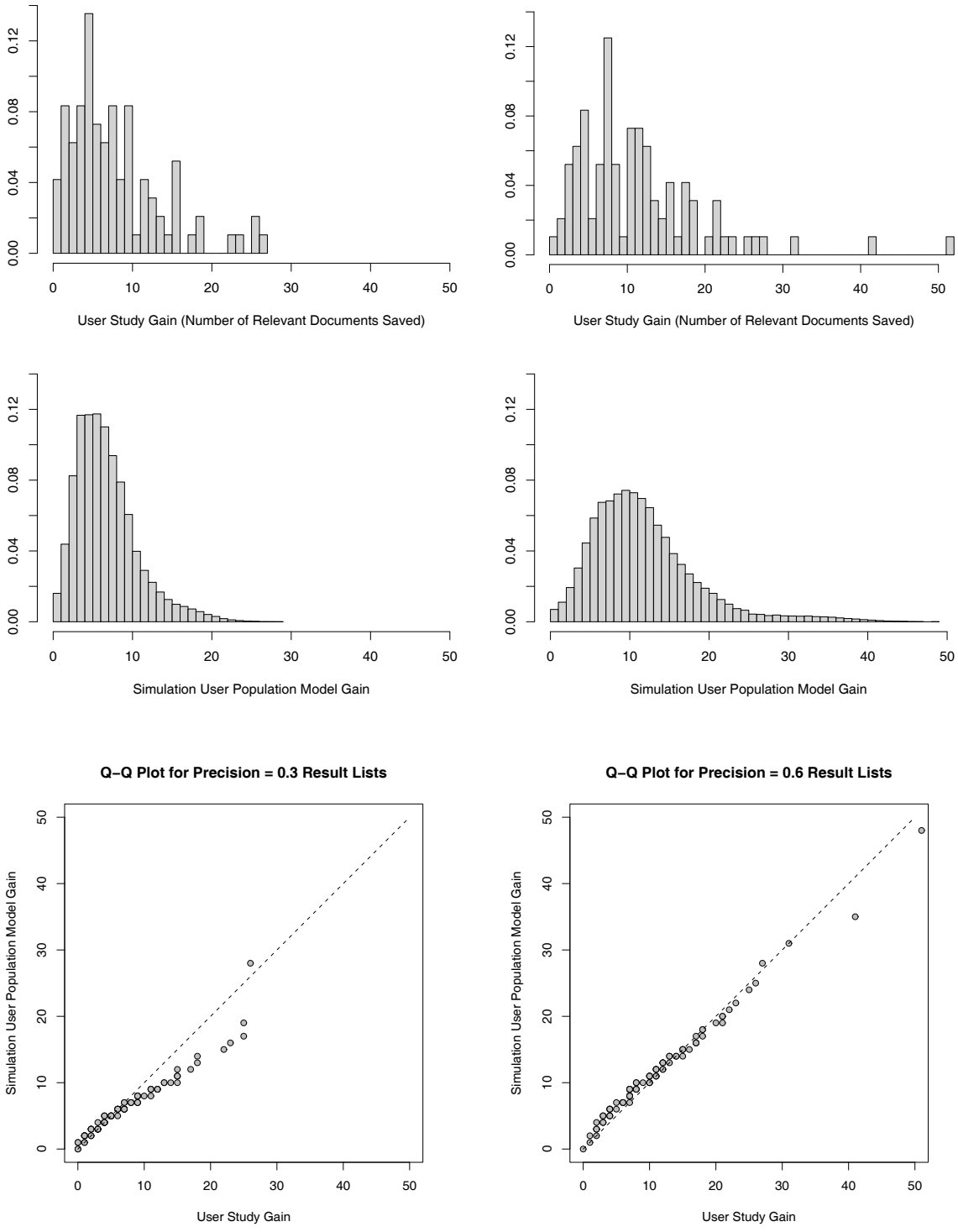

Figure 5. This figure compares the user study gain at 10 minutes with that predicted by the stochastic simulation. The user study has result lists of 0.3 and 0.6 precision. The results for the 0.3 precision lists is shown on the left and 0.6 on the right. The quantile-quantile (Q-Q) plots show that the simulation's predicted distribution is in excellent agreement with the user study for the 0.6 precision lists, while the simulation's distribution for the 0.3 precision lists is good but does not spread out as fast as the user study distribution.

comparison to be meaningful. The user study had 48 participants complete 4 search topics. In total, there are 192 measures of cumulative gain at 10 minutes. The study used 2 levels of precision: 0.3 and 0.6. Thus, at each level of precision we have 96 gain values. If we also broke the lists down by topic, we would only have 12 gain values for a topic and precision pairing, which is not enough to make a good comparison between predicted and actual distributions. Thus, we restrict our comparison to the distributions of gain values at the two levels of precision at 10 minutes.

To produce the predicted distributions, we took the 8 results lists at each precision and ran the simulation with $B = 10000$ samples per result list to produce 10000 samples of the cumulative gain at 10 minutes. Each saved relevant document adds 1.0 to the cumulative gain.

Figure 5 shows the results for the 0.3 precision lists on the left and the 0.6 precision lists on the right. The topmost plots in Figure 5 are the actual user study distributions of the number of saved relevant documents. The middle plots are the predicted distributions. To compare the distributions, we will look at the means of the distributions as well as the shape of the distributions.

The user study had mean cumulative gains at 10 minutes for the 0.3 and 0.6 lists of 7.4 and 10.9 respectively. The predicted means and standard errors were 6.22 ± 0.01 and 11.28 ± 0.02 for the 0.3 and 0.6 precision lists, respectively. While the simulation's estimate of gain falls below and above the actual user study amounts for the 0.3 and 0.6 precision lists respectively, there is considerable variation in the number of relevant documents saved by participants. The 95% confidence interval for the user study's mean cumulative gain is 6.1-8.6 for the 0.3 precision lists and 9.2-12.6 for the 0.6 precision. Thus, for both precisions, the differences between the user study's means and the simulation's predicted means are not statistically significant differences.

While we can visually compare the actual and predicted distributions, a better way to compare one distribution with another is with a quantile-quantile (Q-Q) plot. When two distributions are equivalent, the points on a Q-Q plot fall on the $y = x$ line. Deviations from the $y = x$ line can inform us in the differences between the distributions.

The bottom plots in Figure 5 are the Q-Q plots comparing the user study distribution of number of relevant documents saved to the simulation's distribution. The simulation's distribution of gain for the 0.3 precision lists is very good up to about 8 documents and then begins to slowly worsen. What we see here is that the simulation's distribution does not spread itself in the tail as much as the user study's distribution. For the 0.6 precision lists, the simulation's distribution is in excellent agreement with the user study distribution.

In general, we found outliers in performance difficult to model. For example, the bottom plot in Figure 3 shows that the fit model effectively says there is zero probability of spending more than 30 seconds on a summary. The reality is that this participant spent in excess of 30 seconds on a summary 3 times during the user study. This was a real person, and real people produce outliers in performance. We believe that modeling outliers is important and their effective modeling will improve the predicted distributions of gain.

EFFECT SIZE OF PERFORMANCE DIFFERENCES

In this section, we look at the value of having a retrieval metric that produces both an estimate of expected performance in human terms (number of relevant documents saved) and the variance of this estimate on a per-topic level.

The TREC 2005 Robust Track overview [26] compares two top performing title-only, automatic runs: uic0501 and indri05RdmmT. The geometric mean average precision (gMAP) is 0.233 for uic0501 and is 0.206 for indri05RdmmT. The MAP (arithmetic mean) is 0.310 for uic501 and 0.332 for indri05RdmmT. Both of these runs are also top performing runs when evaluated with TBG. The mean TBG for uic0501 is 4.98 and the mean TBG for indri05RdmmT is 4.70. We used 10,000 samples per topic to estimate TBG and its variance. Of note, we think that reporting an expected 4.98 relevant documents to be saved by users is a much more meaningful metric than a 0.310 MAP. TBG produces a measure of retrieval effectiveness in human terms.

Figure 6 shows each run's per topic TBG for a few example topics. For each topic, the runs' performances are shown paired with the uic0501 run always on the left of the pair and the indri05RdmmT run on the right. This plot is based on 1000 samples per topic to reduce the clutter of outliers in the plot, but otherwise the plot remains nearly the same as for the 10000 samples data.

When analyzing the performance difference between two runs, we now can better understand the significance of a difference at the topic level. For example, it is clear that on topic 433, uic0501 is significantly better than indri05RdmmT while we see that the difference on topic 436 is likely not significant. We can formally capture this idea that some differences are more important than others using measures of *effect size* [11].

There are two measures of effect size that we will demonstrate. The first measure of effect size is called Cohen's d and the second is known as the probability of superiority. To illustrate these measures of effect size, we'll use these two runs' performance on topics 325 and 303. On both topics, indri05RdmmT has a higher mean TBG than uic0501, but the distribution of TBG values overlap to varying degrees.

Cohen's d is a standardized measure of the difference between means. The measure is also known as Hedges' g [11] or the standardized mean gain [16]. The measure is defined to be:

$$d = \frac{\overline{Y}_A - \overline{Y}_B}{s_p} \tag{6}$$

where $\overline{Y}_X$ is the mean of system X, and s_p is the pooled standard deviation of systems A and B:

$$s_p = \sqrt{\frac{(n_A - 1)s_A^2 + (n_B - 1)s_B^2}{(n_A - 1) + (n_B - 1)}} \tag{7}$$

where n_X is the number of samples of system X, and s_X is the standard deviation of system X [16].

The idea of the standardized mean difference is to represent the difference between two means in terms of the spread of the data that form the means. For example, on topic 325, there is a 2.3 difference in TBG between the indri run and the uic run, and $d = 0.53$. In other words, on topic 325, the indri run's mean is 0.53 standard deviations better than the uic run's mean. In comparison, for topic 303, there is a 2.1 difference in TBG between indri and uic, and $d = 1.2$. By Cohen's d, the 2.1 difference in TBG on topic 303 is a larger effect than the 2.3 difference on topic 325. Cohen gave the guideline that a $d \leq 0.2$ is a small difference, a $d = 0.5$ is medium, and $d \geq 0.8$ is large [16].

A possible issue with Cohen's d is that the distributions of TBG are not normal but are more log-normal in their shape. Thus, one solution to this issue would be to measure the difference in the log means, rather than the means, except that some notion of smoothing would be needed to be applied to avoid taking the log of zero. Another solution would be to use a non-parametric measure of effect size.

The probability of superiority (PS) is a non-parametric measure of effect size [11]. In terms of two IR systems' performance on a topic, PS is defined to be the probability that users of system A have greater performance than users of system B, i.e.

$$PS = P(Y_A > Y_B). \tag{8}$$

In other words, PS is the probability that a randomly chosen user of system A has a greater performance than a randomly chosen user of system B. The PS can be naively computed by taking all pairs of scores for systems A and B on a topic (the cross product) and counting the number of pairs that have a higher score for A and then dividing by the total number of pairs. A tied pair is counted as 0.5 rather than 1. A more efficient means to compute the PS is to compute the Mann-Whitney U statistic, which is also known as the Wilcoxon rank sum statistic, and divide by the number of pairs. PS ranges from 0 to 1 and a PS of 0.5 corresponds to no effect. A PS of 0 or 1 both mean that the two groups of scores are completely separate. Note that $P(Y_A > Y_B) = 1 - P(Y_B > Y_A)$. PS is equivalent to the area under the curve (AUC) if we treat the scores as the output of a classifier and label one system's scores the positive instances and the other system's scores the negative instances. The PS can also be converted to an odds ratio (OR) [11]: $OR = P(Y_A > Y_B)/P(Y_B > Y_A)$ For example, a PS of 0.75 can be understand as an odds ratio of 3 to 1. In other words, for every three users that perform better with system A, one user performs better with system B.

For topic 325, there is a 2.3 difference in TBG, and the PS is 0.66 (OR $= 1.9 \approx 2$ to 1). In comparison, for topic 303, there is a 2.1 difference in TBG, and the PS is 0.81 (OR $= 4.3 \approx$ 4 to 1). The lower TBG values on topic 303 mean that the variance is less as well. In terms of user experience, the 2.1 difference on topic 303 is a much more important difference than the 2.3 difference on topic 325, for a much larger portion

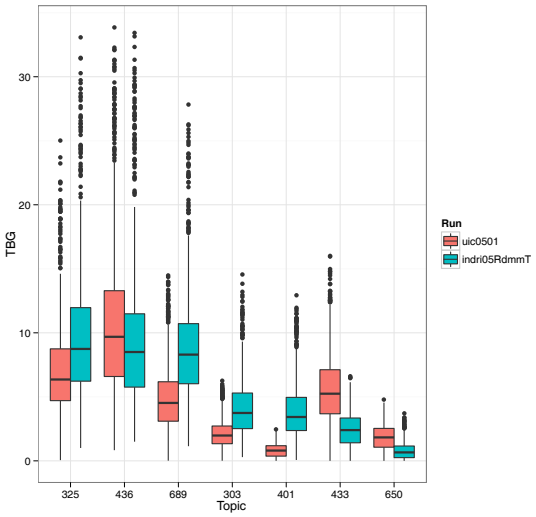

Figure 6. TBG on example topics for runs uic0501 and indri05RdmmT. For each topic, the runs are shown as a pair. The uic0501 run is always on the left of the pair and the indri05RdmmT run on the right.

of the user population will experience a difference between the two systems.

As pointed out by Cormack and Lynam [8], with a distribution of effectiveness scores on a per-topic basis, as produced by TBG, we can treat each topic as a separate experiment and then report comparisons between runs using meta-analysis [16]. We leave for future work the investigation of the application of meta-analysis to system performance comparison.

RELATED WORK

Others have proposed models of user behavior for evaluating search engines that incorporate notions of user effort and time. Dunlop [10] directly inspired our own approach, both in this paper and in prior work [23, 24]. He defined a *time-to-view graph* as a function indicating the time required for a user for to view a given number of relevant documents — essentially the inverse of $G(t)$. For computing time-to-view graphs, Dunlop developed and calibrated a model of user behavior that includes estimates for the times needed to load screens, read summaries, view documents, etc., on an actual user interface.

Zhang et al. [32] analyzed logs from a commercial search engine to compute parameters for cumulated gain effectiveness measures. They created a user model from query and click logs and applied it to compute empirical discount values. They compared these empirical values to the discount functions underlying traditional measures, concluding that rank biased precision's (RBP) [17] geometric distribution provides the best fit.

Yilmaz et al. [31] present a model of a user interacting with a standard commercial search engine, applying it to compute *expected browsing utility* (EBU), which is similar to the expected total gain computed by TBG. Like Zhang et al., they calibrate and validate their model from the queries and clicks appearing in the logs of a commercial search engine. EBU

incorporates probabilities for clicking on summaries, returning to the result page after viewing a document, and stopping after viewing relevant and non-relevant documents. However, EBU does not model the time required for these actions, implicitly adopting the fixed-rate traversal assumption.

Carterette et al. [5] present a method for estimating the parameter in RBP's geometric distribution from a user's queries and clicks. By estimating parameters for a large number of users, they create a distribution of parameter values, where different query types (e.g, informational vs. navigational) may have different distributions of parameter values. Sampling from the user population implied by the distribution of parameters values allows them to compute variance due to user behavior. We follow a similar approach in the current paper, sampling from a user population and simulating user behavior over a result list, but with a more substantial user model.

Cormack and Lyman [8] have investigated how variation in the collection affects IR evaluation. While we have added user variance to time-biased gain, we do not address collection variation in this paper. Cormack and Lyman's work is also notable for the points it makes with regard to the need to measure the magnitude and importance of differences and not merely to report the statistical significance of differences. Our work here on incorporating more variance into IR evaluation, fits within the framework outlined by Cormack and Lynam for the meta-analysis of IR systems.

While rarely making time-based predictions, others have aimed to simulate the use of interactive IR, or have applied HCI user-modeling techniques to IR-related tasks [2–4, 6, 14, 15, 18, 30]. In some cases, simulations are compared to human studies to determine if the user model accurately reflects human performance.

CONCLUDING DISCUSSION

User studies allow us to measure the impact of specific design and interface choices on user experience. System-oriented tests, as typified by Cranfield-style evaluation, allow for low cost, repeatable evaluation, but lack the realism of user studies. Time-biased gain is a Cranfield-style evaluation metric that tries to capture benefits of both user-oriented studies and system-oriented tests. For example, time-biased gain measures effectiveness in human terms — the number of saved relevant documents — rather than difficult to interpret scores that are not directly predictive of human performance. Time-biased gain is able to make predictions of human performance behavior because it employs a user model of result list processing that has been calibrated and validated using actual user behavior data.

The stochastic simulation of time-biased gain models behavior in terms of the distribution of times to complete actions and the probabilistic nature of decisions made during the processing of a ranked list. The simulation's user model operates in the context of a hypothetical user interface consisting of document surrogates that when clicked on allow the user to view the full documents. The simulation gives us a model of result list processing behavior that we can apply to other result lists and obtain estimates of both the gain and variance where the gain is measured in units of number of relevant documents saved. In other words, we can "replay" the conducted user study over and over again with new ranked lists to evaluate performance.

The simulation in this paper is unique in its use of multiple user models to simulate a population. An advantage of using multiple models over one single complex model is that it gives us greater experimental flexibility. We can study performance in terms of the population and also in terms of specific users or classes of users. Our user model and validation is limited by the amount of data we have collected, but the idea of creating multiple models to simulate a population should be easy to apply to larger datasets, e.g. those datasets collected by commercial search engines.

By simulating a user population, we can model user variance in performance on a per-topic basis. For each search topic, time-biased gain produces a distribution of the number of relevant documents saved (cumulative gain). We showed that the simulation's fit to the user study's distribution of cumulative gain is good.

Knowing the per-topic distribution of gain allows for measurement of the effect size of differences. Most existing metrics ignore user variance and only produce a single number for the quality of a ranked list. Metrics that produce an estimate with a variance of zero, overstate the effect size of the difference between two ranked lists. Real users are variable and as such some will perform better with one list than the other and vice versa. Time-biased gain allows for the use of effect size measures on a per topic basis so that we can determine, for example, the probability that a random user of one ranked list will save more relevant documents than a random user of another ranked list. As such, time-biased gain allows for a prediction of the impact that retrieval improvements have on actual user performance. In short, we can start to have conversations about the degree to which measured improvements in retrieval quality result in noticeable changes to the user experience.

ACKNOWLEDGMENTS

We thank the reviewers for their helpful feedback. This work was supported in part by NSERC, in part by the GRAND NCE, in part by Google, in part by Amazon, in part by the facilities of SHARCNET, and in part by the University of Waterloo.

REFERENCES

1. Aula, A., Majaranta, P., and Räihä, K.-J. Eye-tracking reveals the personal styles for search result evaluation. In *Human-Computer Interaction – INTERACT 2005*, vol. 3585 of *LNCS*, Springer (2005), 1058–1061.
2. Azzopardi, L. The economics in interactive information retrieval. In *SIGIR*, (2011), 15–24.
3. Azzopardi, L., Järvelin, K., Kamps, J., and Smucker, M. D. Report on the SIGIR 2010 workshop on the simulation of interaction. *SIGIR Forum*, (January 2011), 35–47.

4. Baeza-Yates, R., Hurtado, C., Mendoza, M., and Dupret, G. Modeling user search behavior. In *Proceedings of the Third Latin American Web Conference*, IEEE (2005), 242-251.
5. Carterette, B., Kanoulas, E., and Yilmaz, E. Simulating simple user behavior for system effectiveness evaluation. In *CIKM*, (2011), 611–620.
6. Chi, E. H., Pirolli, P., Chen, K., and Pitkow, J. Using information scent to model user information needs and actions and the web. In *SIGCHI*, (2001), 490–497.
7. Clarke, C. L., Craswell, N., Soboroff, I., and Ashkan, A. A comparative analysis of cascade measures for novelty and diversity. In *WSDM*, (2011), 75–84.
8. Cormack, G. V., and Lynam, T. R. Statistical precision of information retrieval evaluation. In *SIGIR*, (2006), 533–540.
9. Dumais, S. T., Buscher, G., and Cutrell, E. Individual differences in gaze patterns for web search. In *IIiX*, (2010), 185–194.
10. Dunlop, M. D. Time, relevance and interaction modelling for information retrieval. In *SIGIR*, (1997), 206–213.
11. Grissom, R. J., and Kim, J. J. *Effect Sizes for Research*, 2nd ed. Routledge, Taylor and Francis Group, 2012.
12. Hersh, W., Turpin, A., Price, S., Chan, B., Kramer, D., Sacherek, L., and Olson, D. Do batch and user evaluations give the same results? In *SIGIR*, (2000), 17–24.
13. Järvelin, K., and Kekäläinen, J. Cumulated gain-based evaluation of IR techniques. *TOIS*, (2002), 20(4):422–446.
14. Keskustalo, H., Järvelin, K., Sharma, T., and Nielsen, M. L. Test collection-based IR evaluation needs extension toward sessions: A case of extremely short queries. In *AIRS*, (2009), 63–74.
15. Lin, J., and Smucker, M. D. How do users find things with PubMed? Towards automatic utility evaluation with user simulations. In *SIGIR*, (2008), 19–26.
16. Lipsey, M. W., and Wilson, D. B. *Practical Meta-Analysis*. Sage Publications, Inc., 2001.
17. Moffat, A., and Zobel, J. Rank-biased precision for measurement of retrieval effectiveness. *TOIS*, (2008), 27(1):1–27.
18. O'Brien, M., Keane, M. T., and Smyth, B. Predictive modeling of first-click behavior in web-search. In *WWW*, (2006), 1031–1032.
19. Pavlu, V., Rajput, S., Golbus, P. B., and Aslam, J. A. IR system evaluation using nugget-based test collections. In *WSDM*, (2012), 393–402.
20. Robertson, S. A new interpretation of average precision. In *SIGIR*, (2008), 689–690.
21. Smith, C. L., and Kantor, P. B. User adaptation: good results from poor systems. In *SIGIR*, (2008), 147–154.
22. Smucker, M. D. An analysis of user strategies for examining and processing ranked lists of documents. In *HCIR*, (2011).
23. Smucker, M. D., and Clarke, C. L. A. Stochastic simulation of time-biased gain. To appear in *CIKM*, (2012), 5 pages.
24. Smucker, M. D., and Clarke, C. L. A. Time-based calibration of effectiveness measures. In *SIGIR*, (2012), 95–104.
25. Smucker, M. D., and Jethani, C. Human performance and retrieval precision revisited. In *SIGIR*, (2010), 595–602.
26. Voorhees, E. M. Overview of the TREC 2005 Robust Retrieval Track. In *TREC*, (2005).
27. Voorhees, E. M. I come not to bury Cranfield, but to praise it. In *HCIR*, (2009), 13–16.
28. Voorhees, E. M., and Harman, D. K., Eds. *TREC*. MIT Press, 2005.
29. Weiss, E. N., Cohen, M. A., and Hershey, J. C. An iterative estimation and validation procedure for specification of semi-Markov models with application to hospital patient flow. *Operations Research*, (1982), pp. 1082–1104.
30. White, R. W., Ruthven, I., Jose, J. M., and van Rijsbergen, C. J. Evaluating implicit feedback models using searcher simulations. *TOIS*, (2005), 23(3):325–361.
31. Yilmaz, E., Shokouhi, M., Craswell, N., and Robertson, S. Expected browsing utility for web search evaluation (2010). In *CIKM*, (2010), 1561–1564.
32. Zhang, Y., Park, L. A., and Moffat, A. Click-based evidence for decaying weight distributions in search effectiveness metrics. *Information Retrieval* (2010), 13:46–69.

APPENDIX A

To get Equation 1 into the form of Equation 2, our first step is to go from a probability density function $f(t)$ to a survival probability, $D(t)$, where $D(t)$ is the fraction of the population that survives to time t. To do this, we let $F(t)$ be the cumulative distribution function for $f(t)$, i.e. $f(t) = F'(t)$. The survival probability, $D(t) = 1 - F(t)$, and thus $f(t) = -D'(t)$. As such, Equation 1 can be restated as:

$$\begin{aligned} E[G(t)] &= -\int_0^\infty G(t)D'(t)dt \\ &= -G(t)D(t)\Big|_0^\infty + \int_0^\infty G'(t)D(t)dt \\ &= \int_0^\infty G'(t)D(t)dt \end{aligned} \tag{9}$$

Measuring G' is difficult, and instead we make gain discrete and be obtained at the point in time when the user has finished processing the document at rank k, i.e. at $T(k)$, and we write Equation 9 as Equation 2.