Effective User Interaction for High-Recall Retrieval: Less is More

Haotian Zhang, Mustafa Abualsaud, Nimesh Ghelani, Mark D. Smucker, Gordon V. Cormack, and Maura R. Grossman
University of Waterloo, Ontario, Canada
{haotian.zhang,m2abuals,nghelani,mark.smucker,gvcormac,maura.grossman}@uwaterloo.ca

ABSTRACT

High-recall retrieval — finding all or nearly all relevant documents — is critical to applications such as electronic discovery, systematic review, and the construction of test collections for information retrieval tasks. The effectiveness of current methods for high-recall information retrieval is limited by their reliance on human input, either to generate queries, or to assess the relevance of documents. Past research has shown that humans can assess the relevance of documents faster and with little loss in accuracy by judging shorter document surrogates, e.g. extractive summaries, in place of full documents. To test the hypothesis that short document surrogates can reduce assessment time and effort for high-recall retrieval, we conducted a 50-person, controlled, user study. We designed a high-recall retrieval system using continuous active learning (CAL) that could display either full documents or short document excerpts for relevance assessment. In addition, we tested the value of integrating a search engine with CAL. In the experiment, we asked participants to try to find as many relevant documents as possible within one hour. We observed that our study participants were able to find significantly more relevant documents when they used the system with document excerpts as opposed to full documents. We also found that allowing participants to compose and execute their own search queries did not improve their ability to find relevant documents and, by some measures, impaired performance. These results suggest that for high-recall systems to maximize performance, system designers should think carefully about the amount and nature of user interaction incorporated into the system.

CCS CONCEPTS

• **Information systems → Information retrieval**;

KEYWORDS

High-Recall; Electronic Discovery; Systematic Review

ACM Reference Format:
Haotian Zhang, Mustafa Abualsaud, Nimesh Ghelani, Mark D. Smucker, Gordon V. Cormack, and Maura R. Grossman. 2018. Effective User Interaction for High-Recall Retrieval: Less is More. In *The 27th ACM International Conference on Information and Knowledge Management (CIKM '18), October 22–26, 2018, Torino, Italy.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3269206.3271796

1 INTRODUCTION

High-recall information retrieval (HRIR) is integral to many tasks that require the finding of all, or nearly all, relevant documents in a collection. Example applications of HRIR include electronic discovery ("eDiscovery"), systematic review, and the construction of information retrieval (IR) test collections. In the legal discovery process, both parties in a case are required to find substantially all relevant material from their own document collections and provide it to the opposing party. An important part of evidence-based medicine is the systematic review of all relevant research studies. Finally, if an IR test collection knows all relevant documents for a given search topic, we can confidently measure the quality of search results using the effectiveness measure of our choice.

In all of these applications of high-recall retrieval, human interaction is required to find relevant documents. Users of high-recall systems are often required to provide a large number of relevance assessments. Some systems may also engage the user in the search by allowing them to interactively search for relevant documents. The amount of human interaction required to achieve high-recall can be large and thus expensive.

For legal discovery, traditionally each document in a collection would be reviewed by an attorney and take a few minutes per document (1-4) [24]. With the digitization of information, the sizes of collections to search have grown rapidly. One eDiscovery firm reports that their usual case has between 600,000 and 1 million documents to review [32]. Thus eDiscovery is now concerned with finding all relevant documents without requiring human review of every document, but at a minimum, a human is required to examine the final set of documents before being declared relevant.

The process of building IR test collections has long used various processes to avoid judging all documents in the collection, but even so, the amount of human judging required can be large. For example, the TREC Legal Track [6] ran from 2006 to 2011 and developed two reusable test collections involving human assessments. Baron et al. [6] reported the average assessment rate was 24.7 documents per hour for different topics at the TREC 2006 Legal Track. At the TREC 2008 Legal Track [23], assessors reported that it took a collective 631.2 hours to review 13,543 documents, or a rate of 21.5 documents per hour.

To reduce the cost of achieving high-recall, factors to consider include the total number of relevance assessments required, the time spent per assessment, the hourly pay rate for assessors, and the quality of the assessor. In this paper, we examine the effect of using short document excerpts in place of full documents for the assessment of relevance as part of a high-recall retrieval system

CIKM '18, October 22–26, 2018, Torino, Italy

ACM ISBN 978-1-4503-6014-2/18/10…$15.00
https://doi.org/10.1145/3269206.3271796

based on continuous active learning (CAL). We hypothesize that for a given amount of time, users will be able to find a greater number of relevant documents if they judge short excerpts in place of full documents. In simulation studies [39], we have found that even a single extracted sentence may be adequate for CAL to perform well. In addition, we examine the effect of allowing users to interactively search. In cases where the CAL system has trouble finding relevant documents, interactive search may help users find relevant documents that can then be used by the CAL algorithm to find other relevant documents.

We base our experiment's high-recall system on Cormack and Grossman's state-of-the-art autonomous technology-assisted review (AutoTAR) method [10]. AutoTAR is version of Continuous Active Learning (CAL) [9], which is an iterative relevance feedback process. A CAL system provides a document to a human assessor for judging and based on the judgment, the system uses machine learning to learn a new model of relevance and then selects the next most likely relevant document to present to the assessor for judging. AutoTAR is autonomous in that it has no parameters to tune, it starts with a single seed relevant document or query, and only requires a user to provide relevance judgments for each document AutoTAR returns in sequence.

The TREC 2015 and 2016 Total Recall tracks [14, 26] focused on the problem of high-recall retrieval. The Total Recall track organizers provided a version of AutoTAR called the baseline model implementation (BMI) to participants, and used BMI as the track's baseline method. To the organizers' surprise, after running the track for two years, no other method could consistently outperform BMI [15]. This result is all the more amazing because many of the competing methods made use of manual searches for relevant documents, which prior to the running of the track, was assumed by many to be a sure-fire method to achieve high recall.

Our experimental system was configurable to provide a version of CAL that only showed a paragraph-length document excerpt for judging and a version of CAL that showed the excerpt and allowed the user to click to view the full document. In addition, our system could be configured to only allow users to provide judgments to CAL or could be configured to also allow users to use a search engine to find relevant documents.

As designed, our CAL system had 4 variations to support the 2×2 factorial experiment. One factor was the display of the document in the CAL component: an excerpt alone or an excerpt plus the ability to click and view the full document. The other factor was whether or not a search engine was available to users. Any relevance judgments made with the search engine were then available to the CAL component's machine learned model. While BMI has been very successful, the proposition that a CAL system alone could match or do better than a system that combines CAL and manual searching has not been tested in a controlled fashion prior to this paper, to the best of our knowledge.

We had 50 participants use the system, in each of its variations, for one hour to find as many relevant documents as possible for a given search topic. We used the search topics and documents from the TREC 2017 Common Core Track [3]. To evaluate performance, we used several measures to reflect the different needs of different high-recall applications. We found that:

- For our primary measures of performance, CAL with paragraph excerpts outperformed the version of CAL that allowed users to view full documents.
- Allowing users to interactively query a search engine, did not help them find relevant documents, and for some measures, hurt performance.
- Any value of being able to view full documents or interactively search for relevant documents was offset by the significant time cost of using these interactive features.

In the remainder of the paper, we review related work, detail our experiment, present and discuss our results, and then conclude the paper.

2 BACKGROUND AND RELATED WORK

High-recall information retrieval has been a long standing problem in the IR field. In this section we highlight the most related work and provide background on the high-recall methods we used.

In IR, the builders of test collections have strived to find all relevant documents for the collections' search topics. The traditional approach to this problem has had multiple groups perform retrievals for the topics and submit these retrieval runs to the collection builders. With the received runs, a pooling of the top k documents from each run is performed and then assessors judge whether or not each document in the pool is relevant [35].

Cormack et al. [13] showed that a process of interactive searching and judging (ISJ) could be used to construct a set of relevance judgments as good as those produced by the pooling method but at lower cost in terms of number of documents judged. Using ISJ, less than a quarter of the number of documents needed to be judged compared to the pooling method. Further investigations confirmed the potential and effectiveness of ISJ [27, 30].

There have also been methods created to more effectively use the runs submitted for test collection construction. In the same paper, Cormack et al. presented an algorithm, move-to-front (MTF), that would order the documents to judge based on the quality of the runs, and like ISJ, it also found relevant documents with fewer judgments than pooling. In another approach, Aslam et al. [5] showed that their Hedge algorithm could learn online which runs were better than others as assessors made judgments and significantly reduce the number of judgments needed to find relevant documents. In effect, Cormack et al. with MTF and Aslam et al. with the Hedge algorithm, showed early on that online machine learning approaches could be used to find relevant documents better than simple pooling.

Both the TREC and CLEF conferences have conducted tracks that have focused on high-recall retrieval. At TREC, both the Legal (2006-2012) and Total Recall (2015-2016) tracks developed collections and methods to evaluate the performance of systems designed to find all relevant documents. At CLEF, the 2017 eHealth lab had a task focused on systematic reviews for medicine [18].

Cormack and Mojdeh [12] used a combination of ISJ and machine learning in the TREC 2009 Legal track [17] and achieved the highest recall, precision, and F_1 scores. Cormack and Grossman [10] showed that their continuous active learning (CAL) AutoTAR algorithm could outperform ISJ on the same TREC 6 task of test collection construction without any human interaction other than judging the relevance of documents. As noted in the introduction, at the conclusion of the Total Recall tracks, the BMI version of AutoTAR was not consistently beat by any other method [7, 15, 40].

In the systematic review task of the CLEF 2017 eHealth lab [18], Cormack and Grossman [11] used BMI to achieve the highest recall at a given amount of shown documents. The second ranked team, Anagnostou et al. [4] applied a similar approach as CAL, but they incorporated learning-to-rank with relevance feedback in an active learning process.

Miwa et al. [21] have used active learning for systematic review in the public health and social science domain. Their method involves manual curation of an initial training set from a random set of citations. This is followed by manual assessments of classifier output which is used to train the classifier. This step is repeated until the stopping condition is satisfied.

The TIPSTER Text Summarization Evaluation (SUMMAC) [19] found that short summaries of documents could reduce relevance assessment time by 40-43%. These summaries were between 10-17% of the length of the original document. No statistically significant difference was found between judgment accuracy with summaries and with full documents. Smucker and Jethani [29] reported that summaries of news articles with no more than 50 words (approximately 2 sentences or less) were judged in approximately 16 seconds while full documents took 49 seconds. Smucker and Jethani also reported that the accuracy of relevance judging these short summaries ranged from 62-72% depending on conditions and that accuracy of judging full document was 75-76%.

3 METHODS AND MATERIALS

In this section, we describe our experiment in detail. We next describe the search topics and document collection, the study design, the high-recall system and its implementation, and other details of the experiment including how we measured performance and determined statistical significance.

3.1 Search Topics and Documents

We used the TREC 2017 Common Core Track [3] test collection for our search topics and documents. We used the 50 NIST assessed topics as opposed to the full set of 250 topics. The track's task was adhoc retrieval of documents from the New York Times dataset [28], which includes over 1.8 million news articles.

Submitted runs were either manual or automatic runs. An automatic run involves no manual interaction or tuning of the system based on the topics. Any run that is not an automatic run, is a manual run.

The topics were copied and updated from the TREC 2004 Robust Track [36]. This allowed teams to train models based on the existing relevant assessments ("qrels") for these topics. Thus, both manual and automatic runs are further classified by whether or not the runs made use of these existing qrels.

We ourselves participated in the Common Core track [38]. We used an early variant of our high-recall system to find relevant documents and submitted several manual runs based on this effort. Based on our experience with this system, we revised it for the experiment that we report in this paper. Using only the first 10 study participants, we submitted preliminary results from our experiment to the track as a run. By participating in the track, we hoped to increase the chance that the documents that our participants reported as relevant would also be judged by the NIST assessors as part of the construction of the test collection. To reduce issues of bias, we were careful to make sure the 10 study participants used each of the system variations and also covered all 50 search topics. We finished running the experiment with the remaining 40 participants after the track submission deadline, and thus these judgments were not used to create any of our submitted runs. We are careful to exclude our runs and related runs from our university when it matters to the analysis of experiment results.

Table 1: The 2×2 factorial design and our shorthand designations for each treatment.

Search Available	CAL types	
	Full document Available	Paragraph Excerpt only
No	CAL-D	CAL-P
Yes	CAL-D&Search	CAL-P&Search

When we displayed search topics to participants, we used hand edited versions that combined the topic's description and narrative. We did this both to better clarify the topic and to shorten the amount of material displayed to the participant. Regardless of the retrieval system variation used by the participant, the participants all saw the same topic descriptions.

3.2 Study Design

We had participants use four different variations of a high-recall retrieval system to find as many relevant documents as possible within 1 hour. All of the system variations incorporated a continuous active learning (CAL) system derived from the TREC Total Recall Track's baseline model implementation (BMI). We designed our experiment to investigate two factors, each of which had two levels, i.e. a 2×2 factorial design. The first factor determined whether participants using the CAL portion of the system would judge a paragraph-length excerpt of a document or be shown the excerpt and also be able to click to view the full documents. The other factor determined whether or not a search engine was made available to the participants. Judgments made from the search engine became part of the training set that the CAL system, which was always available, could use to learn its model of relevance.

Table 1 summarizes the 2×2 factorial design. Throughout the rest of the paper, we will refer to each of the treatments by their shorthand: **CAL-P** (CAL with paragraphs and no search), **CAL-D** (CAL with full documents), **CAL-P&Search** (CAL with paragraphs and search), and **CAL-D&Search** (CAL with full documents and search).

Each participant completed 5 tasks, and each of a participant's tasks was associated with a unique search topic. For each of 4 tasks, the participant used one of the system variations as per Table 1 to find relevant documents for one hour. For the fifth task, the participant judged the relevance of 60 documents randomly selected based on their likelihood of being relevant as determined by a model trained on our own judgments. We call this the *reference* treatment. For this treatment, we showed participants the full document and the participants had to judge the document's relevance to proceed to the next document. There was no time limit for the reference treatment. We intend in future work to use this treatment to compare user behavior on a traditional relevance judging task to user behavior on the other four treatments. By the end of the experiment, each system variation had been applied once to each of the 50 topics.

We created a balanced study design as follows. We first divided the 50 topics into 10 groups of 5 topics each. For each group of 5 topics, we created a 5×5 Graeco-Latin square. The rows of the square were users and the columns were task numbers. The five topics and five treatments were assigned to cells of the squares, and then the squares were randomized. After running the experiment, we discovered that the topics were not randomly divided into groups of 5 but were instead assigned to groups in their numeric order. While there is perhaps some association across topics given their number, we do not think this lack of randomization is a cause for concern.

3.3 High-Recall Retrieval System

At the core of our high-recall retrieval system is an implementation of continuous active learning (CAL). As mentioned in the introduction, CAL is an iterative relevance feedback method whereby the user judges one document after another as selected by CAL. CAL selects documents for judging based on its learned model of what is and is not relevant.

The CAL system has two possible configurations. In the first configuration, CAL shows a document's title, date, a selected paragraph from the document, and allows the user to click to view the entire document. Clicking to view the full document displays it below the paragraph excerpt. In the second configuration, CAL is the same as the first but does not provide a means for the user to view the full document. Once the user has decided on the document's relevance, the user can then use the buttons on the right hand side to submit the judgment. After receiving the judgment, CAL selects and shows the user the next unjudged document. In our implementation of CAL, we select documents by ranking all paragraphs in the collection and selecting the paragraph most likely to be relevant from the set of unjudged documents. These two configurations and corresponding CAL user interfaces are detailed in paper [1].

As we detail in Section 3.3.1, we carefully engineered CAL so that there was no noticeable delay from submitting a judgment to receiving the next document to judge.

Our system provides a 3-level relevance scale: non-relevant, relevant, and highly-relevant. As noted by Harman [16, Section 2.4.3, page 39], NIST assessors reportedly prefer a three level judgment scale over binary decisions because it makes decision making easier. We treat both relevant and highly-relevant judgments as relevant in our analyses. Keyboard shortcuts are available for judging in addition to the buttons. A keyword highlight feature is provided so that user can use the *"Ctrl + F"* shortcut and enter keywords to highlight them. Multiple keywords separated by spaces can be entered to highlight each keyword simultaneously.

Algorithm 1: Paragraph Level Continuous Active Learning

```
Step 1. Treat the topic statement as a relevant document and
add this document into the training set;
Step 2. Temporarily augment the training set with 100 random
documents from the corpus, assuming their label as
"non-relevant" ;
Step 3. Train a logistic regression classifier using the training
set;
Step 4. Discard the 100 random documents added in Step 2
from the training set;
Step 5. Score all the paragraphs from all unjudged documents
using the newly trained classifier;
Step 6. Present the highest-scoring paragraph $p$ for assessment,
and record the judgment as the label for paragraph's
corresponding document $d$;
Step 7. Add the labeled document $d$ to the training set;
Step 8. Repeat steps 2 through 7 until some stopping criteria is
satisfied.
```

Should a user make a mistake and want to change a CAL judgment, the interface provides means for the users to view and modify their previous 10 CAL judgments.

Our system [1] can operate with CAL alone or can also provide the user with access to a search engine. The interface and functionality of search engine are also detailed in paper [1]. The user can query and select the number of documents to return (10, 20, 50, or 100) with 10 results being the default. Users can judge the relevance of a document from the search engine results page (SERP). Any judged document shows the user's judgment in the SERP, and the user can change the judgment if so desired. Users can also click on each result to view the full document content. When viewing a full document, users are provided with the same judging interface and keyword highlighting tool as in the CAL interface. Users have the freedom to choose what documents to judge or not. Users can specify phrases in their queries with double quotes and can require the presence of a word with a plus sign.

When the system includes search, users can switch between CAL and Search at any time using two buttons on the left hand side of the interface.

Judgments collected from the search interface go into the same set of judgments created in the CAL interface. Thus, CAL will train based off of all judgments and will not show already judged documents to the user. While search may be useful on its own for finding documents, it might also help CAL for those times when it seems to fail and requires a new relevant seed document [10].

We next detail the implementations of CAL and search.

3.3.1 *Implementation of CAL.*

For our CAL implementation, we modified the algorithm used in the baseline model implementation (BMI) [26] to enable the rankings to work on a paragraph level instead of documents. The details of the modified algorithm are shown in Algorithm 1.

For each document, we extracted the paragraphs wrapped by the $\langle p \rangle \langle /p \rangle$ tags. In total, there are around 30 million paragraphs in the collection, with an average of 16.7 paragraphs per document.

[1] Code publicly available at https://hical.github.io/

We also extracted the document's title, date, and id for logging and displaying purposes.

BMI used the word-based *tf-idf* as document feature vectors for training the classifier. A word is considered to be any sequence of two or more alphanumeric characters not containing a digit, that occurs at least twice in the corpus. All the words in the corpus are stemmed using the Porter stemmer. We do not remove stopwords.

We calculated the $\textit{tf-idf} = (1 + \log(\textit{tf})) \cdot \textit{log}(N/\textit{df})$ weight for each word in paragraphs and documents. *tf* is the term frequency, N is the total number of documents, and *df* is the document frequency. When calculating the *tf-idf* weight for paragraphs, we use the same values of N and *df* used for documents.

For each feature vector of document d and paragraph p, we normalized the *tf-idf* weight for each word t using two different L2 normalization methods as follows:

$$\begin{cases} \textit{tf-idf}_{t\in d} = \dfrac{\textit{tf-idf}_t}{\sqrt{\sum\limits_{t\in d} \textit{tf-idf}_t^{\,2}}} \\ \textit{tf-idf}_{t\in p} = \dfrac{\textit{tf-idf}_t}{\max\{20, \sqrt{\sum\limits_{t\in p} \textit{tf-idf}_t^{\,2}}\}} \end{cases} \tag{1}$$

We used the same hyperparameters as BMI for training the logistic regression classifier in Sofia-ML[2]: *–learner-type logreg-pegasos –loop-type roc –lambda 0.0001 –iterations 200000.*

In our experiment, the topic statement mentioned in Step 1 of Algorithm 1 is the concatenation the title and description of the topic. Note that the classifier trains on the documents and scores on the paragraphs in order to select the paragraph most likely to be relevant. An assessment on a paragraph p is considered to be true for the document d it is part of.

The original BMI algorithm is written in Bash, which is suitable for simulations but inefficient for practical use. In addition to the algorithmic modifications, we reimplemented BMI in C++.

Training and scoring the entire corpus is resource intensive. The original implementation performed this step after receiving a batch of judgments whose size increased exponentially. One of the reasons for doing this was to save computation time. Our implementation is capable of efficiently training the model and re-scoring all the documents whenever a new assessment is made from the user.

The original BMI implementation suffered from the heavy reliance on file I/O and suboptimal intermediate operations. To enable fast processing and use of efficient intermediate data structures, we stored all the vector representations of paragraphs and documents in memory, and parallelized the computations across paragraphs and documents.

A key difference between Algorithm 1 and our actual implementation is the asynchronicity of steps 6 and 7 with the rest of the algorithm. Steps 3 through 5 have a user-noticeable latency which can negatively impact user experience. Instead of waiting for the assessments to be processed, we simply reuse the scores computed in Step 5 and present the next highest-scoring paragraph to the assessor. Meanwhile, training and paragraph scoring are performed in the background. Under our experiment setup, the system performed steps 3 through 5 in less than 2 seconds. Since most users take more than 2 seconds to make a judgment, they always perceive the impact of their last judgment as soon as they perform their next judgment.

3.3.2 Implementation of Search. For the search engine, we processed the LDC New York Times Annotated Corpus [28] by extracting from each document its guid, title, date, and text body. To extract these fields, we used the provided Java NYTCorpusDocumentParser class that is packaged with the collection. We then split each document into sentences using a java port of the sentence splitter [22] packaged as part of an early version the Wikipedia Miner software [20].

Using Indri [31], we indexed each document's title and body and stemmed words with the Krovetz stemmer. Retrieval uses Indri's default parameters. To build snippets, we retrieve the top 2 scoring sentences from a document, concatenate them, and then truncate them to a maximum of 75 words.

3.4 User Study Procedure

After receiving ethics approval from our university's office of research ethics, we recruited study participants using posters and emails to various student lists.

After giving their consent to participate in the study, each participant underwent an in-person tutorial covering installation of our system on their own computers and instructions on how to use various features of the system. As part of the tutorial, we instructed participants to follow Voorhees' definitions for graded relevance levels of non-relevant, relevant, and highly relevant [34]. Following usual notions of TREC relevance, we told participants that a document is relevant if any portion of it is relevant. We also told participants to strive to be consistent in their judgments and not to adjust their notion of relevance to meet any notion that more or fewer documents should be found relevant. We warned participants to not rely on keywords for making judgments.

We used 2 topics from the TREC 2004 HARD Track [2] to give participants practice making graded relevance judgments. For one topic, participants judged 6 documents and discussed with the researcher any differences between their judgments and the NIST judgments. For the other topic, the same process was followed, but in this case, only a paragraph-length excerpt was shown to the participants for each document.

When it came to effort, we asked participants to "work as fast as possible while maintaining your accuracy." We made clear to the participants that for the 4 tasks requiring 1 hour of work, nothing they would do would cause the session to end before an hour of work was completed. In particular, we told participants to not submit random judgments quickly in the hopes that the system would run out of documents to judge. Indeed, the system would not run out of documents until all 1.8 million documents in the collection had judgments.

The tutorial also included a practice task to familiarize participants with the system. Both interfaces, search and CAL, were thoroughly explained, and each participant used both during the practice task. We used one of the non-NIST judged Common Core topics for this practice.

We informed participants that during some tasks, both search and CAL would be available to utilize and they could switch between the two as they wished. We made it clear to participants

[2] https://code.google.com/archive/p/sofia-ml/

that the purpose of the study is to try to retrieve as many relevant documents as they can using the methods provided.

After the tutorial, participants then proceeded with the remainder of the study on their own. We asked participants to try and finish within 5-7 days. During the course of the study, participants could choose to take a break or continue their progress whenever they wished. We encouraged the participants to try finishing one task in one sitting and take breaks between tasks.

We had participants work on their own computers in whatever location or environment they preferred. We made this choice largely because we felt it would be too difficult to schedule six hours of work for 50 participants in a limited period of time. Any variation across the participants is random and does not effect the results because we carefully balanced the experiment design. A benefit of allowing participants to work on their own gives us a sense of how crowd-sourced workers might perform at this task.

Because participants did work on their own, we implemented our system to monitor their activity and only count their active time working. If the participant did not make any mouse movements, mouse clicks, or keyboard clicks within two minutes, the system would pop-up a dialog box and remind the participant to continue working. We did not count these inactive periods towards a participant's total time on task.

For the four tasks that required participants to work for one hour, we had participants keep working on the task until we had recorded one hour's worth of work. Unfortunately, our software allowed some participants to work in excess of one hour on some tasks. To make sure we only measured performance for one hour, we truncated user activity to one hour. As part of this truncation, we also treated any gaps between recorded events greater than 5 minutes as inactive, and we removed these gaps from the user's total time.

When participants started work on the study, they first answered a demographics questionnaire. Then they performed their 5 search tasks (see Section 3.2). Each task included a pre- and a post-task questionnaire. After completing all five tasks, participants answered an exit questionnaire to collect their feedback and overall experience. Once finished, participants returned to be paid $100 for their participation.

3.5 Participants

Before conducting the full study, we completed a pilot test with two participants to uncover any potential problems or concerns. After the pilot test, 50 participants completed the study.

Out of the 50 participants, 1 participant did not answer our demographics questionnaire. Participants' age ranged between 18 and 42 years old (mean = 24.8). There were 31 male and 18 female participants. Of these participants, 42 of them were from science, technology, engineering, or math, 5 from arts, and 2 did not specify their major.

3.6 Performance Measures

As discussed in the introduction, there are many tasks that require high-recall retrieval. We consider two classes of tasks. The first are tasks such as eDiscovery and systematic review. The second is test collection construction for information retrieval evaluation.

For all of our measures, we consider the documents judged by the user as relevant to be the result set. While there may be value in examining the documents judged non-relevant, our study participants were not trying to find non-relevant documents. A participant who decides to use the search engine and only mark relevant documents will not provide any useful non-relevant judgments.

Tasks such as eDiscovery and systematic review can involve two passes of relevance judging. The first pass could be conducted by someone qualified to identify relevant material. The second pass would be conducted by an expert who examines the output of the first pass and makes a final determination about which documents are relevant. For example, in systematic review, a lead researcher might assign to graduate students the task of finding all relevant literature. The graduate students deliver to the lead researcher the documents judged relevant. The lead researcher then examines each document and makes the final decision on which are relevant documents to include in the review.

Both eDiscovery and systematic review want to find all relevant documents. Any document that is missed, could lead to legal issues for eDiscovery or could affect the conclusions made by a systematic review. A good first measure to use is simply the number of documents found and reported by the user as relevant, U_{rel}. Given that different search tasks have different numbers of relevant documents, it can be helpful to normalize the number of relevant documents found, and we use recall as our normalized measure of performance. Recall is the fraction of all relevant documents found by a user: $recall = |U_{rel} \cap R|/|R|$, where U_{rel} is the set of documents judged by the user as relevant, and R is the set of relevant documents as defined by NIST. The NIST assessors act as the expert who decides on final relevance.

In the cases where two passes are used to find relevant documents, each non-relevant document that is returned as relevant by the first pass wastes the time of the final reviewer. Thus, another useful performance measure is the precision of the set of documents returned by the first pass: $precision = |U_{rel} \cap R|/|U_{rel}|$, and F_1 is a useful measure that combines both recall and precision and captures the tradeoff between them: $F_1 = (2 \times recall \times precision)/(recall + precision)$.

IR test collection construction, in most cases, calls for attempting to find all relevant documents. Mistakes made in judging and missing relevant documents affect not only the values of effectiveness measures, but also affect our ability to correctly rank retrieval systems from best to worst. We compare the ranking of the runs submitted to the 2017 TREC Common Core track using the relevance judgments produced by our study participants to the ranking produced with the NIST qrels. For measuring the ranking quality of a run, we use mean average precision (MAP). We eliminate the runs we submitted to the track as well as related runs from another group at our university.

The most common measures of ranking performance are Kendall's rank correlation coefficient, τ, and Yilmaz et al.'s τ_{AP} [37]. The τ_{AP} measure places more weight on high scoring runs. We use Urbano and Marrero's implementation of τ_{AP} [33].

While ranking retrieval systems is the highest priority, it is possible to rank systems well while also producing scores that are very different from the score produced by the NIST qrels. To measure the error in MAP for the runs, we compute the root mean squared error (RMSE):

$$RMSE = \sqrt{\frac{\sum_i^n (nist.map_i - t.map_i)^2}{n}}, \tag{2}$$

where there are n runs, $nist.map_i$ is the MAP as per NIST for the i-th run, and $t.map_i$ is the MAP as produced by the relevance judgments produced by a given variation (treatment) of our high-recall retrieval system.

3.7 Statistical Significance and Modeling

We used generalized linear mixed-effects models, as implemented in the lme4 [8] package in R [25], to measure the statistical significance of our results. We treat our study participants and the search topics as random effects. The independent variables (factors) of our experiment were fixed effects. The factors were whether CAL was with paragraph excerpts or full documents, and whether or not search was available. The dependent variables are the various measures of performance of Section 3.6. We analyze the significance of each factor by building a complete model with all factors and random effects and then a model without the factor of interest. We then compare these two model using a likelihood ratio test that reports a p-value.

4 RESULTS

In the study, participants used 4 variations of our high-recall retrieval system to find as many relevant documents as possible in one hour. With our study participants only able to work for one hour, we did not expect to achieve high-recall on average. What matters in this experiment is the effect of each of the two factors (independent variables) on the performance measures (dependent variables).

Our first experimental factor was whether the CAL component showed a paragraph-length excerpt to participants for judging or whether the CAL component would show the excerpt and also allow the participants to click to view the full document. Our second experimental factor was whether or not the CAL system would be augmented with a search engine. Judgments made in the search system are used to train CAL.

In this section, we will refer to the 4 variations of our system by these shorthands: CAL-P, CAL-D, CAL-P&Search, and CAL-D&Search (see Table 1).

4.1 Main Results

Table 2 is a key/primer to help read our tables of performance measures for those unfamiliar with this style of reporting. The table format mirrors the 2 × 2 factorial experiment design of Table 1. For each combination of factors, we report the mean performance. In addition, we report the marginal means of each factor, i.e. the mean performance for a factor regardless of the other factor. In the lower right hand corner, we report the overall mean of all the treatments. In our analysis, we are interested in the effect each factor has on the experimental outcomes, and as described in Section 3.7, we report the p-values from likelihood ratio tests to determine if a given factor produces a statistically significant difference in the measured outcome.

Our first performance measure is the number of self-reported relevant documents found by study participants, and Table 3a shows these results. In this case, both factors produce statistically significant differences. Only showing a paragraph-length excerpt in CAL results in significantly more relevant documents being reported. Likewise, a CAL-alone system without search is significantly better than a CAL system that includes search. Participants using CAL-P found on average 97.9 relevant documents, which is a 50% improvement over the next best result for CAL-P&Search with 65.4 relevant documents found.

In some scenarios, high-recall retrieval operates with a first pass by one set of workers to find relevant documents, and a second pass where the documents are verified by an expert. Table 3b reports such a scenario where the first pass is by our study participants and the second pass is by the NIST assessors. If a document is unjudged by the NIST assessors, we assume it to be non-relevant. Here again, CAL with paragraphs has statistically significant better performance over CAL with full documents. Search hurts performance, but this is not a statistically significant effect ($p = 0.065$), and we see that search slightly helped CAL with documents while it caused a large decrease in performance for CAL with paragraphs.

Given that some topics have more relevant documents than others, insight can be gained by normalizing the topics by number of relevant documents, and we did this by computing recall, which Table 3c shows. Again, CAL with paragraphs is superior to CAL with documents ($p = 0.002$). For recall, we found a statistically significant interaction effect between the CAL and search factors ($p = 0.04$). Here, search has helped CAL with documents and hurt CAL with paragraphs.

4.2 Ranking of IR Systems

Another important use of high-recall retrieval systems is to find a search topic's relevant documents for evaluation of IR systems. We used each set of judgments produced by our four system variations to score the TREC 2017 Common Core runs. We excluded our runs and related runs from our university. Table 4 reports Kendall's τ, τ_{AP}, and root mean squared error (RMSE) for each treatment's judgment set when compared with NIST's judgment set. We also report bootstrap BCa 95% confidence intervals for each measure. Again, we see that CAL with document excerpts and without search performed the best at ranking the IR systems with the highest τ and τ_{AP} scores.

5 DISCUSSION

Our experimental results support our hypothesis that users working for a given amount of time would find a greater number of relevant documents by judging the relevance of documents by viewing only paragraph-length excerpts rather than full documents. In addition, our results show that CAL without search performs as well or better than a CAL system augmented with interactive searching.

The only difference between CAL-P and CAL-D is that CAL-D allowed the user to view the full document in addition to the paragraph excerpt. Given that we expect judgment quality with full documents to be as good or even better than the judgment quality for paragraphs, it is apparent that allowing people to view full documents slows down their rate of judging. Likewise, because

Table 2: Key/primer for reading Tables 3 and 5.

Search Available	CAL types		Marginal means (search)	
	Full doc available	Paragraph Excerpt only		
No	CAL-D Average	CAL-P Average	Average without Search	p value (Search vs. No Search)
Yes	CAL-D&Search Average	CAL-P&Search Average	Average with Search	
Marginal means (CAL types)	Average of Full doc available	Average with only Paragraph excerpt	Overall Mean	
	p value (Full doc vs. Paragraph)			

Search Available	CAL types		Marginal means (search)	
	Full doc available	Paragraph only		
No	58.3	97.9	78.1*	$p < 0.001$
Yes	51.4	65.4	58.4	
Marginal means (CAL types)	54.8	81.6*	Overall Mean	
	$p < 0.001$		68.2	

(a) Mean Number of User Reported Relevant Documents

Search Available	CAL types		Marginal means (search)	
	Full doc available	Paragraph only		
No	26.5	42.3	34.4	$p = 0.065$
Yes	27.8	33.4	30.6	
Marginal means (CAL types)	27.2	37.8*	Overall Mean	
	$p < 0.001$		32.5	

(b) Mean Number of User Found NIST Relevant Documents

Search Available	CAL types		Marginal means (search)	
	Full doc available	Paragraph only		
No	0.20	0.27	0.24	$p = 0.108$
Yes	0.23	0.25	0.24	
Marginal means (CAL types)	0.22	0.26*	Overall Mean	
	$p = 0.002$		0.24	

(c) Mean Recall

Table 3: The main results of comparing four system variations. We have marked with a $*$ the differences that are significant at $p < 0.05$.

Table 4: Performance measures for the task of test collection construction (see Section 3.6). Shown are Kendall's τ, τ_{AP}, and the $RMSE$ computed based on scoring the TREC 2017 Common Core runs with mean average precision. We compare the 4 high-recall system variations / treatments (see Table 1) with their qrels versus the NIST qrels. Shown in brackets are 95% confidence intervals.

treatment	τ	τ_{AP}	$RMSE$
CAL-P	0.70 [0.54, 0.80]	0.58 [0.43, 0.70]	0.12 [0.11, 0.14]
CAL-D	0.52 [0.32, 0.69]	0.43 [0.27, 0.61]	0.11 [0.09, 0.13]
CAL-P&Search	0.45 [0.26, 0.64]	0.38 [0.19, 0.56]	0.10 [0.08, 0.12]
CAL-D&Search	0.47 [0.27, 0.63]	0.39 [0.21, 0.57]	0.08 [0.06, 0.10]

the number of relevant documents found decreases when search is available, it is clear that people on average find relevant documents at a slower rate via searching than via CAL. Indeed, with CAL-P, participants took an average of only 22.7 seconds per submitted relevance judgment while they took an average of 56.8 seconds using CAL-D.

For CAL-D and CAL-D&Search, users were allowed to click to view the full documents in the CAL interface. We found that users on average made 73% (CAL-D) and 63% (CAL-D&Search) of total judgments from CAL by viewing the full document. It reflects that users like to view the full document in many cases. For those assessments by only viewing excerpts without viewing full documents in CAL, users spent on average 13.2 (CAL-D) and 12.3 seconds (CAL-D&Search) on assessing each excerpt. While for the assessments made by clicking to view full documents in CAL, users spent 52.7 (CAL-D) and 44.1 (CAL-D&Search) seconds on judging each document. In short, viewing a full document cost significantly more time than just viewing a paragraph-length excerpt.

Under the treatments CAL-P&Search and CAL-D&Search, a search interface was provided to users for use. By summing up the active time on the search interface, we found that users on average spent 40% (CAL-P&Search) and 44% (CAL-D&Search) of total time on the search interface. Accordingly, the users made 29% (CAL-P&Search) and 39% (CAL-D&Search) of total judgments by using search. They on average made 5.1 (CAL-P&Search) and 4.5 queries (CAL-D&Search) with the search engine within one hour. They also made 4.8 (CAL-P&Search) and 3.8 (CAL-D&Search) switches between the CAL interface and the search interface.

Unfortunately, adding search to CAL can further slow down the rate of finding relevant documents. With CAL-P&Search, participants took an average of 35.4 second per judgment. CAL-D is so slow that adding search does not slow it down, for with CAL-D&Search, participants took 54.1 seconds per judgment.

Since the availability of search hurts the number of NIST relevant documents found and hurts the recall of CAL-P, the question arises whether this decrease in performance is because of judgment mistakes during search or because of the lower rate of judgments. To answer this question, we examined the rate at which participants found NIST relevant documents compared to the rate at which participants found self-reported relevant documents. To see how these rates changed as participants made judgments, for each participant we computed the cumulative number of NIST relevant documents found vs. the number of self-reported relevant documents found. We then averaged the cumulative number of NIST relevant documents across participants at each number of self-reported relevant documents for the participants who at least had that many self-reported relevant documents. Thus, as the number of self-reported

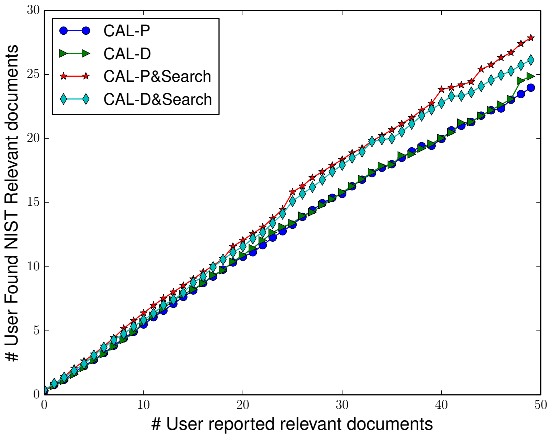

Figure 1: Average number of participant found NIST relevant documents vs. number of self-reported relevant documents for the first 50 self-reported relevant documents.

relevant documents increases, there are fewer participants in the average. Figure 1 shows this analysis. The slope of each curve in Figure 1 is the precision of each treatment.

The first thing noticeable in Figure 1 is that both CAL-P and CAL-D have effectively the same precision. As participants report finding relevant documents, effectively the same fraction are also considered relevant by NIST for CAL-P and CAL-D. The second thing we see is that when search is available, the precision improves regardless of the type of CAL. Thus we can conclude that search hurts CAL with paragraphs because it slows down the rate of judgment rather than somehow hurting the quality of the judgments or CAL's ability to find relevant documents.

Providing users with an ability to view documents in CAL slows down their rate of judgment, and so does providing them with search. However, search improves precision. Thus if a user is to work slowly with CAL-D, the user is better off with CAL-D&Search. While search helps the precision of CAL-P, the decrease in speed overwhelms the small increase in precision, and the user is better with CAL-P alone rather than with CAL-P&Search.

Table 5a reports the mean precision, but these figures need to be interpret with caution. Our analysis with Figure 1 showed that precision of CAL with paragraphs, and CAL with documents were effectively the same, but Table 5a shows CAL with documents has a better precision at a statistically significant level. The issue here is that precision is being measured over different sets and amounts of documents. In high-recall retrieval systems, we expect relevant documents to be easier to find in the beginning and be harder as the search continues. CAL with paragraphs allows users to work much faster and thus explore more of the collection. As the prevalence experienced by the user drops, it is reasonable to expect the user to falsely judge documents relevant at higher rates.

Search Available	CAL types		Marginal means (search)	
	Full doc available	Paragraph only		
No	0.50	0.45	0.48	$p = 0.006$
Yes	0.57	0.52	0.54*	
Marginal means (CAL types)	0.53*	0.48	Overall Mean	
	$p = 0.043$		0.51	

(a) Mean Precision

Search Available	CAL types		Marginal means (search)	
	Full doc available	Paragraph only		
No	0.24	0.31	0.28	$p = 0.063$
Yes	0.28	0.29	0.29	
Marginal means (CAL types)	0.26	0.30*	Overall Mean	
	$p < 0.001$		0.28	

(b) Mean F_1

Search Available	CAL types		Marginal means (search)	
	Full doc available	Paragraph only		
No	0.53	0.52	0.52	$p = 0.091$
Yes	0.57	0.57	0.57	
Marginal means (CAL types)	0.55	0.55	Overall Mean	
	$p = 0.935$		0.55	

(c) Mean Precision at Min. Number of User Reported Relevant Docs.

Table 5: The secondary results of comparing four system variations. We have marked with a $*$ the differences that are significant at $p < 0.05$.

To better compare the precision of different treatments, we report the mean precision of a reduced judgment set in Table 5c. The reduced judgment set is created by considering only the first k documents that the participant reported as relevant. Given a topic, k is the minimum total number of relevant documents that participants reported across all the treatments. We found no statistically significant difference for precision at k judgments with or without the option to search, and with or without the ability to view full documents.

To capture the tradeoff between recall and precision, we report the average F_1 in Table 5b. CAL with paragraphs is better than CAL with documents at a statistically significant level. While search improves precision, the increased precision does not compensate for the loss of recall. As with recall, search improves the F_1 performance of CAL with documents and hurts CAL with paragraphs, and there is a statistically significant interaction effect between CAL and search ($p = 0.03$).

Shown in Table 4, the CAL-P judgment set produces MAP scores for the runs that achieve the highest τ and τ_{AP} rank correlation with the MAP scores produced with NIST qrels. This result is consistent with the results in Table 3 where we found that CAL with documents is worse than CAL with paragraphs. However, the CAL-D&Search produced MAP scores have the lowest RMSE compared to MAP scores from NIST qrels. While CAL-P produces the best ranking of runs compared to NIST, the addition of search appears to help run scores on average be closer to the scores produced with the NIST qrels (lower RMSE). We hypothesize that search may help find high value documents that CAL had not yet found in the hour of searching. All system variations had trouble producing good scores for the automatic runs that used existing qrels (routing runs). Future work is needed to better understand the issues with scoring the routing runs.

6 CONCLUSION

We conducted a user study with 50 participants, each tasked with finding as many relevant documents as possible in one hour. The participants used four variations of a high-recall retrieval system built around an implementation of continuous active learning (CAL). For the CAL component, we tested whether it is better for participants to have the ability to view a full document or for participants to be restricted to viewing a machine selected paragraph-length excerpt. We found for our primary measures of performance—finding more user reported relevant documents, finding more NIST relevant documents, and achieving higher recall—that a single excerpt was better than a full document. We also tested whether or not having the ability to use a search engine to find relevant documents would help or hurt performance. Having access to search hurt performance, but this difference was not always statistically significant.

High-recall information retrieval (HRIR) makes large demands on the user. State-of-the-art HRIR has the user provide relevance feedback on a stream of documents until some stopping criteria is met. In restricting user interaction to the viewing and judging of short document excerpts, our study participants were able to find more relevant documents in the same amount of time as compared to versions of our system that gave the participants more freedom to examine and search for relevant documents. There may be situations where users refuse to be so restricted and demand that they must be able to see full documents as needed. In these cases, it appears that making search available to users would actually improve performance when performance is measured in terms of recall or F_1. We did see an increase in the precision of the system when search is available, and thus carefully limited usage of search early in a high-recall task may be beneficial, but we leave testing of this idea for future work.

ACKNOWLEDGMENTS

This work was supported in part by the Natural Sciences and Engineering Research Council of Canada (Grants CRDPJ 468812-14, RGPIN-2017-04239, and RGPIN-2014-03642), in part by a Google Founders' Award, and in part by the University of Waterloo.

REFERENCES

[1] Mustafa Abualsaud, Nimesh Ghelani, Haotian Zhang, Mark D. Smucker, Gordon V. Cormack, and Maura R. Grossman. 2018. A System for Efficient High-Recall Retrieval. In *SIGIR*. 1317–1320.

[2] James Allan. 2004. HARD Track Overview in TREC 2004: High Accuracy Retrieval from Documents. In *TREC*.

[3] James Allan, Evangelos Kanoulas, Dan Li, Christophe Van Gysel, Donna Harman, and Ellen Voorhees. 2017. TREC 2017 Common Core Track Overview. In *TREC*.

[4] Antonios Anagnostou, Athanasios Lagopoulos, Grigorios Tsoumakas, and Ioannis P. Vlahavas. 2017. Combining Inter-Review Learning-to-Rank and Intra-Review Incremental Training for Title and Abstract Screening in Systematic Reviews. In *CLEF*.

[5] Javed A Aslam, Virgiliu Pavlu, and Robert Savell. 2003. A unified model for metasearch, pooling, and system evaluation. In *CIKM*. 484–491.

[6] Jason R Baron, David D Lewis, and Douglas W Oard. 2006. TREC 2006 Legal Track Overview. In *TREC*.

[7] Gaurav Baruah, Haotian Zhang, Rakesh Guttikonda, Jimmy Lin, Mark D Smucker, and Olga Vechtomova. 2016. Optimizing Nugget Annotations with Active Learning. In *CIKM*. 2359–2364.

[8] Douglas Bates, Martin Mächler, Ben Bolker, and Steve Walker. 2015. Fitting Linear Mixed-Effects Models Using lme4. *J. of Stat. Soft.* 67, 1, 1–48.

[9] Gordon V Cormack and Maura R Grossman. 2014. Evaluation of machine-learning protocols for technology-assisted review in electronic discovery. In *SIGIR*. 153–162.

[10] Gordon V. Cormack and Maura R. Grossman. 2015. Autonomy and Reliability of Continuous Active Learning for Technology-Assisted Review. *CoRR* abs/1504.06868 (2015).

[11] Gordon V. Cormack and Maura R. Grossman. 2017. Technology-Assisted Review in Empirical Medicine: Waterloo Participation in CLEF eHealth 2017. In *CLEF*.

[12] Gordon V Cormack and Mona Mojdeh. 2009. Machine Learning for Information Retrieval: TREC 2009 Web, Relevance Feedback and Legal Tracks. In *TREC*.

[13] Gordon V Cormack, Christopher R Palmer, and Charles LA Clarke. 1998. Efficient construction of large test collections. In *SIGIR*. 282–289.

[14] Maura Grossman, Gordon Cormack, and Adam Roegiest. 2016. TREC 2016 Total Recall Track Overview. In *TREC*.

[15] Maura R Grossman, Gordon V Cormack, and Adam Roegiest. 2017. Automatic and Semi-Automatic Document Selection for Technology-Assisted Review. In *SIGIR*. 905–908.

[16] Donna Harman. 2011. *Information Retrieval Evaluation*. Morgan & Claypool.

[17] Bruce Hedin, Stephen Tomlinson, Jason R Baron, and Douglas W Oard. 2009. *Overview of the TREC 2009 legal track*. In *TREC*.

[18] Evangelos Kanoulas, Dan Li, Leif Azzopardi, and Rene Spijker. 2017. Clef 2017 technologically assisted reviews in empirical medicine overview. In *CLEF*. 11–14.

[19] Inderjeet Mani, Gary Klein, David House, Lynette Hirschman, Therese Firmin, and Beth Sundheim. 2002. SUMMAC: a text summarization evaluation. *Natural Language Engineering* 8, 1 (2002), 43–68.

[20] David Milne. 2014. WikipediaMiner. https://github.com/dnmilne/wikipediaminer.

[21] Makoto Miwa, James Thomas, Alison O'Mara-Eves, and Sophia Ananiadou. 2014. Reducing systematic review workload through certainty-based screening. *Journal of biomedical informatics* 51 (2014), 242–253.

[22] Marcia Munoz and Ramya Nagarajan. 2001. Sentence Splitter. Cognitive Computation Group, CS, UIUC, http://cogcomp.org/page/tools_view/2.

[23] Douglas W Oard, Bruce Hedin, Stephen Tomlinson, and Jason R Baron. 2008. *Overview of the TREC 2008 legal track*. In *TREC*.

[24] Douglas W Oard and William Webber. 2013. Information retrieval for e-discovery. *Foundations and Trends® in Information Retrieval* 7, 2–3, 99–237.

[25] R Core Team. 2014. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. http://www.R-project.org/

[26] Adam Roegiest, Gordon Cormack, Maura Grossman, and Charles Clarke. 2015. TREC 2015 Total Recall track overview. In *TREC*.

[27] Mark Sanderson and Hideo Joho. 2004. Forming test collections with no system pooling. In *SIGIR*. 33–40.

[28] Evan Sandhaus. 2008. The New York Times Annotated Corpus. (October 2008). LDC Catalog No.: LDC2008T19, https://catalog.ldc.upenn.edu/ldc2008t19.

[29] Mark D Smucker and Chandra Prakash Jethani. 2010. Human performance and retrieval precision revisited. In *SIGIR*. 595–602.

[30] Ian Soboroff and Stephen Robertson. 2003. Building a filtering test collection for TREC 2002. In *SIGIR*. 243–250.

[31] Trevor Strohman, Donald Metzler, Howard Turtle, and W. Bruce Croft. 2005. *Indri: A language-model based search engine for complex queries (extended version)*. Technical Report IR-407. CIIR, CS Dept., U. of Mass. Amherst.

[32] John Tredennick. 2011. E-Discovery, My How You've Grown! https://catalystsecure.com/blog/2011/04/e-discovery-my-how-youve-grown/.

[33] Julián Urbano and Mónica Marrero. 2017. The Treatment of Ties in AP Correlation. In *ICTIR*. 321–324.

[34] Ellen M Voorhees. 2001. Evaluation by highly relevant documents. In *SIGIR*. 74–82.

[35] Ellen M Voorhees and Donna K Harman. 2005. The text retrieval conference. *TREC: Experiment and evaluation in information retrieval* (2005), 3–19.

[36] Ellen M Voorhees, Donna K Harman, et al. 2005. *TREC: Experiment and evaluation in information retrieval*. Vol. 1. MIT press Cambridge.

[37] Emine Yilmaz, Javed A. Aslam, and Stephen Robertson. 2008. A New Rank Correlation Coefficient for Information Retrieval. In *SIGIR*. 587–594.

[38] Haotian Zhang, Mustafa Abualsaud, Nimesh Ghelani, Angshuman Ghosh, Mark D. Smucker, Gordon V. Cormack, and Maura R. Grossman. 2017. UWaterlooMDS at the TREC 2017 Common Core Track. In *TREC*.

[39] Haotian Zhang, Gordon V. Cormack, Maura R. Grossman, and Mark D. Smucker. 2018. Evaluating Sentence-Level Relevance Feedback for High-Recall Information Retrieval. *CoRR* abs/1803.08988 (2018).

[40] Haotian Zhang, Jimmy Lin, Gordon V Cormack, and Mark D Smucker. 2016. Sampling Strategies and Active Learning for Volume Estimation. In *SIGIR*. 981–984.