Increasing the Efficiency of High-Recall Information Retrieval

by

Haotian Zhang

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Computer Science

Waterloo, Ontario, Canada, 2019

Examining Committee Membership

The following served on the Examining Committee for this thesis. The decision of the Examining Committee is by majority vote.

External Examiner: Ian Soboroff
Group Leader, Information Retrieval Group
National Institute of Standards and Technology

Supervisor(s): Mark D. Smucker
Associate Professor, Department of Management Sciences
University of Waterloo
Gordon V. Cormack
Professor, School of Computer Science
University of Waterloo

Internal Member: Maura R. Grossman
Research Professor, School of Computer Science
University of Waterloo
Daniel M. Berry
Professor, School of Computer Science
University of Waterloo

Internal-External Member: Lukasz Golab
Associate Professor, Department of Management Sciences
University of Waterloo

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

The goal of high-recall information retrieval (HRIR) is to find all, or nearly all, relevant documents while maintaining reasonable assessment effort. Achieving high recall is a key problem in the use of applications such as electronic discovery, systematic review, and construction of test collections for information retrieval tasks. State-of-the-art HRIR systems commonly rely on iterative relevance feedback in which human assessors continually assess machine learning-selected documents. The relevance of the assessed documents is then fed back to the machine learning model to improve its ability to select the next set of potentially relevant documents for assessment. In many instances, thousands of human assessments might be required to achieve high recall. These assessments represent the main cost of such HRIR applications. Therefore, their effectiveness in achieving high recall is limited by their reliance on human input when assessing the relevance of documents. In this thesis, we test different methods in order to improve the effectiveness and efficiency of finding relevant documents using state-of-the-art HRIR system. With regard to the effectiveness, we try to build a machine-learned model that retrieves relevant documents more accurately. For efficiency, we try to help human assessors make relevance assessments more easily and quickly via our HRIR system. Furthermore, we try to establish a stopping criteria for the assessment process so as to avoid excessive assessment. In particular, we hypothesize that total assessment effort to achieve high recall can be reduced by using shorter document excerpts (e.g., extractive summaries) in place of full documents for the assessment of relevance and using a high-recall retrieval system based on continuous active learning (CAL). In order to test this hypothesis, we implemented a high-recall retrieval system based on state-of-the-art implementation of CAL. This high-recall retrieval system could display either full documents or short document excerpts for relevance assessment. A search engine was also integrated into our system to provide assessors the option of conducting interactive search and judging. We conducted a simulation study, and separately, a 50-person controlled user study to test our hypothesis. The results of the simulation study show that judging even a single extracted sentence for relevance feedback may be adequate for CAL to achieve high recall. The results of the controlled user study confirmed that human assessors were able to find a significantly larger number of relevant documents within limited time when they used the system with paragraph-length document excerpts as opposed to full documents. In addition, we 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. Moreover, integrating sampling methods with active learning can yield accurate estimates of the number of relevant documents, and thus avoid excessive assessments.

Acknowledgements

I was fortunate to have two supervisors as my mentors for my PhD study. First and foremost, I give my deepest gratitude to Prof. Mark D. Smucker, for his endless support and guide during my PhD study. He has guided me on the exploration of the information retrieval research and encouraged me in my choice of PhD research direction. He always instructed me to be an independent and self-motivated researcher. We also spent a lot of time together and worked on papers till very late before submission deadline. I really enjoyed being mentored by Mark and appreciate his influence on my whole PhD study.

Prof. Gordon V. Cormack, my co-supervisor, who is a pioneer of the high-recall information retrieval (HRIR) research. I first approached the high-recall information retrieval problem when participating the TREC Total Recall track 2015, which was organized by Gordon. He gave me numerous valuable guidances and advices on tackling this problem. He suggested me work on improving the efficiency of HRIR when I was stuck with increasing the effectiveness. He always encourages me to conduct research more rigorously and considerately.

I would like to thank all the other members in my PhD examination committee: Prof. Maura Grossman, who is also a pioneer in the HRIR field. She let me know the broad background and the real world of this research area. Her helpful advices and guidances help me insist on my research topics. Prof. Daniel Berry, who met me in the 2017 Cheriton Research Symposium poster session. He gave me a amount of suggestions on my research paper and how to write the PhD thesis. Prof. Lukasz Golab, who severs as my internal-external member and Dr. Ian Soboroff, who severs as my external examiner.

I would like to thank Prof. Charles L.A. Clarke, who guided me in the beginning of my PhD study. We worked together on the participation of Total Recall Track 2015 and wrote two research papers. He always acted as an additional supervisor when Mark and Gordon were not available. It was a great time to work with Charlie and get inspired by his insightful thoughts. I also really appreciate the help and guide from Prof. Jimmy Lin. Jimmy is a great researcher to work with. He always reminds me to be a smart and sharp researcher. I truly appreciate his help on the volume estimation project.

Thanks to the many colleagues who inspired me during my PhD study in the data system group, most notably Nimesh Ghelani, Mustafa Abualsaud, Luchen Tan, Guarav Baruah, Aiman Al-Harbi, Royal Sequiera, Salman Mohammed, Zhucheng Tu, Yipeng Wang, Jinfeng Rao, Wei Yang, Adam Roegiest, Adriel Dean-Hall, and many others. I would like to thank Ruth Taylor for taking the time to proofread this thesis.

Dedication

To my parents and Violet.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

In many application areas, such as electronic discovery (eDiscovery), systematic review, and the construction of test collections for information retrieval research, the need to find all, or nearly all, relevant documents is critical. Any solution to this research problem has as its goal high-recall information retrieval (HRIR), or finding as many relevant documents as possible for a given information need with minimal assessment effort. In the fields of eDiscovery and information retrieval evaluation, many methods to achieve high recall require the help of search experts [Hogan et al., 2008] or topic- or database-specific training [Zhang and Zhang, 2010; Zhang et al., 2009].

Information retrieval (IR) is finding relevant information (usually documents) that satisfies an information need (usually a query) from a large data collection. One of the traditional IR tasks is ad-hoc search on the Web. The commercial search engines (e.g., Google or Bing) return 10 blue links for a user query. These web search tasks may not directly address the need of high recall (finding all the relevant documents) but focus on high precision (showing top-ranked results to users). In such cases, users may stop reviewing documents once their information needs are satisfied. In contrast to these high-precision tasks, HRIR aims at finding all or substantially all the relevant documents for an information need, using the least amount of (human) effort possible. In the early days without the invention of computers and software, assessors need to review through mountains of documents. The only solution to find all the relevant documents is reviewing documents one by one, until the entire set is complete [Borden, 2010]. However, this process is usually both time- and money- consuming. Meanwhile, assessors may waste a lot of time reviewing non-relevant documents. To solve this problem, the HRIR methods always try to select the relevant documents for assessors to review in order to reduce the review effort. More specifically, the HRIR methods allow users to use keyword searches or some other methods (e.g., machine learning) to find all the relevant documents with the least effort possible. The judgments from users can be utilized to help improve the HRIR methods.

State-of-the-art HRIR methods heavily rely on a large number of human inputs, either to issue queries and judge documents returned by the search engine (interactive search and judging) or to assess the relevance of machine learning-selected documents. The effectiveness of current HRIR methods in achieving high recall is limited by their reliance on such human inputs. In most cases, the effort of human assessment when judging documents represents the primary cost of such high-recall tasks. Being able to reduce the assessment cost of finding relevant documents will be beneficial for achieving high recall within a limited monetary and temporal budget. In addition, knowing when to stop further assessment during high-recall retrieval will help avoid excessive review cost.

For this thesis, we developed various methods to improve state-of-the-art HRIR system from different perspectives. First, we tried to improve the effectiveness of document retrieval by applying different machine-learned classifiers, incorporating additional document features, and integrating query expansion. Second, we tried to enhance the efficiency of document assessment by using shorter document excerpts for the assessment of relevance, enabling the human assessors to judge documents more quickly. Third, we integrated various sampling methods into the active learning process to estimate when to stop assessing, thereby avoiding excessive assessments while still ensuring high recall.

We first approached the high-recall problem by participating in TREC Total Recall Track 2015 [Roegiest, Cormack, Grossman and Clarke, 2015]. The organizers of Total Recall Track provided a baseline method—baseline model implementation (BMI) — which is based on continuous active learning [Cormack and Grossman, 2014]. We modified BMI to incorporate more document features and submitted runs for evaluation. The results revealed that no other method was able to beat BMI consistently. We have yet to find a mode that is demonstrably superior to the continuous active learning protocol with respect to the accurate retrieval of relevant documents. For that reason, approaching the problem from another angle, we tried to improve the efficiency of document assessment by helping assessors find a larger number of documents within a certain time limit.

One of our main contributions is our evaluation of whether assessors can find a larger number of relevant documents within a limited time by confining their interactions to the assessment of shorter document excerpts. Past research has shown that humans can assess the relevance of documents faster and with little loss in accuracy by judging short document excerpts (e.g., extractive summaries) in place of full documents [Smucker and Jethani, 2010]. To test the hypothesis that using short document excerpts can reduce assessment time and effort required for high-recall retrieval, we conducted a simulation experiment, and separately, a 50-person controlled user study. We designed a high-recall retrieval system based on continuous active learning (CAL) that could display either full documents or short document excerpts for relevance assessment. In the simulation study, we tried to answer two question: (1) Is assessing short document excerpts as effective for achieving high recall as assessing full documents? (2) How could we integrate document excerpt-level relevance feedback into CAL? Some research has helped to find suitable approaches to selecting documents for assessors to review [Cormack and Grossman, 2014, 2015*a*]. There is relatively less research about selecting short document excerpts for relevance assessment [Tombros and Sanderson, 1998; Sanderson, 1998; Smucker and Jethani, 2010], especially in a high-recall scenario. For this thesis, we implemented various strategies to incorporate document-excerpt-level relevance feedback into CAL. In a simulation study, in which assessors were simulated by a predefined relevance label set, we found that presenting a single sentence from each document for relevance assessment in CAL was enough to achieve recall as high as assessment based on full documents [Zhang, Cormack, Grossman and Smucker, 2018]. In a follow-up controlled user study, we investigated the effects of assessing document excerpts for high-recall retrieval [Zhang, Abualsaud, Ghelani, Smucker, Cormack and Grossman, 2018]. We deployed a high-recall retrieval system to evaluate whether human assessors were able to find a large number of relevant documents by assessing document excerpts alone [Abualsaud, Ghelani, Zhang, Smucker, Cormack and Grossman, 2018]. The participants of the controlled study were asked to use our HRIR system to find as many relevant documents as possible within one hour. We found that: (1) assessors were able to find a larger number of relevant documents within a limited time by viewing the machine-learning-selected document excerpts than by viewing full documents; and (2) giving assessors more options to interact with our HRIR system did not improve their ability to find relevant documents. In our experiment, some variants of our HRIR system allowed assessors to view the full document content or to use a search engine. We compared the number of relevant documents found by using each variant within the same limited time frame. The results showed that these additional options slowed down assessors' assessments, thus, reducing the total number of relevant documents found within a limited time. In addition, we considered the log of user behaviour and measured the effects of different user interactions on the effectiveness of finding relevant documents. We found that some user interactions significantly reduced the assessment speed (e.g., viewing the full document). Other interactions, improved the precision of assessments (e.g., using a search engine). We detail this analysis in Chapter 6.

Another problem in high-recall retrieval is when to stop assessing, especially when the dataset is huge. In other words, the number of remaining relevant documents should be estimated during the assessment process to determine when to terminate the review. We explored several different sampling strategies and integrated them to CAL into estimate the number of relevant documents [Zhang et al., 2016]. The results showed that the combination of sampling methods and CAL was able to estimate the volume of relevant documents accurately without requiring a large assessment effort.

1.1 Motivation

High-recall information retrieval (HRIR) is integral to many tasks that require the finding of all, or nearly all, relevant documents in a collection [Cormack and Grossman, 2014]. Example applications of HRIR include electronic discovery (eDiscovery), systematic review, and the construction of information retrieval (IR) test collections. The high-recall application most relevant to the context of this thesis is eDiscovery (the discovery of electronically stored information) [Oard et al., 2013]. eDiscovery is a rapidly growing field that originates from "civil discovery." In civil law, each party to a lawsuit must provide the other party all the documents that are relevant to a given information request by the opposing party. Some IR methods (e.g., a simple keyword search) have been widely and successfully applied in eDiscovery [Oard et al., 2010]. However, high-recall retrieval requires the retrieval of a complete set of relevant documents at reasonable monetary and temporal costs [Cormack and Grossman, 2015*b*]. In theory, an ideal eDiscovery retrieval would achieve 100% recall, that is, the complete set of relevant documents (no False Negatives), and 100% precision, that is, accurate retrieval of only the relevant documents (no False Positives). However, such a perfect search does not exist in the real world. [1]

Web-search systems such as Google, Bing, and Yahoo do not adequately address the needs of high recall [Roegiest, 2017]. In most cases, these web search systems focus on improving the early precision of returned results ("10 blue links"), because users tend to be more interested in top-ranked results than in all the relevant documents. Therefore, the challenges faced by high-recall IR methods are quite different from the challenges faced by IR methods in other fields (such as web search) [Oard et al., 2013]. There are some differences between high-recall IR methods and other precision-oriented IR methods. First, Oard et al. [2013] point out that the evaluation of high-recall retrieval emphasizes the returned set of results rather than the rankings of results. Second, the target of eDiscovery is to achieve high recall with minimal assessment effort. For many other common IR applications, such as web search, users are more interested in the highly ranked results. Therefore, achieving high precision for those top-ranked results is more important than

[1] https://www.brainspace.com/blog/e-discovery-searching-for-the-narrative/

achieving high recall. In a nutshell, the high-recall task and the evaluation of high-recall methods are quite different from many other common information retrieval problems.

The global market for vendors of eDiscovery software and services has been estimated at US$1 billion in 2016, and is expected to reach $13 billion by 2023 [Wood, 2017]. The primary cost of eDiscovery lies in manual assessment of documents, which takes up 70% of total cost [Peacock, 2009]. With the digitization of information, the sizes of collections to search have grown rapidly. One eDiscovery firm reports that a typical case requires the review of between 600,000 and 1 million documents [Tredennick, 2011]. For legal discovery, each document in a collection would traditionally be reviewed by an attorney, and review would take a few minutes per document [Oard et al., 2013], This is called *linear review*. A study of "Second Request" concerning Verizon's acquisition of MCI [Roitblat et al., 2010] found that a team of 225 attorneys spent about 4 months, 7 days a week, and 16 hours per day assessing over 2.3 million documents (1.6 million documents after eliminating duplicates). The team found nearly 200,000 relevant documents, at a cost of over US$13.6 million. It cost approximately around $8.50 to assess each document. A similar scenario exists in the construction of test collections for information retrieval research. The TREC Legal Track [Baron et al., 2006] was run from 2006 to 2011, and two reusable test collections were developed: one collection contained nearly 7 million scanned business records; the other comprised roughly half a million email messages. Baron et al. reported an average assessment rate of 24.7 documents per hour for different topics at the TREC 2006 Legal Track. At the TREC 2008 Legal Track [Oard et al., 2008], the organizers reported that it took a collective 631.2 hours to review 13,543 documents, or a rate of 21.5 documents per hour. Based on these statistics, we can infer that several factors affect the final total assessment cost. Important 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.

By combining these factors, we can devise different ways to evaluate the total assessment cost for reaching high recall. One way is to evaluate the effectiveness of a HRIR method. In other words, an effective HRIR method should be able to achieve high recall or find a majority of relevant documents using the least number of assessments. Therefore, maintaining high precision (avoid selecting non-relevant documents for assessors to review) is critical for improving effectiveness. Some tasks in the TREC Legal track and TREC Total Recall track created different measures to evaluate the effectiveness of HRIR methods. The evaluation emphasizes the total number of assessments to achieve a certain recall but ignores the total time spent on those assessments. Therefore, another perspective of high recall evaluation is to measure the total time of the assessments. This thesis focuses on evaluating high-recall retrieval methods according to the number of relevant documents retrieved within a certain time limit.

The effectiveness of a high-recall retrieval system is determined by its underlying methods. There are several generations of high-recall methods. During the initial stage of eDiscovery research, keyword search methods dominated the applications in this area. Assessors kept reformulating their search queries and judging documents returned from a search engine. Thus, the early eDiscovery systems relied heavily upon human-issued queries. However, state-of-the-art HRIR methods suggest that the manual issuance of queries can be replaced by a continuous active learning (CAL) protocol in which the assessors only assess the documents selected by a machine learning model. The learning model in CAL is able to automatically select likely relevant documents for assessors to judge. In each iteration of CAL, the learning model continuously retrains and improves continuously after receiving the relevance assessments of judged documents. For some tasks of the TREC Legal Track [Cormack and Mojdeh, 2009] and TREC Total Recall Track [Roegiest, Cormack, Grossman and Clarke, 2015; Grossman et al., 2016], CAL showed high effectiveness and outperformed other methods (including manual methods) for achieving high recall with limited assessment. In this thesis, we designed and implemented a high-recall retrieval system based on state-of-the-art CAL implementation: baseline model implementation (BMI). BMI is an implementation of the CAL protocol [Grossman et al., 2017] employed in the TREC 2015 and 2016 Total Recall Tracks. BMI automatically selects full documents for relevance assessment, and then the relevance feedback from the assessed full documents is fed back to improve the learning model. For our HRIR system, we incorporated document-excerpt-level relevance feedback into CAL. In addition, different strategies for integrating document-excerpt-level relevance feedback into BMI were compared.

As mentioned above, we tried to improve state-of-the-art HRIR method from three perspectives: (1) enhancing the effectiveness of retrieving relevant documents, (2) improving the efficiency of making relevance assessment, and (3) determining when to stop assessment. This thesis presents an investigation and evaluation of proposed methods. We next summarize the work in this thesis that addresses how to improve and evaluate the performance of high-recall information retrieval.

1.2 Overview

The demand for high-recall information retrieval has existed for a long time, and many high-recall retrieval systems have been developed to solve this problem. In this thesis, we try to improve state-of-the-art HRIR system in order to find relevant documents more effectively and efficiently. I next describe how we approach and tackle this problem in each chapter.

For Chapter 3, we approached the problem of achieving high recall by participating in the TREC Total Recall Track in 2015 [Roegiest, Cormack, Grossman and Clarke, 2015] and implementing a high-recall system based on the baseline model implementation (BMI) [Zhang et al., 2015]. Our system made some modifications to the BMI, including extending document features, applying different classifiers, and adding different strategies for selecting a seed set of documents. We submitted our results to Total Recall Track 2015 for evaluation and compared our modified BMI with other submitted retrieval systems using the provided test collections.

To improve the effectiveness of finding relevant documents using limited assessment effort, for Chapter 4, we examined whether assessing a single extracted sentence from the document, instead of the whole document, would provide effective relevance feedback in CAL. We modified BMI and incorporated the sentence-level relevance feedback into the CAL process. In each iteration of relevance feedback, there exist binary choices (selecting excerpts or documents) on three different dimensions. These dimensions are (1) presenting excerpts or full documents to assessors for relevance assessment; (2) retraining the machine learning model using the judged documents or judged excerpts; and (3) scoring and ranking documents or excerpts in order to select the next-most-likely to be relevant document for assessment. In this thesis, we investigated different choices in each of these dimensions and determined the best combination for achieving high recall.

For Chapter 5, we conducted, to the best of our knowledge, the first controlled study using human subjects to test human assessment performance in judging document excerpts in CAL. For a given topic, assessors were required to find as many relevant documents as possible within one hour, using different CAL system variants. We implemented a high-recall information retrieval system (HiCAL) for conducting the experiment. The basic system allowed assessors to judge the machine learning-selected paragraph-length excerpts. The other system variants differed either in the presentation of the full document content, or in offering the option of using a search engine. We compared different system variants according to the number of relevant documents found within a limited time frame.

For Chapter 6, we delved further into the assessor behaviour data collected from the user study and tried to understand the underlying behaviours that led to the observed results. We also measured the judging performance of users under each system variant.

For Chapter 7, we investigated the problem of when to stop the assessment process in CAL. In this thesis, we applied several different sampling strategies in CAL to estimate the number of relevant documents. CAL usually reaches a plateau (knee point) when the prevalence of relevant documents drops significantly. At that point, sampling strategies can be used to estimate the prevalence of relevant documents in the rest of the collection. We evaluated different sampling methods for estimating the number of documents for given topics.

I review related work in Chapter 2 and conclude the thesis in Chapter 8.

1.3 Contributions

In this thesis, we make the following contributions:

- By participating in TREC Total Recall Track 2015, we found that no system consistently outperforming the baseline model BMI run. Our modified BMI implementation shows some improvements on certain topics. However, overall improvements are not significant, nor are they consistent on all topics and datasets. (Chapter 3)
- By simulating the continuous active learning process and using document excerpts for relevance assessment to achieve high recall, we found, based on the same number of assessments, that judging a single extracted sentence for relevance retrieves the same number of relevant documents as judging the full document in CAL. (Chapter 4)
- According to the simulation results of incorporating sentence-level relevance feedback into CAL, the best combination of choices from the three dimensions is to select the highest-scoring sentence from the highest-scoring document for assessors to review, and retrain the machine learning model using judged documents to improve its relevance ranking. (Chapter 4)
- In our 50-person controlled user study, by comparing the number of relevant documents found within one hour, we found that CAL combined with judging only paragraph-length excerpts helps users find the largest number of relevant documents. (Chapter 5)
- In the controlled user study, allowing users to view full documents in CAL and allowing to conduct searches significantly reduced the rate at which users found relevant documents. Restricting interactions improved users' ability to find a larger number of relevant documents. (Chapter 5)
- A combination of sampling strategy and CAL protocol can be used to estimate the number of relevant documents accurately without requiring a large number of extra

assessments. After CAL reaches its plateau, where the prevalence of relevant documents drops, sampling methods can be used to estimate the prevalence of relevant documents in the remaining collection. (Chapter 7)

- Among different sampling strategies, the stratified sampling method yields the most accurate estimate of the number of relevant documents using the same amount of effort. (Chapter 7)

Chapter 2

Background and Related Work

2.1 High-Recall Information Retrieval: The Problem

High-recall information retrieval (HRIR) refers to the problem of assessors' needing need to identify all, or nearly all, relevant documents for a given topic using a reasonable review effort, in terms of assessment time and budget. Examples of applications of HRIR include electronic discovery (eDiscovery), systematic review, and the construction of test collections for information retrieval research. Currently, most HRIR research focuses on improving the effectiveness of technology-assisted review (TAR), a tool for eDiscovery in litigious, regulatory, and access-to-information contexts that use human assessments to find substantially all documents that meet specified criteria [Grossman and Cormack, 2014; Cormack and Grossman, 2014; Carroll, 2013]. The underlying methods applied in TAR include, but are not limited to, active learning, uncertainty sampling, and interactive search and judging (ISJ). These approaches in TAR aim to help assessors find as many relevant documents as possible using a reasonable review effort.

A similar HRIR problem also exists in the area of systematic review. One example comes from the field of evidence-based medicine, where there is a need to find all relevant research related to a certain topic (e.g., a treatment or diagnostic test). Kanoulas et al. [Kanoulas et al., 2017; Goeuriot et al., 2017] recently launched a systematic review task in CLEF 2017 eHealth lab to study this problem.

Constructing an ideal test collection for information retrieval research similarly needs to achieve high recall. If a test collection contains all the documents relevant to a given topic, different retrieval systems can be evaluated accurately. Large reusable test collections for information retrieval evaluation have been developed by the information retrieval evaluation community active in forums such as the Text Retrieval Conference (TREC), the Conference and Labs of the Evaluation Forum (CLEF), and the NII Testbeds and Community for Information Access Research (NTCIR).

2.2 High-Recall Information Retrieval Methods

2.2.1 Search-Based Approaches

Evolution of Search Technologies in eDiscovery

The earliest eDiscovery method is called *linear review*. It entailed lawyers' manually reviewing mountains of poorly organized documents in serial order [Borden, 2010; Oard et al., 2013]. The process is tedious, time-consuming, and extremely inefficient. With the advent of widespread use of computer systems and software, Boolean search was added to improve document retrieval. Many common legal search tools such as Lexis® [1] and Westlaw® [2] apply Boolean searches in which assessors can use logical operators (e.g., AND, OR, and NOT) to improve recall. In general, Borden [2010] states that the Boolean search represents the first generation (1G) of search methods in eDiscovery.

In the same paper, Borden summarizes the evolution of search technologies used in eDiscovery. Second generation (2G) technologies began to incorporate more complicated feature engineering methods to represent documents. Among different 2G methods, TF-IDF, short for term frequency inverse document frequency, is one of the most widely used ranking methods. Another commonly used method among 2G technologies is *concept clustering*. It groups the documents that overlap to some degree together. Assessors can code the documents to form a single cluster and judge the grouped documents together. Another method in 2G is using synonymy. A synonymous search engine can return documents containing terms synonymous with the query terms. The synonymous terms are generated based on their context.

According to Borden, third-generation methods (3G) leveraged all the meaningful information in the document and helped alleviate the 2G methods' dependence on search terms. 3G methods return not only the documents containing the search terms but also documents that do not contain the search terms but show other similarities. These similar relevant documents might be located in the same directory or created and edited by the

[1] http://www.lexisnexis.com/litigation/products/ediscovery
[2] http://www.westlawnextcanada.com/litigation-solutions/legal-discovery

same person. Strong cohesion among these similar documents allows assessors to assess these documents together.

One of the earliest studies about the use of Boolean search to achieve high recall was conducted by Blair and Maron [1985]. In this 1985 experiment, a team of lawyers and paralegals were allowed to perform as many Boolean queries as they thought necessary to achieve high recall. They continued assessing until they were satisfied with the retrieved set of documents, which they believed represented 75% of the relevant documents. In fact they had only found 20%.

Blair and Maron concluded that recall was so low [Blair, 1996], because it is extremely difficult for human assessors to predict all the related query terms, or all the combinations of different terms that would retrieve all the relevant documents. Blair and Maron argued that, faced with such a large dataset, two lawyers and two paralegals were insufficient for a comprehensive understanding of all aspects of the information need. In general, the lawyers were able to retrieve only a small subset of important information.

Interactive Search and Judging

One commonly used method for the HRIR is to conduct multiple searches. This is called *interactive search and judging* (ISJ) [Cormack et al., 1998]. In ISJ, assessors repeatedly reformulate queries and assess the top-ranked results returned from a search engine [Zhang, Abualsaud and Smucker, 2018].

A study conducted by Cormack et al. [1998] found that using ISJ to find relevant documents can yield a set of relevance judgments (qrels) whose quality is comparable to the gold standard relevance set provided by NIST but requires considerably less review effort. The Waterloo team composed of Cormack et al. implemented and used an interactive search system (MultiText), which performed well in TREC 4 and TREC 5 [Clarke et al., 1995; Clarke and Cormack, 1996]. The MultiText system supported the use of Boolean query and relevance ranking based on the length and number of passages that satisfy the query. A principal difference between the study by Blair and Maron and the one by Cormack et al. was that the MultiText system used the “shortest substring ranking and an interface that displayed relevant passages with the search terms highlighted and allowed judgments to be recorded,” whereas Blair and Maron used Boolean searches and reviewed printed versions of entire documents. The Waterloo team spent in total 105 hours, on average 2.1 hours judging 261 documents per topic on the TREC 6 collection.

The judgments collected from the Waterloo team showed reasonable agreement with the NIST gold standard. Qrels derived from the Waterloo team and NIST qrels identified almost the same number of relevant documents ($3,900$ for Waterloo and $3,923$ for NIST). However, only 40% of the judgments were the same in both qrels sets. The Waterloo team's judgments achieved macro-averaged recall of 0.8 (per topic). According to Kendall's tau (τ) rank correlation (refer to Equation 2.14 for a definition of Kendall's τ) , with respect to the NIST's mean average precision (MAP) results, the Waterloo team's ISJ qrels achieved $\tau = 0.89$, which was slightly lower than Voorhees's Kendall's τ threshold of 0.9 (indicating that the qrels generated by the Waterloo team were not distinguishable from the NIST gold standard qrels [Voorhees, 2000, 2001*a*]). Nevertheless, the judgment pool composed of NIST assessors was five times larger than the ISJ judgment pool (refer to Section 2.2.3 for a definition of the pooling method). Cormack et al. further explained the lower τ correlation compared to NIST qrels was due mainly to the use of different approaches to building the relevance judgment set. Cormack et al. concluded that disagreements between the Waterloo team and the NIST assessors was also a factor.

In another study, Sanderson and Joho [2004] further confirmed the effectiveness of ISJ for building test collections. In this study, Sanderson and Joho collected all the manual runs submitted to the ad-hoc task of TREC 5, 6, 7, and 8. They treated each of the submitted manual runs as a simulation of the ISJ process. They formed a qrels set using the top-$1,000$ ranked documents from each of the manual runs and then padded the remaining ranks with the NIST qrels. The automatic runs submitted were then ranked by mean average precision (MAP) scores. The resulting ranked list was then correlated (using Kendall's τ) with the ranked list generated from the NIST qrels. From all the submitted manual runs, they found that using 69% of manual runs can form a qrels set of as high quality (Kendall's $\tau > 0.9$) as NIST qrels. As a conclusion, Sanderson and Joho [2004] reported, "ISJ is broadly applicable regardless of retrieval system used or people employed to conduct the searching process."

Soboroff and Robertson [2003] applied a variation of ISJ in constructing the relevance judgment set for the TREC 2002 Filtering Track [Robertson and Soboroff, 2002]. Soboroff and Robertson conducted an ad-hoc search to retrieve and label the 100 most-likely-relevant documents for each topic. Then these labelled documents were sent to different retrieval systems and used to provide relevance feedback. The top-ranked documents returned from each retrieval system were then fused using CombMNZ fusion [Montague and Aslam, 2002] for further human assessment. The above process was repeated until very few relevant documents could be found by the ad-hoc search engine, or until the assessment budget was exhausted. Soboroff and Robertson augmented the labelled set using a pooling method after receiving the submitted runs from different participants. They found that the Kendall's τ correlation between the MAP scores produced from the ISJ labelled set and the MAP scores derived from the augmented labelled set was 0.91. Soboroff and Robertson also observed that additional assessment after ISJ judgment did not significantly affect the evaluation results.

2.2.2 Machine Learning Based Approaches

Active Learning

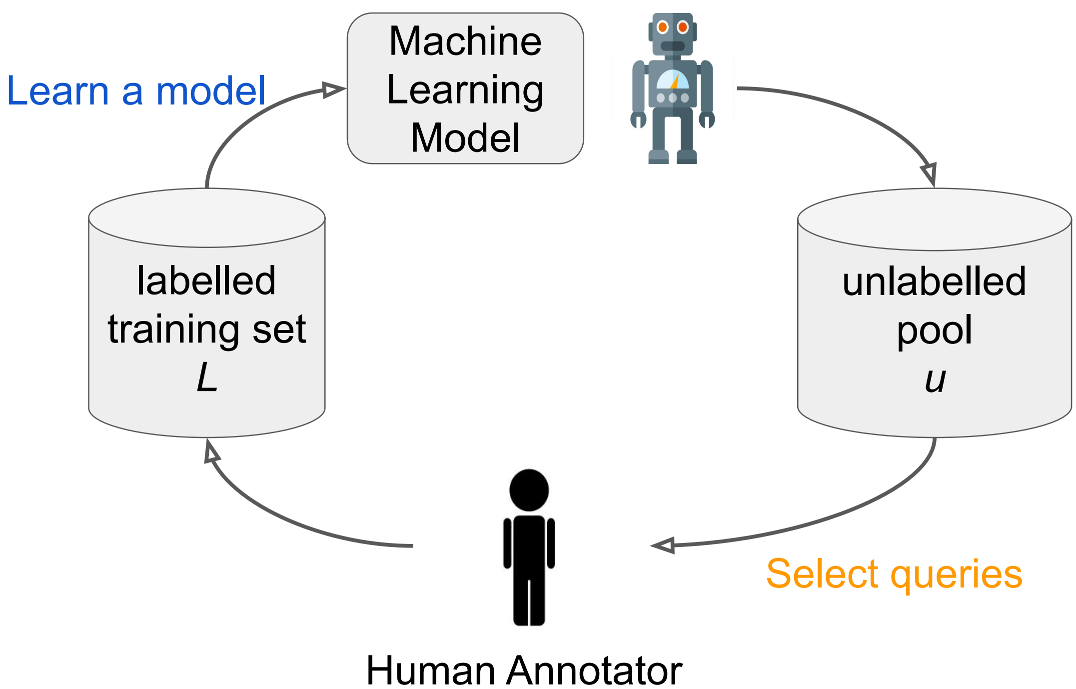

Figure 2.1: The pool-based active learning cycle from Settles's active learning overview paper (page 9) [Settles, 2009].

Recent research has started to use machine learning methods to select documents for assessors to judge [Cormack and Mojdeh, 2009; Cormack and Grossman, 2014]. Without any need to conduct search queries, assessors need only to keep assessing the machine-learning-selected documents to achieve high recall. This process is similar to the active learning protocol, which is a subfield of machine learning [Settles, 2012; Prince, 2004]. According to Settles's definition [Settles, 2009], an active learning system aims to achieve high accuracy using as few labelled instances as possible, thereby reducing the cost of labelling. An active learner can pose *queries* to select unlabelled instances for labelling by a human annotator.

Pool-based sampling in active learning was introduced by Lewis and Gale [1994]. It uses a small set of labelled data L and a large pool of unlabelled data u, as shown in Figure 2.1. The active learner selects informative instances from the pool for a human annotators to label. All the instances labelled thus far are then used to retrain the machine learning model. The above process repeats until the annotation budget is exhausted.

Each iteration of a pool-based active learning protocol for a document classification task selects only a subset of documents from the pool for labelling, thereby reducing the total assessment effort [Settles, 2012; Lewis and Gale, 1994]. There are three main sampling methods for active learning:

- *Uncertainty sampling* selects hard-to-classify documents to label;
- *Relevance sampling* selects the most likely to be relevant documents to label;
- *Random sampling* selects random documents from unassessed documents to judge.

Lewis and Gale [1994] conducted a study to compare different sampling methods in active learning that used the Bayes' Rule to estimate the posterior probability of document d in a given category C_i

$$p(C_i|d) = \frac{p(d|C_i) \times p(C_i)}{\sum_{j=1}^{n} p(d|C_j) \times p(C_j)} \tag{2.1}$$

where C_i is a disjoint set of classes to which a document d might belong. n is the total number of classes. d is the document vector containing the features of multiple words. Given a classifier that estimates $P(C_i|d)$, uncertainty sampling selects a document d with the $P(C_i|d)$ value closest to 0.5. Lewis and Gale state that the classifier is most uncertain about the class label of the document d when $P(C_i|d) = 0.5$. For relevance sampling, the classifier chooses the document with the highest $P(C_i|d)$ value to label.

Lewis and Gale compared uncertainty sampling, relevance sampling and random sampling in performing a text categorization task for titles of news articles. The results suggest that the effectiveness (F_1 score) of uncertainty sampling and relevance sampling using a large sample size is similar. Random sampling yielded the lowest F_1 score. In addition, Lewis and Gale found that uncertainty sampling can generally yield the best classifier within a limited assessment budget (when only a limited number of samples labelled).

Continuous Active Learning

ALGORITHM 1: The continuous active learning (CAL) algorithm.

Step 1. Find a set of documents (a seed set) using ad-hoc search or random sampling;
Step 2. Label as "relevant" or "not relevant" each document in the initial training set of seed documents identified in step 1; Step 4. Using the classifier, rank all the unassessed documents U in the collection;
Step 5. Select the highest-scoring B documents from U;
Step 6: Assess the selected B documents, labelling each as "relevant" or "not relevant";
Step 7: Add the labelled documents to the training set;
Step 8: Repeat steps 3 through 7 until enough relevant documents have been found.

As described in Section 2.2.2, Settles states that the goal of active learning is to yield a better classifier. The experiments conducted for building a better classifier usually require a training set for training the learner and a test set to measure the effectiveness of the trained learner [Settles, 2012]. However, in the area of technology-assisted review (TAR) for eDiscovery, the problem is different [Cormack and Grossman, 2014]. TAR starts with no knowledge and continues until the majority of the relevant documents have been found. The target of TAR is to achieve high recall, while maintaining reasonable assessment effort [Carroll, 2013]. Therefore, there is no predefined training set or test set for TAR.

Cormack and Mojdeh [2009] used a method which combined active learning with interactive search and judging (ISJ) for the TREC Legal Track in 2009 [Hedin et al., 2009]. Cormack and Mojdeh called this interactive learning process.

Cormack and Mojdeh began by using ISJ to find an initial set of documents to comprise the training set for the first stage of assessment. These labelled documents set were used to build a learning model. Then, using relevance sampling, a new subset of documents were selected for labelling. The interactive learning process was repeated until enough relevant documents had been found. Cormack and Mojdeh submitted their runs and those

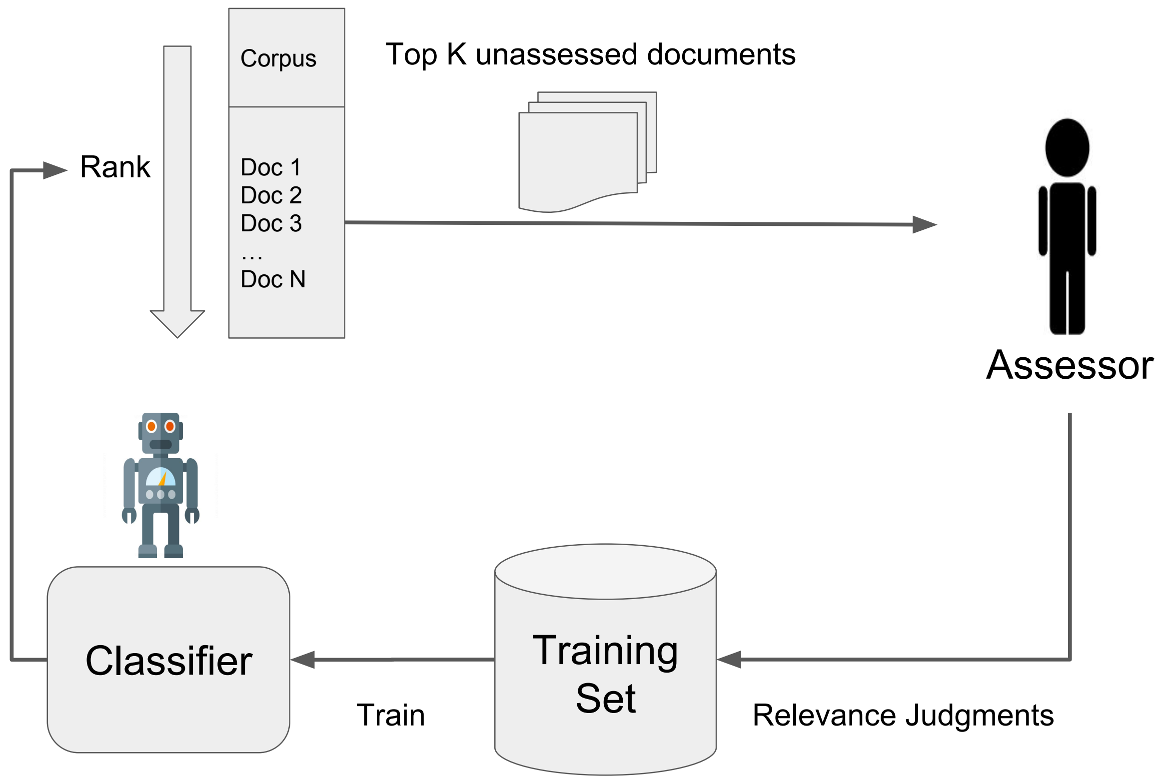

Figure 2.2: The human-in-the-loop framework of continuous active learning.

runs achieved the highest recall, precision, and F_1 in the 2009 TREC Legal Track [Hedin et al., 2009] and remained the state of the art in the TREC Legal Track [Cormack et al., 2010; Grossman et al., 2011]. The steps of the CAL algorithm are detailed in Algorithm 1. The corresponding human-in-the-loop relevance feedback framework of CAL is shown in Figure 2.2.

In a later study, Cormack and Grossman [2014] compared three different sampling strategies in active learning to determine which was able to achieve the highest recall following a fixed number of assessments. The underlying general algorithm of the three strategies was similar. First, an initial set of documents (a seed set) was selected by keyword search or random sampling and then labelled by human assessors. The labelled documents in the initial document set were then used to train a learning algorithm. The learning algorithm scored each unassessed document in the collection and ranked all the documents. Then k unassessed documents were selected and assessed by assessors. The set of all the documents labelled thus far were used to retrain the learning algorithm. The process was repeated until a certain stopping criterion was met. The three strategies developed by Cormack and Grossman [2014] differed mainly in the way they selected k documents.

- Simple Passive Learning **SPL**: k unassessed documents are randomly selected. Selection does not rely on a learning algorithm.
- Simple Active Learning **SAL**: The uncertainty sampling method applied by Lewis and Gale [1994] samples k hard-to-classify documents.
- Continuous Active Learning **CAL**: k most-likely-to-be relevant documents as ranked by the learning algorithm are selected.

By comparing these three strategies, Cormack and Grossman found that SPL and SAL achieved lower recall compared to CAL at different levels of assessment effort (number of assessments). Cormack and Grossman also found that using keyword search to build the initial training set generally yielded better results compared to using random sampling documents to select the initial set.

In a follow-up study, Cormack and Grossman [2015*a*] created an effective TAR tool (AutoTAR) which extended the CAL protocol. The steps of the AutoTAR algorithm are detailed in Algorithm 3. Cormack and Grossman showed that a topic statement can be used in place of the seed set of documents to train the initial model. AutoTAR outperformed ISJ on the TREC 6 Legal Track task of building test collections [Cormack and Grossman, 2015*a*].

Recently, for the systematic review task of CLEF 2017 eHealth lab [Kanoulas et al., 2017], Cormack and Grossman submitted a run using AutoTAR and achieved the highest recall after a given number of assessments [Cormack and Grossman, 2017]. The second-place team used a method similar to CAL, but they applied learning to rank with relevance feedback in CAL [Anagnostou et al., 2017]. In short, the applications of CAL are very effective in achieving high recall with less effort [Grossman et al., 2017].

Scalable Continuous Active Learning

Cormack and Grossman further extended the continuous active learning protocol to a new variant: scalable continuous active learning (S-CAL) [Cormack and Grossman, 2016*b*]. S-

ALGORITHM 2: The scalable continuous active learning (S-CAL) algorithm

Step 1. Find a relevant seed document using ad-hoc search or construct a synthetic relevant document from the topic description;
Step 2. Label as “relevant” the initial training set of seed documents identified in step 1;
Step 3. Draw a large uniform random sample U of size N from the document populations;
Step 4. Select a subsample size n;
Step 5. Set the initial batch size B to 1;
Step 6. Set $\hat{R}$ to 0;
Step 7. Temporarily augment the training set by adding 100 random documents from U, temporarily labelled “not relevant”;
Step 8. Construct a classifier from the training set;
Step 9. Remove the random documents added in step 7;
Step 10. Select the highest-scoring B documents from U;
Step 11. If $\hat{R} = 1$ or $B \leq n$, let $b = B$; otherwise let $b = n$;
Step 12: Draw a random subsample of size b from the B documents;
Step 13: Assess the subsample, labelling each as “relevant” or “not relevant”;
Step 14: Add the labelled subsample to the training set;
Step 15: Remove the B documents from U;
Step 16: Add $\frac{r \cdot B}{b}$ to $\hat{R}$, where r is the number of relevant documents in the subsample;
Step 17: Increase B by $\lceil \frac{B}{10} \rceil$;
Step 18: Repeat steps 7 through 17 until U is exhausted;
Step 19: Train the final classifier on all labelled documents;
Step 20: Estimate the prevalence of relevant documents $\hat{\rho} = \frac{1.05\hat{R}}{N}$.

CAL was created to reduce the labelling effort and to build a classifier with effectiveness comparable to that of running CAL over the entire collection. Cormack and Grossman state that there are several differences between S-CAL and CAL. The first difference is that S-CAL selects only a finite sample from the document collection to build the classifier. The second difference is that only a subsample of documents is selected for assessors to review in each iteration of S-CAL instead of the whole batch of documents. Once the S-CAL process has exhausted all the sampled documents, a classifier is built and used to classify the whole data collection. In contrast to CAL, in which the target is to find all the relevant documents, S-CAL is targeted at building the *best* classifier with minimum of labelling effort. This goal is similar to the goal of uncertainty sampling in the active learning protocol described in Section 2.2.2. Nevertheless, compared to uncertainty sampling, S-CAL incorporates the advantage CAL offers of using relevance sampling to select most likely relevant documents for assessment, hence reducing total review effort. Furthermore, the labelled documents in S-CAL comprise a stratified statistical sample of the entire collection, which can be used to further provide calibrated estimates of recall, precision, and prevalence.

The steps of the S-CAL algorithm are detailed in Algorithm 2. In steps 1 and 2, similar to the first step of CAL, an initial seed set of documents is derived from an ad-hoc search or from the topic description. The documents in the seed set are labelled as relevant. A uniform subset of documents U of size N are randomly sampled in step 3. Then 100 randomly selected documents from U are temporarily labelled in step 7 as non-relevant. A classifier is trained upon these labelled documents. In steps 8-9, the classifier then ranks all the unassessed documents from U. In each batch, only the B highest-scoring documents are selected, as elaborated in step 11. In Steps 12–13, a random subsample of size $b = min(B, n)$ is selected from B documents for review. In steps 14–17, the labelled subsample is added to the training set and used to retrain the classifier. The above process repeats until the documents in U are exhausted. In the meantime, S-CAL estimates the prevalence of relevant documents $\hat{\rho} = \frac{1.05\hat{R}}{N}$. $\hat{R}$ is the estimated number of relevant documents, calculated based on the labelled relevant documents r from subsample b. The value 1.05 is used to estimate $\hat{\rho}$, Cormack and Grossman [2016*b*] found a small positive bias yields a more accurate estimate. The estimated prevalence $\hat{\rho}$ can be further used to estimate precision and F_1.

2.2.3 Pooling

As mentioned in Section 2.1, one application of high-recall information retrieval is the construction of a robust test collection for evaluating different information retrieval systems [Voorhees et al., 2005]. To properly evaluate a retrieval system, each document in the test collection should have a relevance label of "relevant" or "not relevant." Therefore, the construction of a relevance label set is also a high-recall problem (finding as many relevant documents as possible) that requires a large amount of human assessment effort [Voorhees, 2001*b*].

The "pooling" method has been widely applied in TREC conferences to reduce label effort [Voorhees et al., 2005]. Pooling selects the top k ranked documents from each retrieval system to compose a judgment pool. Using this method, assessors need only to assess the documents in the pool. Documents outside of the pool are deemed to be non-relevant. In the context of the TREC conferences, the size of k usually varies from 10 to 100. The pooling method yields a stable evaluation of IR effectiveness [Voorhees, 2000].

Zobel [1998] argues that "recall (for TREC) is overestimated: it is likely that many relevant documents have not been found." Zobel estimates that relevant documents in the pool comprised only 50% to 70% of total relevant documents. Blair [2002] also pointed out that "there are no shortcuts to accurate recall estimations." Overestimated recall can be mitigated by searching all the unassessed documents for the documents that are relevant to particular topics. However, conducting such an evaluation process can be expensive and time consuming [Blair, 2002].

Subset pooling applies statistical sampling to select a subset of documents for assessment [Yilmaz and Aslam, 2006]. Subset sampling draws a uniform or stratified random sample from the pool for assessment, and then applies to it a statistical estimator to estimate the effectiveness of a run (e.g., average precision or $P@k$). Yilmaz, Kanoulas and Aslam [2008] used infAP to estimate average precision and normalized discounted cumulative gain (NDCG). In infAP, a sampling strategy is used for random sampling and an estimator estimates each stratum independently. The results collected from each stratum are combined to estimate overall results. Another family of statistical sampling methods used in subset pooling is statAP [Pavlu, 2008]. It attempts to select documents that can most accurately estimate mean average precision (MAP). It applies the Horvitz-Thompson estimator [Pavlu and Aslam, 2007; Zhang et al., 2016; Cormack et al., 2019] to form an overall estimation of the strata based on the inclusion probabilities of the relevant documents. Another method of subset pooling is minimal test collections (MTC) [Carterette et al., 2006]. MTC selects the documents that can cause the biggest change in MAP scores for different systems.

Cormack et al. [1998] used a cost-effective pooling strategy—move-to-Front pooling (MTF) —which compared well with traditional ISJ in terms of the quality of constructed relevance judgments. In contrast to the depth-pooling method in which documents are judged in arbitrary order, MTF examines the documents in the order of their probability of relevance. The submitted runs that produce more recently found relevant documents get more documents judged. Cormack and Grossman [2018] mention that MTF follows the idea of active learning by selecting the top-ranked documents from participating runs for assessment based on the relevance assessment of the previously selected documents. Cormack et al. [1998] found that the relevance judgment set generated by MTF correlated well with the gold standard qrels built using depth-100 pooling, but required much less effort.

Aslam et al. [2003] adapted a Hedge algorithm for online learning to learn which run yields better quality. Using this method, more documents are selected from high-quality runs to form the judgment pool. The results show that the Hedge algorithm can also find relevant documents with fewer judgments than the standard pooling methods [Aslam et al., 2003]. Their study also shows that applying machine learning approaches is more effective than the simple pooling methods for building relevance judgment sets.

In the TREC 2017 Common Core Track [Allan et al., 2017], the organizers of the Common Core Track formulated a pooling-based document selection problem as a multi-armed bandit problem [Losada et al., 2016; Voorhees, 2018]. Allan et al. [2017] found that using the MaxMean Bandit strategy performed as well as using MTF, while a non-stationary version of MaxMean (in which recent rewards are given more weight than older rewards) performed better than MTF.

2.3 Evaluating High-Recall Retrieval

2.3.1 Evaluation Methods

Evaluating Assessment Cost

The manual review process has been the most expensive portion of eDiscovery—estimated at around 70% of total cost in any given litigation [Mazanec, 2014]. In contrast, the collection of documents accounted for only 8% of total cost, and processing these documents accounted for about 19%[3].

A junior-level associate in an average law firm might cost US$200 per hour.[4]. In a study cited by Borden [Borden, 2010], a team of five assessors used Boolean searches and reviewed documents for 110 working hours at a rate of about 45 documents per hour. The industry average for eDiscovery is around 50 to 60 documents per hour. In short, reviewing a single document might cost more than US$3.

Assessment cost in the eDiscovery industry can measured in different dimensions, such as time, money, labour, or number of assessments [Zhang, Abualsaud, Ghelani, Smucker, Cormack and Grossman, 2018]. Zhang et al. found that there are several factors that effect the final total assessment cost:

[3] http://www.lexisnexis.com/legalnewsroom/litigation/b/e-brief/archive/2012/08/03/caution.aspx
[4] https://www.cmswire.com/cms/enterprise-cms/the-true-cost-of-ediscovery-006060.php

- Total number of relevance assessments required;
- Time spent per assessment;
- Hourly pay rate for assessors;
- Quality or experience of the assessors.

Review speed depends on several factors. Rahbariasl, Shahin [2018] studied the effects of time constraints and document excerpts on relevance judgments. In Rahbariasl's study, users were shown either full documents or document excerpts and asked to judge these documents within a time constraint of 15, 30, or 60 seconds. Rahbariasl found that time constraints can increase the judging speed rate of assessors without hurting judgment quality.

Maddalena et al. [2016] also reported that applying time constraints on assessment would not lead to loss of judgment quality. Wang and Soergel [2010] evaluated the effects of different parameters on relevance assessment. The results showed no significant difference in the assessment speed of different groups of assessors. But assessment speed did vary among individuals. In a follow-up study, Wang [2011] tested a number of influencing factors, such as document subject, length, and legibility, assessor reading skill and subject knowledge, relevance guidelines, and learning effects. The results indicated a strong correlation between perceived difficulty and assessment speed. Some difficult documents took noticeably longer for assessors to review. Document length also influenced assessors' speed. Review speed also varied significantly between different topics.

The assessments were performed by volunteers from government, law firms, legal technology firms, and law schools. In the TREC 2006 Legal Track [Oard et al., 2013], review rates for different topics varied from 12.3 to 67.5 documents per hour. The average rate was 24.7 documents per hour in 2006, 20 documents per hour in 2007, and 21.5 documents per hour in 2008 [Tomlinson et al., 2007; Oard et al., 2008].

Cost factors affecting the review effort, are important in the real eDiscovery industry. [5] Normally, legal fees are charged at an hourly rate. [6] Macaulay [2014] reported that 85% of all legal work done in Canada is still billed hourly. In the report by Hannaford-Agor [2013], senior- and, junior-level lawyers, and paralegals were surveyed about billable hourly rates. The results showed that the billing arrangements also vary dramatically from law firm to law firm, and even from client to client.

[5] https://www.lexology.com/library/detail.aspx?g=92a0f54e-d088-46fb-b9b1-57512690c6ce

[6] https://www.theglobeandmail.com/report-on-business/small-business/sb-growth/day-to-day/how-legal-fees-work/article626802/

Evaluating Retrieval Effectiveness

One common evaluation method for high-recall retrieval is to use set-based metrics, in which the effectiveness of a given retrieval system is evaluated by the intersection of retrieved results and the relevance set [Oard et al., 2010]. Retrieved results are the documents returned by the high-recall retrieval system. These documents are assessed by human assessors. The relevance set is the set of documents labelled relevant according to some gold standard (e.g., NIST assessors). This intersection has four subsets, shown in Table 2.1:

- True Positive (TP): Both retrieved and relevant.
- False Positive (FP): Retrieved but not relevant.
- True Negative (TN): Neither retrieved nor relevant.
- False Negative (FN): Relevant but not retrieved.

Table 2.1: Confusion matrix table for relevance assessment. The retrieved documents are retrieved by the system. These retrieved documents are also reviewed by assessors. The relevant documents are the documents marked relevant according to a gold standard.

		Relevant	
		1	0
Retrieved	1	TP	FP
	0	FN	TN

Commonly used set-based metrics include recall, precision, and F_1 [Oard et al., 2010]. These metrics can be derived from the four subsets of the intersection of the retrieved set and the relevance set. Recall measures the proportion of relevant documents that are retrieved, as shown in Formula 2.2. Precision measures the proportion of retrieved documents that are relevant, as shown in Formula 2.3. If one system achieves higher recall while other system achieves higher precision, it is not immediately obvious which system is superior. There is always a trade-off between precision and recall [Baeza-Yates et al., 2011]. Increasing one might lead to a decrease in the other. For instance, recall can be optimized to 1.0 by returning all the documents in the collection. Similarly, precision can be increased if only the most certainly relevant documents are returned for assessment. Achieving higher recall might generally lower precision. In order to evaluate both metrics in one measurement, F_β combines precision and recall, as shown in Equation 2.4. β is an non-negative value that can be used to assign different weights to precision and recall. A traditional

F measure is F_1 (shown in Equation 2.5), a harmonic mean of precision and recall where they are assigned equal weights. F_1 rewards results achieving both high recall and high precision, while penalizing systems that have either low recall or low precision [Büttcher et al., 2016].

$$Recall = \frac{TP}{TP + FN} \tag{2.2}$$

$$Precision = \frac{TP}{TP + FP} \tag{2.3}$$

$$F_\beta = (1 + \beta^2) \times \frac{Precision \cdot Recall}{(\beta^2 \times Precision) + Recall} \tag{2.4}$$

$$F_1 = \frac{2 \times Recall \times Precision}{Recall + Precision} \tag{2.5}$$

Other commonly used statistical evaluation metrics include sensitivity (true positive rate or TPR) and specificity (true negative rate or TNR), which are defined in Equation 2.6 and Equation 2.7, respectively. Sensitivity is equal to recall. Specificity measures the proportion of non-relevant documents that are correctly identified. Fallout (false positive rate or FPR), shown in Equation 2.8, measures the proportion of non-relevant documents that are wrongly identified as relevant. However, all these set-based metrics ignore any ranking of the retrieved documents [Oard et al., 2013].

$$\text{True Positive Rate} = \frac{TP}{TP + FN} = \text{Recall} = \text{Sensitivity} \tag{2.6}$$

$$\text{True Negative Rate} = \frac{TN}{FP + TN} = \text{Specificity} \tag{2.7}$$

$$\text{False Positive Rate} = \frac{FP}{FP + TN} = \text{Fallout} \tag{2.8}$$

Another evaluation method is to measure the total assessment effort required to achieve a certain recall, such as 75% recall [Webber and Pickens, 2013; Schieneman et al., 2013; Grossman and Cormack, 2014; Roegiest, Cormack, Clarke and Grossman, 2015]. However, the selection of a reasonable recall value is an arbitrary decision. For some datasets, where the prevalence of relevant documents is low, it is hard to decide this value [Cormack and Grossman, 2014].

Rank-sensitive metrics extend the use of set-based metrics to evaluation at different ranking depths, such as average precision (AP), precision-recall curves, and the receiver operating characteristic (ROC) curves [Oard et al., 2013]. Rank-sensitive metrics take the ordering of the documents into consideration. However, these metrics are far more complicated than set-based metrics.

$$\mathrm{AP} = \frac{1}{|R|} \cdot \sum_{k=1}^{n} P(k) \cdot r(k) \tag{2.9}$$

$$\mathrm{MAP} = \frac{1}{Q} \cdot \sum_{q=1}^{Q} \mathrm{AP}(q) \tag{2.10}$$

Average precision (AP) is a rank-sensitive metric widely used for evaluating ranked lists. For a ranked list of documents returned by a retrieval system, AP (shown in the Equation 2.9), measures the order in which the returned documents are ranked. R is the number of relevant documents in the returned ranked list. $P(k)$ is the precision at cut-off k in the ranked list. $r(k)$ is the relevance of the k-th document. $r(k) = 1$ if the k-th document is relevant. Otherwise, $r(k) = 0$. The maximum depth of AP is n. Average precision approximates the area under the uninterpolated precision-recall curve. Mean average precision (MAP) measures the mean of average precision $AP(q)$ on different queries Q, as shown in Equation 2.10.

$$\mathrm{DCG_p} = \sum_{i=1}^{p} \frac{r(i)}{\log_2(i+1)} \tag{2.11}$$

$$\mathrm{IDCG_p} = \sum_{i=1}^{|R|} \frac{r(i)}{\log_2(i+1)} \tag{2.12}$$

$$\mathrm{nDCG_p} = \frac{DCG_p}{IDCG_p} \tag{2.13}$$

Graded relevance can be used to evaluate the quality of a ranked list. Documents with higher relevance should be ranked higher than ones with lower relevance [Voorhees, 2001*a*].

Discounted cumulative gain (DCG) shown in Equation 2.11, penalizes the highly relevant documents in the lower ranks by reducing the graded relevance value $r(i)$ logarithmically proportional to the position of the result [Järvelin and Kekäläinen, 2002]. The length of the search result list varies for different queries. To evaluate multiple queries, DCG is normalized by ideal DCG (IDCG) on each query. The definition of IDCG is shown in Equation 2.12. IDCG is measured by sorting all the relevant documents according to their graded relevance, thereby producing the maximum possible DCG through position p. Then normalized DCG (NDCG) for all queries can be averaged to obtain an overall evaluation result for the ranked list returned by the retrieval system.

Evaluating the Quality of Relevance Judgment Sets

In some cases of information retrieval evaluation, several different relevance judgments sets can be generated for the same test collection [Cormack et al., 1998; Voorhees, 2000]. These relevance judgments might be the product of different retrieval systems or labelled by different assessors. In most cases, a test collection should be provided with a gold standard relevance judgment set (also called "qrels"). For instance, the test collections used in TREC conference are usually labelled by NIST assessors [Voorhees and Harman, 2005]. A corresponding gold standard qrels set is derived to evaluate different retrieval systems. Using different relevance judgments for evaluation, different retrieval systems can be given different evaluation scores (e.g., MAP scores) [Voorhees et al., 2005]. Correspondingly, ranked lists from these retrieval systems are generated. In this way, different sets of relevance judgments are compared with the gold standard qrels to measure the overall quality of these judgments. If a ranked list generated from a relevance judgments set has a high correlation with the ranked list derived from the gold standard qrels, this relevance judgment set is similar or highly correlated to the gold standard judgment set [Voorhees, 2001*a*]. In other words, the quality of this relevance judgment set is high and reliable.

Kendall's τ [Kendall, 1938; Yilmaz, Aslam and Robertson, 2008] and Spearman's rank correlation coefficient [Wackerly et al., 2007] are two widely used methods of measuring the rank correlation between different ranked lists. Kendall's τ correlation between two ranked lists is proportional to the number of pairwise swaps needed to covert one ranking list into the other. When various relevance judgments are used to compare different retrieval systems, Kendall's τ measures which relevance judgment set is relatively closer to the gold standard set. The higher Kendall's τ between the ranked list of a judgment set and the ranked list from the gold standard set, the higher the quality of the relevance judgment set. Kendall's τ has become the standard way to measure the correlation between two ranked lists. Two ranked lists are often considered effectively equivalent with Kendall's τ values at or above 0.9 [Voorhees, 2001a; Yilmaz, Aslam and Robertson, 2008]. The formula of Kendall's τ is defined in Equation 2.14.

$$\tau = \frac{C - D}{N(N-1)/2} \tag{2.14}$$

where C is the number of the concordant pairs (pairs that are ranked identically in both ranked lists) and D is the number of discordant pairs (pairs that are ranked in different order in the two ranked lists). N is the total number of ranked items in the ranked lists. There are in total $N(N-1)/2$ pairs in the ranked list. The sum number of concordant and discordant pairs is equal to the total number of pairs so that $C + D = N(N-1)/2$. If two ranked lists are identical, then $\tau = 1$, while if the two ranked lists completely disagree, $\tau = -1$.

Yilmaz, Aslam and Robertson [2008] and Carterette [2009] pointed out that Kendall's τ penalized inversions across a ranked list equally. Therefore, for the purpose of evaluation, there is no difference between documents at different ranks. However, documents ranked higher in the ranked list should be treated as more important than the documents ranked lower [Yilmaz, Aslam and Robertson, 2008]. Evaluation measures should also assign more weight to highly ranked documents.

Yilmaz, Aslam and Robertson [2008] applied a new rank correlation coefficient (*AP correlation*) to measure rank correlation between two ranked lists. AP correlation (τ_{ap}) is based on average precision and gives more weights to the top-ranked documents. The definition of τ_{ap} is as follows:

$$\tau_{ap} = \frac{2}{N-1} \cdot \sum_{i=2}^{N} \left(\frac{C(i)}{i-1} \right) - 1 \tag{2.15}$$

where $C(i)$ is the number of documents ranked above i that are correctly ranked with respect to the document at rank i according to the gold standard. N is the length of the ranked list. Thus, τ_{ap} computes the probability that each item is ranked correctly with respect to the items above the current item and averaged over all items [Yilmaz, Aslam and Robertson, 2008]. Similar to Kendall's τ, the value of τ_{ap} ranges from -1 to 1. The higher the τ_{ap} value, the higher correlation is achieved between two ranked lists. In addition, Urbano and Marrero [2017] and Smucker et al. [2012] resolved the issue of ties in AP correlation.

TREC Legal Track

The TREC Legal Track (2006–2011) [Baron et al., 2006; Tomlinson et al., 2007; Oard et al., 2008; Hedin et al., 2009; Cormack et al., 2010; Grossman et al., 2011] investigated search technology for eDiscovery. The goal of the Legal Track was to evaluate the ability of information retrieval methods to "meet the needs of the legal community for tools to help with retrieval of business records." [Baron et al., 2006] Accordingly, the tasks in the Legal Track have evolved from relatively simple ad-hoc searches to more complicated learning tasks (using relevance feedback), a reflection of the fact that the research direction (in real electronic discovery) has evolved from simple keyword searches to more complicated methods involving machine learning. Similar to all other TREC tracks, the Legal Track aimed to build reusable test collections and establish baseline results for future research.

Three main tasks were developed during the TREC Legal Track: a ranked-retrieval task, a learning task, and an interactive task [Oard et al., 2013]. The ranked retrieval task started in TREC Legal 2006 as an ad-hoc task [Baron et al., 2006]: a single-pass automatic search. In the ad-hoc task, participants were asked to produce relevant documents. However, this kind of ad-hoc task was operated without any interaction with users and based only on search queries.

In the TREC 2007 Legal Track [Tomlinson et al., 2007], a relevance feedback task was added. The goal of this relevance feedback task was to automatically discover previously unknown relevant documents based on the available relevance assessment from TREC 2006. Teams were allowed to use the positive or the negative judgments from the judged documents to improve their models. This task followed a two-pass search process in a controlled setting, by providing some labelled training samples in the first pass and improving the model in the second pass. The ad-hoc task and the relevance feedback tasks were merged as a single batch task in TREC 2009 [Hedin et al., 2009]. The batch task was basically a continuation of the relevance feedback task, but some teams could skip using the available relevance assessment and regarded the batch task as a simple ad-hoc task.

TREC 2007 Legal Track also included an interactive task. In this task, real users could iteratively reformulate their search queries based on their examination of search results. Users could perform more than one iteration of query reformulation and spend as much review effort as they desired. For their retrieval system, participants in the interactive task could use any combination of: a system they designed themselves, the Legacy Tobacco Document Library system (LTDL, a web-based system provided by the University of California, San Francisco), or the Tobacco Documents Online system (TDO, the same web-based system that was used for relevance assessment in the TREC 2006 Legal Track) [Tomlinson et al., 2007].

In the TREC 2008 Legal Track, the interactive task was completely redesigned to more accurately model real practice of eDiscovery settings [Oard et al., 2008]. A lead attorney formulated a conception to define the purpose and scope of the topic. Participants were allowed to seek clarification of the relevance of a topic from the TREC coordinators. The target of the task was to provide a binary assessment (relevant or non-relevant) for all documents in the collection. Relevance ranking could be submitted but the final deliverable was a binary classification. In case assessors made assessment errors, an appeals process was introduced to correct any possible errors.

The batch task was replaced by a learning task in TREC 2010 [Cormack et al., 2010]. Given a *seed set* of documents that had been coded as relevant or non-relevant, participants were required to estimate the probability of relevance for each remaining document in the collection. The seed set was constructed by the TREC coordinators or directly derived from the previous relevance assessment. The learning task involved a multi-stage document review process. In the beginning, the seed set documents were provided to the assessors so that they could learn what constituted relevance according to the sampled documents. Then a learning model was used to rank and estimate the relevance of each document. The learning model was either an information retrieval method or a human manual assessment process. In the human review process, different strategies could be used to review documents. Sometimes reviewers only reviewed documents above a certain rank.

Related TREC Tracks

Some other TREC Tracks are related to this thesis. Routing and filtering tasks [Robertson, 2002] were introduced in the TREC Filtering Tracks, which ran from 1995 [Lewis, 1995] to 2002 [Robertson and Soboroff, 2002]. From a stream of incoming documents, a filtering system retrieved relevant documents to meet a set of user needs (user profiles). By incorporating the relevance feedback from the users, user profiles were updated to better represent the needs of users. Another important factor in the filtering track was time. The potentially relevant documents had to be presented to the user immediately. There was no time to accumulate and rank the incoming documents. The retrieved documents were ordered by time of retrieval. The results were then evaluated based on the quality of the retrieved set.

The TREC Interactive Tracks in 2001 and 2002 also investigated the interactive search process with users [Hersh and Over, 2002; Hersh, 2002]. Searchers were allowed to use retrieval systems to find relevant information for a given subject. Effectiveness and efficiency of and user satisfaction with the searches were evaluated. Effectiveness measured completeness of the tasks and efficiency measured the time cost of each search. However, the Interactive Track did not aim to achieve high recall.

The TREC High Accuracy Retrieval from Documents (HARD) Track [Allan, 2003, 2004, 2005] was held from 2003 to 2005. The task of the HARD Track was to explore methods for improving the accuracy of retrieval systems. Participants were able to have time-limited interaction with the searchers to clarify the definition of the topic. Some additional metadata about the topic and the context of the search were also provided to participants to improve the accuracy of their searches. Passage-level relevance judgments and retrieval were incorporated into the task to focus attention only on relevant material. To evaluate the effectiveness of passage retrieval systems, the TREC 2004 HARD Track employed an adapted form of test collection, in which assessors were asked to partition each relevant document, separating the regions of text containing relevant information from the regions containing no relevant information.

The TREC Relevance Feedback Track 2008 [Buckley and Robertson, 2008] examined how the amount of relevant information could affect the performance of retrieval. The results showed that using relevance feedback consistently improved system performance. However, the effect of varying the size of the inputted relevance feedback was not consistent for different systems. With more relevant information, the performance of retrieval was not always improved.

2.3.2 Stopping Criteria for High Recall

For recall-oriented retrieval, knowing when to stop the review process is crucial, especially when the assessment budget is limited [Cormack and Grossman, 2016*a*]. The goal is to achieve high recall while minimizing the cost of labelling data. A similar scenario exists when building better classifiers [Lewis and Gale, 1994]. In many cases, the researchers aim to build a better classifier with the least labelling effort [Bagdouri et al., 2013]. A classifier is built from a set of labelled training samples (training set), and then evaluated on another set of randomly sampled documents (test set). Developers might want to add training samples to improve the effectiveness of a classifier.

Webber et al. [2013] conducted sequential testing to evaluate classifiers. The performance of a classifier can be re-examined when new samples are added to the training set or the test set or both sets. In their experiment, three different conditions were compared: variable test and fixed training; fixed test and variable training; and variable test, variable training. In each run of these three conditions, 20 randomly sampled documents were added to the corresponding set. Webber et al. [2013] assumed that the developer had set a target value of F_1. If the lower confidence interval of F_1 exceeded the target, the developer would stop the training and test the process. The results showed that, for all three conditions, the developer stopped early. The actual performance of the classifier was lower than the target performance. The fixed training and variable test scenario, in which the early stop appeared at 31.6% of the time, was the worst.

In a follow-up study, Bagdouri et al. [2013] noted that using a control test set to decide when to stop increasing the training set could introduce a biased estimate. The stopping rule (exceeding a target F_1 score) is more likely to stop training at an overestimate than at an underestimate. As an alternative, they used cross-validation to estimate the true effectiveness of a classifier. Then they used a "hold out" certification test set to certify the classifier once. Based on estimated classifier performance, the smallest size of effective certification test-set can be inferred in order to minimize total annotation cost. This model avoids sequential bias when certifying classifiers.

In the TREC Total Recall Track 2015, there was a subtask called "call your shot." Participants were allowed to indicate when they would have stopped their review to optimize different criteria (70% recall, 80% recall, and "reasonable and proportional") without actually stopping. Cormack and Grossman [2016*a*] investigated three different "when to stop" strategies for high-recall retrieval. These three strategies are defined as follows:

- *Target method*: 10 random relevant documents are chosen as the target. A retrieval method retrieves the documents without the knowledge of these 10 documents, until each document in the target set has been found.

- *Knee method*: This is a geometric method based on the shape of the gain curve (recall versus effort) derived from continuous active learning. The normal gain curve is generally convex, with high slope in the beginning, and near-zero slope once most relevant documents have been found. For a given rank r, the slope up to rank r is $slope_{<r}$ and the slope after rank r is $slope_{>r}$. The slope ratio is $\rho = \frac{slope_{<r}}{slope_{>r}}$. The knee method stops when ρ is larger than a given value. In difference to the target method, the knee method is independent of the number of relevant documents in the collection.

- *Budget method*: This variation of the knee method adjusts for low prevalence topics, which have a relatively small numbers of relevant documents. The upper bound of the review effort is determined by the Target method. The Budget method retrieves documents using CAL until the review effort exceeds the upper bound.

Cormack and Grossman's results showed that all three methods can achieve high recall (around 0.9). The Knee method achieves higher recall than the Target method while requiring much less review effort. The Budget method achieves higher recall and superior reliability compare to both the Target method and the Knee method. However, the Budget method sometimes costs more review effort than the Knee method. In addition, the Budget method provides consistently high recall at the expense of high effort for low prevalence topics.

Di Nunzio [2018] used a stopping strategy based on the geometry of two-dimensional document space. This method also applied relevance feedback to determine "when to stop" but the process is not the same as a CAL protocol. A probabilistic model was proposed based on a two-dimensional BM25 model. One dimension measures the probability of a document d that is relevant—$P(d|R)$—and the other dimension measures the probability of non-relevant—$P(d|NR)$. Both dimensions were calculated based on the BM25 scores. Two interpolated lines composed from these two-dimensional representations and different parameters divide the document space into three areas: a possibly relevant document set, a possibly non-relevant document set, and the documents in between. In each iteration, the slopes and the intercepts of the two interpolated lines are updated based on the new probabilistic model generated from relevance feedback. A fixed number of highly ranked unjudged documents between the two interpolated lines are judged. The review process stops when (1) the slope of the interpolated line equals 1; or (2) the precision of total judgments thus far is lower than 0.1.

2.4 Document Excerpt Retrieval and Assessment

2.4.1 Summary-Based Retrieval

Many retrieval methods rely on retrieval if the full document. However, a relevant document can be very long and can contain much irrelevant information. Many IR researchers have investigated using different granularities of relevant information from documents to improve ad-hoc retrieval [Bendersky and Kurland, 2010].

One common way is to extract passages from the documents and build a retrieval model based on these passages [Callan, 1994]. The relevance of these passages can capture the relevance of the full document. A document can be regarded as a set of passages, in which each passage is a contiguous sequence of text [Kim and Kim, 2004; Kaszkiel and Zobel, 2001]. The earliest research on passage retrieval dates back to the early 1990s [Salton et al., 1993; Hearst and Plaunt, 1993], and various types of passages have been defined and tested for their effectiveness in document retrieval.

The first step of passage retrieval is extracting passages from the documents. Callan [1994] classified the types of passages into three classes: discourse passage, semantic passage, and window passage. These three passage types are defined as follows:

- *discourse passage* is based on textual discourse units such as sentences, paragraphs, and sections.
- *semantic passage* is based on the subject or content of the text.
- *window passage* is based on a number of subsequent words.

Callan states that a discourse passage can be effective if the discourse boundaries are well defined by content. However, there are some problems with discourse passages. Sometimes writers of documents are not consistent when defining the discourse boundaries. In addition, poorly structured documents without passage demarcation are sometimes supplied [Kaszkiel and Zobel, 2001]. Another problem with discourse passages is the great variation in their lengths. Some passages can be very long; others can be very short.

Segmenting documents into semantic passages that correspond to a topic or subtopic is an alternative method. Many algorithms have been developed for dividing a document into semantic passages. One well-established method used with TREC data is that of Hearst [1994], known as TextTiling. It partitions document into coherent multiparagraph units. The subtopic structure of a document can be represented by this segmentation. The TextTiling algorithm splits a document into small text blocks and measures the similarities between adjacent blocks. The similarity measurement is based on word frequencies. Adjacent blocks with high similarity are merged while ones with low similarity are separated.

Segmenting based on window passages is another simple way to extract passages. It is based on the sequence of words. There are two types of window passages: overlapping windows and non-overlapping windows. An overlapping window shares some text with its adjacent window at the boundary. In contrast, non-overlapping windows are derived when documents are evenly divided into fixed-length sequences of words.

Callan [1994] compared different types of passages on four TREC 1 and 2 collections using the INQUERY retrieval system. The results showed that using fixed-length window passages was much more effective than using paragraph-based passages.

Kaszkiel and Zobel [2001] introduced an *arbitrary passage* that is defined as any sequence of words of any length starting at any point in its document. There are two types of arbitrary passages: fixed-length and variable-length. Test results show that ranking arbitrary passages can substantially improve retrieval effectiveness compared to ranking full documents. Moreover, there is no unified passage length that achieves the best effectiveness for different collections and different query sets.

After extracting passages from documents, the next step is to build a retrieval model based on these passages. Many methods have been developed to incorporate passages into retrieval models. Callan [1994] found that using the single best-scoring passage from the document to rank the collection is 20.7% better in terms of precision than ranking based on the full document. In a MultiText experiment, Cormack et al. [1997] retrieved short document passages of around 20 words in length. They used the *shortest substring ranking* method and Boolean queries to match segments. The retrieved passages were assessed as being either relevant or non-relevant. The relevance feedback of the retrieved passages were used to refine the new query terms.

Allan [1995] examined the effectiveness of using fixed-length window passages for relevance feedback. In Allan's relevance feedback experiment, a query was modified based on the relevance of retrieved documents. The terms that appeared frequently in the relevant documents were extracted, weighted, and added to the new query. The results showed that using fixed-length passages for relevance feedback was consistently more effective than using full documents.

Liu and Croft [2002] applied a language-modelling approach to overlapping window passages. Their results showed that in the language-modelling context, ranking documents by their best passage score achieves effectiveness comparable to ranking based on the full document. Bendersky and Kurland [2010] combined passage-level language models with full-document language models and assigned different weights to each. They found the combination of the two models is more effective than the standard passage-relevance model or document relevance model alone. Krikon and Kurland [2011] further integrated document-based, passage-based, and cluster-based information to improve the ranking method.

Salton et al. [1993] developed a vector-space passage retrieval model on an encyclopedia collection. Their results revealed that paragraph and section retrieval yield a significant improvement on mean average precision over full-document retrieval.

Wang and Si [2008] developed two different discriminative probabilistic models for passage-based retrieval. The independent passage model was based on a language modelling. In contrast to previous work that considered only the best-matching passages in each document for ranking, their independent passage model measured each individual passage independently. The other model was a correlated-passage model, based on a TF-IDF vector space. They also developed a combination of both models that significantly improved the effectiveness of document retrieval and passage retrieval over either model separately.

Ai et al. [2018] developed a neural passage model. Their neural passage model uses a convolutional neural networks and an aggregated-passage model with a fusion network. Compared to Bendersky and Kurland's state-of-the-art passage-language model, their neural-passage model made a significant improvement in ranking performance.

Yulianti et al. [2018] used external community question answering (CQA) data to improve passage ranking. They used a CQA service to obtain the best answers to a given query. Then they derived several passage quality features based on the overlap between document passages and CQA answers. The sequential dependency model (SQM) [Metzler and Croft, 2005] was integrated with passage-quality features to rank the documents. They found that incorporating answer-passage quality features significantly improved the retrieval model.

Yang et al. [2019] used BERT to measure the relevance degree between query and each sentence in a document. The sentence-level relevance scores were interpolated with the document-level relevance scores to help rerank documents. The results were significantly improved on Robust04 dataset compared to other neural methods.

2.4.2 Evaluation of Summary Assessment

The TIPSTER Text Summarization Evaluation Conference (SUMMAC) was the first large-scale conference for evaluating automatic text summarization systems [Mani et al., 2002]. The *ad-hoc* evaluation task asked participants to generate indicative summaries according to the full context of documents. The relevance of the document summaries was measured in terms of the relevance of the full text.

In total 16 systems participated for the SUMMAC evaluation. In each task, participant submitted a fixed-length summary $S_{10\%}$ (10% of full text) and a summary without length limitation S_{var} for evaluation. For the ad-hoc task, 20 TREC topics were selected and the documents evaluated in the data collection were all newspapers. The relevance judgments on submitted documents and summaries were judged by 51 professional information analysts. Each of these information analysts took from 16 to 21 hours to assess documents for different tasks.

The evaluation results showed that "automatic text summarization is very effective in relevance assessment tasks on newspaper articles." In the ad-hoc task, the summaries with varied length S_{var} were compressed to 17% of full text length on average. Judging S_{var} took only 33.12 seconds while full-text assessment took 58.89 seconds. More importantly, the accuracy of judging summaries was only 4% less than that of judging full documents, not a significant difference. In short, the length of different summaries ranged from 10% to 17%

of the length of the full document but were able to reduce assessment time by 40% to 43% without impairing accuracy significantly. Mani et al. [2002] also found False Negatives (FNs) to be the main reason for the loss of accuracy in summarization. This means that document summaries were missing information relevant to the topic leading assessors to misjudge the relevant document as non-relevant. The disagreements of assessments between assessors could be another factor affecting the results.

Smucker and Jethani [2010] confirmed the effectiveness and efficiency of using query-biased document summaries for relevance assessment in a controlled user study. Smucker and Jethani reported that summaries of news articles containing 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 for these short summaries ranged from 62% to 72% depending on the quality of the ranked list and that the accuracy of judging full documents was from 75% to 76%.

Tombros and Sanderson [1998] found that users can judge the relevance of documents from summaries as accurately as they can from full documents. In their study, users were required to judge as many relevant documents as possible within 5 minutes. They compared the users' judgments on topic-related and generic summaries. In their study setting, users were allowed to refer to the content of full document for a given document summary. They found the average time to judge document summaries was only 24 seconds whereas the average time to assess the full document was 61 seconds. Showing only document summaries helped assessors retrieve nearly 75% of the relevant documents, as identified by those who had access to the full document.

In the same paper, Tombros and Sanderson [1998] found that reviewers could use query-biased summaries for relevance judgments without needing to refer to the full text of the document. In a subsequent study, Sanderson [1998] found that "the results reveal that reviewers can judge the relevance of documents from their summary almost as accurately as if they had had access to the document's full text." An assessor took, on average, 24 seconds to assess each summary and 61 seconds to assess each full document.

Chapter 3

Participation in the Total Recall Track 2015

3.1 Task of the Total Recall Track 2015

The purpose for the Total Recall track 2015 is to evaluate methods to achieve very high recall through a controlled simulation study –with a human assessor in the loop [Roegiest, Cormack, Grossman and Clarke, 2015]. More detailedly, the task for the participants to tackle in Total Recall Track is quite simple. Give a simple information need, similar to the simple query words in the ad-hoc and web search, identify the documents in a corpus, one at a time, such that, to find nearly all the relevant ones. Immediately after each document is identified, its relevance or non-relevance is given by the simulated assessor for relevance feedback. The set of documents returned so far can be used to evaluate the effectiveness of a high-recall method. Rank-based and set-based evaluation metrics were used to evaluate the results.

The organizers of Total Recall Track provided datasets, topics, and automated relevance assessment via a web server. The participants can download the datasets and topics from the web server. They can also submit documents for relevance assessment via the same online web server. The web server simulated the role of a human-in-the-loop assessors in real time. The retrieval method can either be fully automated (automatic) or semi-automated (manual). A fully automatic solution would contact the web server and conduct the task without any human intervention. In contrast, manual method allowed human intervention, hence the participants were allowed to manually select documents and judge the relevance of selected documents.

In total eight collections were used in Total Recall Track 2015 to evaluate different runs. Three of them are for "Practice", three for "At Home" participation, and two for "Sandbox". Practice test collections were severed as test runs for participants to test their model before the final submissions. The At Home and Sandbox collections were used for official evaluation. The Practice and At Home runs were conducted via the open web. The participants can run their own systems and connect to the server using an open address. While for the Sandbox runs, participants were required to encapsulate their autonomous solutions into VirtualBox virtual machines. The virtual machine was a Web-isolated platform hosting the data collection. The solutions provided by participants should contact the Web server automatically and conduct the task without any human intervention. The purpose of sandbox run is preventing the disclosure of sensitive test data to participants.

In addition, the Total Recall also introduced another subtask–"call your shot", in which participants were required to indicate when they thought a "reasonable" number of relevant documents had been achieved. In general, the prevalence of relevant documents decreases as the review effort increases. After reaching a certain review effort, the benefit of finding a larger number of relevant document cannot compensate the increase of the review cost to find them. The point of call your shot is to indicate when the participants would recommend terminating the review process since further effort would be disproportionate. When submitting runs for evaluation, the participants were not required to actually stop further assessment. They just need to indicate the point, hence the recall, precision, and F_1 achieved at that point can be measured.

Our goal for us to attend the Total Recall Track 2015 was to investigate the methods to achieve high recall with the help of simulated relevance feedback. We tried to achieve higher recall or find a larger number of relevant documents within limited amount of assessments. The Total Recall Track organizer provided a baseline model implementation (BMI) based on the Continuous Active Learning for comparison. As mentioned in Section 2.2.2, we find that the CAL based approaches achieved high effectiveness in the Legal Track [Cormack et al., 2010]. In this paper, we decided to modify BMI and incorporate more features into it. We first implemented our model and tested it on the Practice collections. Then we applied our best performed model on At Home and Sandbox collections and submitted corresponding runs.

3.2 Test Collections

The TREC Total Recall Track 2015 used a total of eight test collections [Roegiest, Cormack, Grossman and Clarke, 2015; Grossman et al., 2016]. Three of them are just practice test collections. One of them is the *20 Newsgroups* Dataset [1], consisting of 18,828 documents from each of 20 newsgroups. Three of the newsgroup subject categories: "space", "hockey", and "baseball" were used as practice topics in the test practice collection. The second practice test collection is the *Reuters-21578* Test Collection [2], consisting of 21,578 newswire documents. Four of the subject categories: "acquisitions", "Deutsche Mark", "groundnut", and "livestock" were used. The third practice dataset is the *Enron* Dataset used by the TREC 2009 Legal Track. The track organizers used a version of the University of Waterloo from its participation in TREC 2009. It was a corpus of 723,537 documents after excluding the vacuous documents. Two of the topics from the TREC 2009 Legal Track "Fantasy Football" and "Prepay Transactions" were used for this biggest practice collection. The relevance assessment on Enron dataset were derived from the judgments of University of Waterloo team and the official TREC assessments.

For the rest five test collections, three for "At-Home" participation, and two for "Sandbox" participation.

Athome1—The Jeb Bush Emails [3], consisting of 290,099 email messages from Jeb Bush's eight-year tenure as the Governor of Florida.

Athome2—The Illicit Goods dataset was collected for the TREC 2015 Dynamic Domain Track. 465,147 documents were collected from Blackhat World [4] and Hack Forum [5]. For the Athome2 test collection, the track organizers used ten topics that were composed and partially assessed by NIST assessors from the Dynamic Domain Track [Yang and Frank, 2016].

Athome3—The Local Politics dataset—was also collected for the TREC 2015 Dynamic Domain Track. 902,434 articles were collected from news sources in the northwestern United States and southwestern Canada. Similar to Athome2, 10 topics were composed and partially assessed by NIST assessors for use by the Dynamic Domain Track 2015.

For the Sandbox runs, two new datasets were used. One of them was the Kaine Email

[1] http://qwone.com/ jason/20Newsgroups/

[2] http://www.daviddlewis.com/resources/testcollections/reuters21578/

[3] http://jebemails.com/home

[4] http://www.blackhatworld.com/

[5] http://hackforums.net/

Collection at the Library of Virginia [6]. Among the $1.3M$ email messages from Kaine's eight-year tenure as Governor of Virginia, $401,953$ that had previously been labelled by the Library archivist were used. Four categories were used as a topic in the Kaine test collection. The runs themselves were executed on an isolated computer installed at the Library of Virginia and operated by the Library of Virginia staff.

The MIMIC II Clinical Dataset [7], consisting of anonymized records for $31,538$ patient visits to an Intensive Care Unit. The textual record for each patient was consisted of one or more nurses' notes, radiology reports, and discharge summaries. The runs were executed on an isolated computer at the University of Waterloo and operated by the Track Coordinators.

Collection	Type	Description	# Docs	# Topics	# Rel (R)
20 Newsgroups	Practice	20ng	18,828	3	987-999
Reuters	Practice	oldreut Newswire	21,578	4	10-2,448
Enron	Practice	TREC09 Legal Track	723,537	2	2293-7,798
Athome1	Athome	Jeb Bush public email	290,000	10	227-17,135
Athome2	Athome	Hacker forums	465,147	10	179-9,517
Athome3	Athome	Local news	902,434	10	23-2,094
Kaine	Sandbox	Tim Kaine non-public email	401,953	4	14,341-166,118
MIMIC II	Sandbox	MIMIC II Clinical Database	31,538	19	180-19,182

Table 3.1: The statistics of practice, Athome, and Sandbox test collections.

For building the relevance label set for these test collections, the track coordinators assessed the documents using the continuous active learning method of Cormack and Mojdeh [2009] to identify as many of the relevant documents for each topic as reasonably possible. They used ISJ and CAL with two different feature engineering techniques and two different base classifiers to identify and label substantially all relevant documents prior to running the task. These relevance labels were used to simulate assessors feedback and to evaluate the retrieval results of different high-recall retrieval systems.

3.3 Baseline Model Implementation

State-of-the-art high-recall information retrieval method is the Baseline Model Implementation (BMI) method used in the Total Recall Track 2015 [Roegiest, Cormack, Grossman

[6]http://www.virginiamemory.com/collections/kaine/under-the-hood
[7]https://physionet.org/mimic2/

and Clarke, 2015; Grossman et al., 2016]. BMI was supplied to Total Recall Track participants in advance, and used as the baseline method for comparison. AutoTAR is a version of CAL and BMI is an implemented version of AutoTAR, which is effectively a relevance feedback method.

The details of AutoTAR algorithm are listed in Algorithm 3. In AutoTAR, an initial classifier is built from topic description. The initial classifier ranks all the documents in the collection and selects the highest-scoring unassessed document for assessors to label. The relevance feedback of the judged documents are fed back to retrain the classifier. The whole documents collection is then re-ranked and the next most likely to be relevant and unassessed document is selected for assessor to review. The above process is repeated until a sufficient number of relevant documents have been found.

ALGORITHM 3: The autonomous TAR (AutoTAR) algorithm. BMI is implemented based on AutoTAR.

Step 1. Treat the topic statement as a relevant document and add this document into the training set;
Step 2. Set the initial batch size B to 1;
Step 3. Temporarily augment the training set by adding 100 random documents from the collection, temporarily labelled "non-relevant";
Step 4. Train a logistic regression classifier using the training set;
Step 5. Remove the random documents added in Step 3 from the training set;
Step 6. Select the highest-scoring B documents from the not reviewed documents;
Step 7. Append the selected B documents to system output. The system output records the list of documents that have been selected by the classifier and labelled by the assessors;
Step 8. Review the selected B documents, coding each as "relevant" or "non-relevant";
Step 9. Add the labelled B documents into the training set;
Step 10. Increase B by $\lceil \frac{B}{10} \rceil$;
Step 11. Repeat steps 3 through 10 until a sufficient number of relevant documents have been reviewed.

Previous research investigated three different strategies to construct the initial training set for building the first classifier [Cormack and Grossman, 2015*a*]. The method called "Auto-BM25" selected the top-ranked documents given by BM25 retrieval method [Jones et al., 2000]. Another one labelled "Auto-Syn" used a synthetic document created from the query as the initial training document. The last one was called "Auto-Rand" and simply selected a random relevant document at the outset. According to the test results on TREC 2009 Legal Track topics and TREC 2002 Filtering Track topics, Cormack and Grossman found that the Auto-Syn generated better results than the other two methods.

3.4 Modified Baseline Model Implementation

We participated in Total Recall Track 2015, our proposed method followed the procedure of CAL algorithm and was modified on the basis of the BMI implementation. We tried to improve the BMI model from the following perspective.

First, we tried different approaches for selecting the seed documents to train the initial classifier. Previous research only considered some simple strategies (e.g., random sampling, BM25, and using query as synthetic document) [Zhang, Rao, Lin and Smucker, 2017]. More complicated strategies were not applied. We assumed that improving the selection of seed documents can help train a better initial classifier. Furthermore, a superior initial classifier can help the overall CAL process find a larger number of relevant documents within limited effort. In this thesis, we implemented and compared different strategies to explore the benefits of bringing diversity into the selection of seed documents.

Second, the BMI tested only two types of features to represent the documents. They are TF-IDF word-based features and binary byte 4-gram features respectively. In this thesis, we explored more types of representation for document features.

Third, a logistic regression classifier is used to classify, score, and rank all documents in BMI. Cormack and Grossman [2015*a*] also applied Support Vector Machine (SVM) in AutoTAR and compared SVM with logistic regression. They found logistic regression was able to yield slightly superior results. In this thesis, we tried more types of classifiers and compared their performance with respect to achieving high recall.

Fourth, query expansion is widely used when relevance feedback is available. In this thesis, we also tried to incorporate query expansion method into CAL procedure. We hypothesis that query expansion can bring diversity into CAL model hence to help retrieve a larger number of relevant documents given relevance feedback. We next describe our modifications on BMI from these perspectives.

3.4.1 Clustering-Based Seed Documents Selection

For selecting the seed documents to construct the initial classifier, we first used clustering method to group some candidate relevant documents. In this way, similar documents can be clustered and grouped together. Different clusters represent different types of documents. The documents from different clusters can represent different concepts or subtopics of the topic. Furthermore, how to select documents from different clusters to represent seed documents set is an interesting problem to explore. This kind of selection problem is a trade-off between exploration and exploitation. On the one hand, we want to retrieve a larger number of relevant documents from the cluster which has a high prevalence of relevant documents. On the other hand, we also want to bring diversity into CAL model so that different types of relevant documents can be explored. Thus different aspects of relevance can be used to train the CAL model. In this thesis, we tried four different strategies and compared their performance on selecting relevant seed documents.

In our experiment, top D retrieved documents ranked by BM25 are selected as candidate relevant documents for clustering. The probabilistic latent semantic analysis is applied to learn the conceptual correlations of these retrieved documents [Deerwester et al., 1990]. The similarity between documents is measured based on the entropy weighted term-document matrix [Hofmann, 1999]. Clustering is operated upon the features generated from entropy weighting LSI, which is an effective way to group documents based on their conceptual similarity.

Sampling method

This sampling approach uses a similar idea from the multi-armed bandit algorithm [Vermorel and Mohri, 2005]. The multi-armed bandit studies a trade-off problem between exploitation and exploration. The exploration represents the attempts to acquire new knowledge for the relevant documents. In contrast, the exploitation tries to optimize the decision based on existing knowledge (existing relevant documents). In our experiment, we first cluster documents into different groups. The different clusters represent different knowledge about the topic. In order to maximize the gain of finding more knowledge about the topic and also build a high quality initial classifier, we try to explore and exploit different clusters of documents. The detailed steps of this sampling approach are listed as follows:

1. Select top D BM25 scoring documents as candidate documents for seed selection;
2. Group these documents into K $(1 < K < D)$ clusters, where K is predefined;
3. Select one document from each cluster and label the document based on relevance feedback from the assessor;

4. Initialize cluster-specific counter t to 1 and cluster-specific reward r, based on judgments in step 3;

5. Select the next cluster v based on the following conditions:

   (a) the cluster v has at least one unlabelled document;

   (b) the cluster v has the maximum value of $\frac{r_v}{t_v} + \sqrt{\frac{\mu \log(\sum_{c=1}^{|C|} t_c)}{t_v}}$ among cluster set C, where μ is a constant variable;

6. Randomly pick one unlabelled document from cluster v and label the document based on relevance feedback from assessor, which is the same as step 3;

7. Update cluster-specific reward r_{v}, based on the judgment in step 6 and increase the counter t_{v} by 1;

8. Repeat steps 5 to 7 until all D documents are labelled;

Graph method

A graph can be used to represent the relationship between documents. This method is called as graph approach. The main steps of this graph approach are:

1. Select top D BM25 scoring documents as candidate documents for seed selection;

2. Build a weighted graph for these documents based on the clustering results;

3. Select one unlabelled document d from the documents graph and label it based on the relevance feedback from assessors;

4. If the selected document d is:

   (a) relevant, increase the weights of all the edges connected to the document d;

   (b) non-relevant, decrease the weights of all the edges connected to the document d;

5. Go to step 3 until all D documents are labelled.

In the second step of the graph approach, the weighted graph is constructed by:

1. Each document is considered as a node in the graph;

2. We run K-means T times to cluster these documents;

3. The weight $w_{i,j}$ of a undirected edge between document i and document j is $w_{i,j} = \sum_{t=1}^{T} \mathrm{I}_t(i,j)$, where $\mathrm{I}_t(\cdot)$ is an indicator function and $\mathrm{I}_t(i,j) = 1$ denotes document i and document j are in the same cluster based on the t-th clustering result of K-means.

In the third step of the graph approach, a greedy method is used to select the next reviewed document from each cluster. The detailed steps of the third step is as follows:

1. Initialize a priority queue for documents.
2. If the priority queue is:
   (a) empty, select a document i with the highest BM25 score among all unlabelled documents;
   (b) not empty, select a document i with highest weighted score from the queue;
3. Label document i based on the relevance feedback from assessor;
4. If document i is:
   (a) relevant, set $\delta(i) = 1$;
   (b) non-relevant, set $\delta(i) = -1$;
5. For each unlabelled document j connected to document i:
   (a) if document j is not in the queue and the document i is relevant, insert document j into the queue with weighted score $w_{i,j} + \delta(i)$;
   (b) if document j is in the queue, increase the weighted score of document j in the queue by $\delta(i)$.

Jumping method

Different from the sampling method and graph method which require complicated selection algorithms, we also introduced a simple method to select documents from different clusters. This method is called as jumping method. It jumps among different documents clusters and greedily selects the highest-scoring document from clusters.

Here are the detailed steps. We select the document d_i^n (i-th document in n-th cluster) with the highest BM25 score for judging. Assuming document d_i^n is coded as relevant, then a document d_j^n with second-highest score is selected from the same cluster c_n; otherwise, we select the highest-scored document d_k^m from other cluster c_m. This procedure continues until a certain number of relevant documents have been found.

Weighted method

Similar to the jumping method, we implemented another method called as weighted method. The difference between the jumping method and the weighted method is that an initial weight value of 1.0 is assigned to each cluster c_n. If a document d_i^n in cluster c_n is labelled as non-relevant, the corresponding cluster c_n weight is penalized by a heuristic factor which is smaller than 1.0. Then the document d_j^m in the cluster c_m with the highest factorized weight is select for review in the next iteration.

Comparison of different seed documents selection methods

We compared four proposed seed documents selection methods on the test collections provided by Total Recall Track 2015. Table 3.2 shows the comparison results on the seven topics (tr0-tr6) on the Reuters and 20ng practice corpora provided by Total Recall organizers[8].

As mentioned in Section 3.4.1, the first step of our modified BMI method is to select top D BM25 scoring documents for clustering. In our experiment, we set $D = 100$. For evaluating the effectiveness of different selection methods, we compare the number of relevant documents retrieved by each method at the maximum number of 50 assessments. In other words, the method which is able to retrieve a larger number of relevant documents by selecting 50 documents from top 100 documents is regarded to be a better selection method.

It should be noticed that the BM25 algorithm returns only 7 documents for topic tr2 and 63 documents for topic tr3. Therefore, for topic tr2, we just skipped the clustering algorithm and sent all 7 documents for relevance assessment. For topic tr3, we only selected top 32 ranked documents out of the 63 BM25 returned documents for review. According to the number of relevant documents found by each method shown in Table 3.2, the graph method and sampling method achieve slightly superior results compared to other methods

[8] http://quaid.uwaterloo.ca:33333/#/doc

on the majority of topics. All these methods yielded very similar results. No method outperformed the other methods significantly.

Table 3.2: Comparison of seed documents selection methods from top 100 BM25 scoring documents. This tables shows the number of relevant documents found by each method using at maximum 50 times of assessments.

Methods	tr0	tr1	tr2	tr3	tr4	tr5	tr6
Sampling	45	1	2	14	**48**	49	**46**
Graph	**47**	**2**	2	**15**	45	**50**	45
Jumping	46	1	2	10	47	49	40
Weighted	46	0	2	10	47	49	42

3.4.2 Feature Engineering

The Total Recall Track BMI implementation utilized two different features engineering methods to represent document. The document features are vectorized and used as input to train a logistic regression classifier. One of these features is TF-IDF word-based feature and the other one is binary byte 4-gram feature (combinations of 4 sequential characters). In TREC 2007 spam filtering competition [Cormack, 2007; Sculley and Cormack, 2009], the best performing spam classifier used a binary 4-bytes string feature space. According to our experiment results on practice datasets, using TF-IDF word-based features performed better than using binary byte 4-gram feature on the majority of topics. However, we found that using binary byte 4-gram features achieved slightly higher precision especially when the query was a complete sentence or composed by multiple words. For keyword query, TF-IDF is effective enough to train an effective linear classifier.

In Cormack and Grossman [2015*a*]'s AutoTAR implementation, they also mentioned the comparison between these two features. They explained their preference on TF-IDF word-based feature over binary byte 4-gram. Using TF-IDF features for training a logistic regression classifier via the ML package Sofia-ML [9] or Vowpal Wabbit [10] was able to achieve the best performance compared to using other feature engineering methods.

With the options of using two different feature engineering methods for building the machine learning classifier separately, an intuitive idea is to combine or fuse the results generated from these different methods. Different results generated from different classifiers

[9] https://code.google.com/archive/p/sofia-ml/
[10] https://github.com/JohnLangford/vowpal_wabbit

might be able to cover different aspects of relevant information, thus yield better results. We apply the Reciprocal Rank Fusion (RRF) to fuse the ranked lists R generated from different classifiers over the set of returned documents D [Cormack et al., 2009]:

$$RRF_{score}(d \in D) = \sum_{r \in R} \frac{1}{k + r(d)} \tag{3.1}$$

where $r(d)$ is the rank of document d in the ranked list r and we set $k = 60$. RRF fusion ensures that the highly-ranked documents from different ranked lists are assigned more weights. In addition, the lower-ranked documents are not ignored. In some cases, a document might have a low ranking in one ranked list but have a higher ranking in the other ranked list. RRF method tries to balance the rankings of a document from different ranked lists. We tried the RRF method by fusing the ranked lists returned from different classifiers. The results of RRF fusion was not as promising as we expected. RRF generated almost the same results as using TF-IDF feature, but cost much more computation resources.

In our experiment, we also used entropy g_i as a feature to represent the relative frequency of term i within the entire collection of documents. This feature is called as entropy-weighted LSI. It can be defined as:

$$g_i = 1 + \sum_j \frac{p_{ij} \log p_{ij}}{\log n}, \text{where } p_{ij} = \frac{tf_{ij}}{gf_i} \tag{3.2}$$

where n is the number of documents in the corpus, gf_i denotes the occurrences of term i in the whole corpus and tf_{ij} indicates the term frequency of term i in the document j. LSI performs a Singular Value Decomposition (SVD) on the matrix and reduces the high dimensional sparse term-document matrix into a compact matrix with a fixed size [Deerwester et al., 1990]. The LSI based features are also used for documents clustering in seed documents selection described in Section 3.4.1. Although the LSI based features extracted the key information from each document and performed effectively on documents clustering, we found that these LSI based features were not able to train a more precise classifier compared to using TF-IDF features. We expect that the dimension reduction on document features would result in a information loss to some extent. And this kind of information loss would not bring benefits to distinguish relevant documents from non-relevant documents.

In addition, we notice that there is one query—"Deutsche Mark" from the practice test collection provided by Total Recall Track 2015. These two terms in this query together contribute to one meaning. It is a phrases representing the official currency of West Germany from 1948 to 1990. If we only consider unigram feature for this query, we find that many documents with the single term "Mark" can be retrieved. Since "Mark" is just a personal name in many cases. It would be hard to retrieve the documents in which terms "Deutsche" and "Mark" co-occurred. In order to improve the effectiveness of retrieving such documents, n-grams features can be used to represent document. For the query "Deutsche Mark", bigram feature can concatenate these two words together. For the documents containing both these two words, it would be much easier to locate those documents by checking the bigram terms compared to unigram word.

In our experiments, we tried to build TF-IDF features for each document over bigram terms, trigram terms, the union of unigram terms and bigram terms, and also the union of unigram, bigram and trigram. We found that the results of merging unigram features with bigram features or trigram features were usually superior than that of just using unigram features. Moreover, the union of unigram and bigrams features and the union of unigram, bigram, and trigram almost break even on different topics. However, with more features added into classifier, the number of dimensions of feature vectors increase accordingly. It takes more time for the classifier to train classifier and rank documents. Taking the computation cost into consideration, we only applied the union of unigram and bigram as the final feature set.

Besides using n-grams as features for building document classifier, query terms are also reordered to form different query pairs for composing the synthetic seed document. For example, "Deutsche Mark" is regarded as two independent terms in the BMI. The TF-IDF values of "Deutsche" and "Mark" are calculated separately in order to compose synthetic document for initializing the seed documents set. In our model, bigram pairs are composed directly from query terms regardless of the terms' relative positions. For the query "Deutsche Mark", we have four candidate word pairs for composing the synthetic documents. These pairs are "Deutsche", "Mark", "Deutsche Mark", and "Mark Deutsche" respectively. If the word pair does not exist in the vocabulary list (we only consider the terms having document frequency *df* larger than 1), this word pair would be removed from candidate query pairs. In this case, "Mark Deutsche" is removed from synthetic seed documents due to its sparsity. Therefore, the corresponding SVM-light sparse data format features of this new synthetic seed document is:

$$
\begin{aligned}
Relevance : 1 & \\
& \{ \\
& Deutsch : w_1; \\
& Mark : w_2; \\
& Deutsch\ \ Mark : w_3 \\
& \}
\end{aligned}
$$

where w_i corresponds to the TF-IDF value of i-th word. After normalization, the square sum of all the w_i in one synthetic document is normalized to 1.0 in order to make the weights of this feature vector consistent with other documents.

In our modified BMI model, we not only incorporated different feature engineering methods, but also tried different types of document classifiers. The iterations of relevance feedback in continuous active learning process try to improve the effectiveness (the ability of relevance ranking) of classifier continuously. On the one hand, the training set gets more labelled documents as the number of relevance feedback iterations increases. The quality of classifier usually gets improved with more labelled documents added into the training set. On the other hand, due to the low prevalence of relevant documents in the corpus, it usually becomes harder and harder to retrieve relevant documents especially when the prevalence of relevant documents drops. In other words, if a classifier becomes more precise and is able to find relevant document more effectively, it can help the continuous active learning process.

The BMI model that was severed as the baseline method in Total Recall Track used a logistic regression classifier to classify and score documents. The logistic regression model is efficient to train and make predictions on a large document set. In the CAL process where the classifier needs to be retrained after each relevance feedback iteration, the logistic regression classifier is suitable to be applied due to its efficiency. In our experiment, we tried to find a more effective classifier to replace the current logistic regression classifier.

We tried the Gaussian kernel support vector machine (RBF kernel SVM), which was able to fit the maximum-margin hyperplane in a transformed high-dimensional feature space. We assumed that if the maximum-margin hyperplane can be measured precisely, the relevant documents can be retrieved more easily. For the soft margin parameter C and γ used in the RBF-kernel SVM, we used grid search with exponentially growing sequences of C and γ to select the best combination of these two parameters. For example, $C \in \{2^{-5}, 2^{-3}, ..., 2^{13}, 2^{15}\}$; $\gamma \in \{2^{-15}, 2^{-13}, ..., 2^{3}, 2^{5}\}$. Five-fold cross validation was operated in each iteration to pick up the best combination of parameters. According to our comparison results, we found that RBF-kernel SVM was more likely to overfit the imbalanced data. In addition, RBF-kernel SVM took more computation cost and spent around 5 times training time compared with linear classifier (e.g., logistic regression or linear SVM). In addition, we also tried some other classifiers, such as Stochastic Gradient Descent (SGD) linear SVM, random forest, XGBoosting, and Naive Bayes classifier from different machine learning packages. None of these methods can significantly beat logistic regression on the Practice collections. Therefore, we keep using logistic regression classifier with the same hyperparameters as BMI in our experiments. The detailed implementation of these methods can be found [11].

3.4.3 Query Expansion

In our experiment, we also tried to apply a query expansion method to identify potentially relevant documents in each iteration of CAL. Given the relevant documents and non-relevant documents already labelled by assessors in previous iterations, informative terms can be extracted and used to expand query terms. The top ranked documents retrieved according to an expanded query are considered as potentially relevant documents to be judged. We adapted simple mixture method [Zhai and Lafferty, 2001] to expand the query. For query expansion, we want to extract the most informative terms from relevant documents. However, not all terms in the relevant documents are informative. For simple mixture method, a background model is used to represent non-informative terms. Simple mixture method assumes that informative terms in relevant documents can be generated as follows:

1. Given two models θ_0 and θ_1. θ_0 is a model in which informative terms to be estimated. θ_1 is a known background model;
2. Given a mixing coefficient, $\overrightarrow{\pi} = (1 - \pi, \pi)$;
3. For the j-th term in the i-th relevant document:
   (a) Firstly, independently generate a latent model indicator, $z_{ji} \sim \text{Bernoulli}(z|\overrightarrow{\pi})$;
   (b) Then, independently generate a term, $w_{ji} \sim d(w|\theta_{z_{ji}})$;

where π is given (e.g., 0.9), and $d(\cdot)$ is a family of term distributions.

In this paper, the bag-of-word assumption is used and multinomial distributions are used as term distributions. Therefore, in our experiment, $d(\cdot)$ is the family of multinomial distribution. Given a corpus and non-relevant documents obtained from human assessments, we use maximum likelihood estimation (MLE) to estimate a corpus model, θ_{corpus}, and an non-relevant model, $\theta_{\text{non-relevant}}$, respectively. The background model θ_1 used in this paper is:

$$d(w|\theta_1) = 0.5 \cdot d(w|\theta_{\text{corpus}}) + 0.5 \cdot d(w|\theta_{\text{non-relevant}}) \tag{3.3}$$

The inference process for SM [Zhai and Lafferty, 2001] is calculated as follows:

[11] https://bitbucket.org/HaotianZHANG/uwtotalrecall

At k-th iteration for SM,

$$\eta^{(k)}(w) = \frac{(1-\pi)d^{(k)}(w|\theta_0)}{(1-\pi)d^{(k)}(w|\theta_0) + \pi d(w|\theta_1)} \tag{3.4}$$

$$d^{(k+1)}(w|\theta_0) = \frac{\sum_i tf_i(w)\eta^{(k)}(w)}{\sum_{w' \in \text{voc}} \sum_i tf_i(w')\eta^{(k)}(w')} \tag{3.5}$$

where "voc" denotes the vocabulary list for all the terms appeared in the corpus and $tf_i(w)$ represents the raw term frequency of w in the i-th relevant document.

Once the model θ_0 is estimated, we used top K ranked terms in the model to expand a query. In this paper, given an expanded query, we use the Kullback-Leibler divergence (KL) algorithm to rank unassessed documents [Kullback and Leibler, 1951; Lafferty and Zhai, 2001]. w are the words in the expanded query. Top ranked documents are considered as potential relevant documents to be judged. The KL ranking algorithm uses the KL divergence, which measures the term distribution difference between a query q and a document d. The divergence estimates the relevance of a document with respect to a query. The KL divergence is defined as:

$$KL(\theta_q||\theta_d) = \sum_w \Pr_q(w) \times [log(\Pr_q(w)) - log(\Pr_d(w))] \tag{3.6}$$

Note that the KL divergence is asymmetric. Better computation efficiency is the main reason that an asymmetric divergence is used. Usually, a query is much shorter than a document. With the help of the inverted index, the divergence between query and document can be efficiently measured. In contrast, computing a symmetric divergence is much slower.

3.4.4 Augmented CAL Algorithm

Based on our experiments for the seed documents selection, comparison of different classifiers, augmented feature engineering, and query expansion, we incorporated these potential promising methods into our modified CAL model. Our modified CAL process has the following steps:

1. Generate TF-IDF values of both unigram and bigram features for each document. Build index for all the documents in the corpus using Indri [12]. Apply BM25 to rank all the documents and return top 100 highest-scoring documents.

[12] https://www.lemurproject.org/indri/

2. Entropy based feature is generated on the basis of TF-IDF value. The entropy vector of each document is reduced to 200 dimensions using Latent Semantic Indexing(LSI).

3. Perform clustering on the top 100 documents based on their LSI vectors and construct the corresponding clustering weighted graph described in section 3.4.1. Select documents using the Graph method discussed in Section 3.4.1 and send these document for relevance assessment one by one using at most 50 assessments.

4. A synthetic seed document is constructed from the query terms.

5. The initial training set consists of one synthetic seed document and the assessed documents from step 3. Set initial batch size B as 1.

6. Randomly select 100 documents from the corpus and temporarily label them "not relevant". Add these presumptive not relevant documents into train set.

7. Train five logistic regression classifiers with different presumptive train set. Select $\lceil \frac{4B}{5} \rceil$ documents with the highest scores from the fusion list for review and label them as "relevant" or "not relevant".

8. If the prevalence of relevant documents among the $\lceil \frac{4B}{5} \rceil$ documents is high, continue judging the another $\lfloor \frac{B}{5} \rfloor$ documents with the highest-ranking scores.

9. Otherwise, obtain expanded terms from judged documents and generate a new ranked list according to Indri TF-IDF retrieval model. Perform RRF fusion with the lists generated from step 7 and select top $\lfloor \frac{B}{5} \rfloor$ documents for assessment.

10. Add all the reviewed documents to the train set. Increase B by $\lceil \frac{B}{10} \rceil$. Return to step 6 and start the next iteration until all the documents in the corpus have been assessed.

By following the BMI implementation [Cormack and Grossman, 2015*a*], our CAL algorithm combined the unigram and bigram features occurring at least twice in the collection. TF-IDF values were calculated for each feature. The squared sum of feature vector for each document is normalized to 1. We also applied Porter stemming method to stem word based features. Indri package provided both BM25 and TF-IDF ranking methods to retrieve documents. Therefore, we selected Indri to build index and rank documents for seed documents selection and query expansion.

LSI in Step 3 requires the operations of dimensionality reduction and singular value decomposition (SVD). RedSVD [13] is an effective tool to accelerate SVD computation. For documents clustering, we used K-Means clustering tool from Scikit Learn package [14]. We selected n top ranked documents with the highest BM25 score for seed documents selection. We set $k = \log(n)$ as the number of clusters. In this case, $k = 7$ where $n = 100$.

For the classifier, we still applied the Sofia-ML implementation of Pegasos logistic regression, with the same hyperparameters used in BMI: "–iterations 2000000 –dimensionality 110000". However, the number of dimensions sometimes needs to be dynamically updated according to the varied size of features. For some large corpus and n-gram features, we might need to increase the number of dimensions.

In Total Recall Track 2015, there were two main tasks: At Home task and Sandbox tasks. For the At Home task, we deployed our high-recall system on local machine and got the relevance feedback for documents via the online server. The other task was SandBox in which we were required to deploy our system in a virtual machine environment. In addition, we need to submit the virtual machine for evaluation so that the Track coordinators were able to execute our system within a restricted and private environment. We submitted our run "WaterlooClarke" for both tasks. Our submitted algorithms for both tasks were almost the same. We only changed some configuration settings in order to adapt our algorithm into virtual machine environment for Sandbox task.

3.5 Evaluation and Results

The TREC Total Recall track [Roegiest, Cormack, Grossman and Clarke, 2015; Grossman et al., 2016] used the gain curve as the main evaluation method. The gain curve plots recall as a function of the number of documents assessed. It provides a visible trade-off between recall and assessment effort without targeting any arbitrary threshold. By visualizing the relationship between recall and effort, we can easily compare the effectiveness of different high-recall retrieval systems or submitted runs.

In addition, Total Recall Track 2015 introduced a "call your shot" subtask, in which participants need to indicate when to stop a retrieval effort. The density of relevant documents decreases as the number of assessments increases. At some point, the benefit of finding a larger number of relevant documents no longer compensates the increase of assessment effort to find them. In the Total Recall Track, the participants were able to

[13] https://code.google.com/archive/p/redsvd/

[14] http://scikit-learn.org

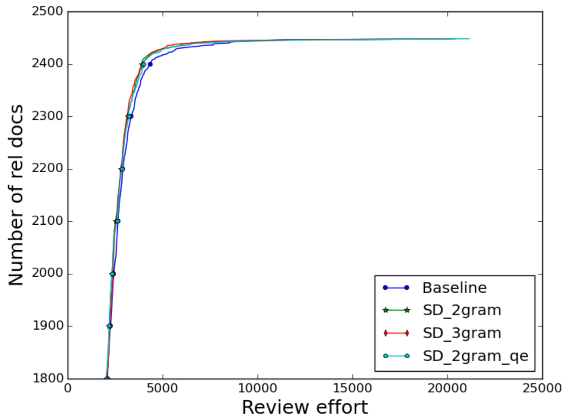

Figure 3.1: Comparison on test topic tr0.

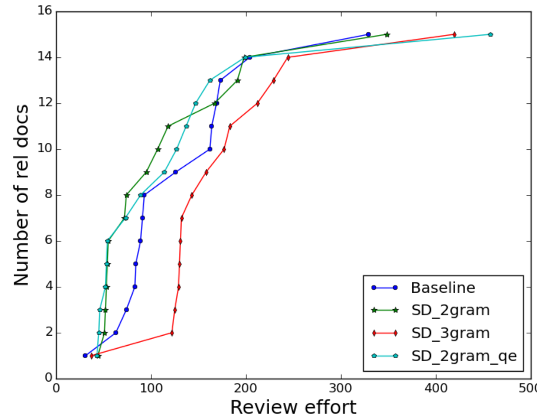

Figure 3.2: Comparison on test topic tr1.

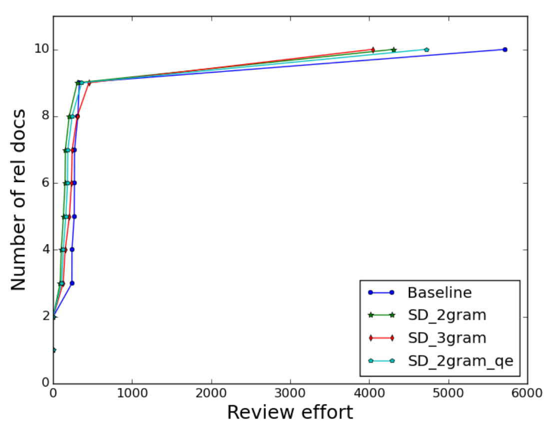

Figure 3.3: Comparison on test topic tr2.

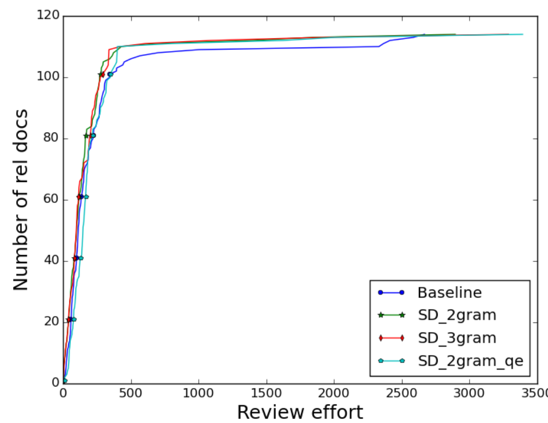

Figure 3.4: Comparison on test topic tr3.

indicate when they believed enough relevant documents had been found. At that point, the traditional set-based measures, including recall precision, effort, and F_1 were reported.

Another evaluation metric used in TREC Total Recall track is $recall@aR + b$ effort, where R is the number of relevant documents for a topic, a and b are constant variables. This metric measures recall achieved when $aR+b$ documents have been assssed. It normalizes the effort by the number of relevant documents for a particular topic. Then the recall over different topics can be averaged and the statistical tests can be performed at a certain $aR + b$. We can make corresponding gain curves according to different $recall@aR + b$ by varying a and b.

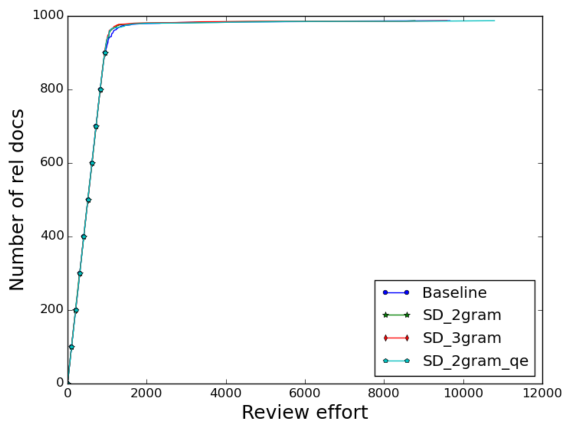

Figure 3.5: Comparison on test topic tr4.

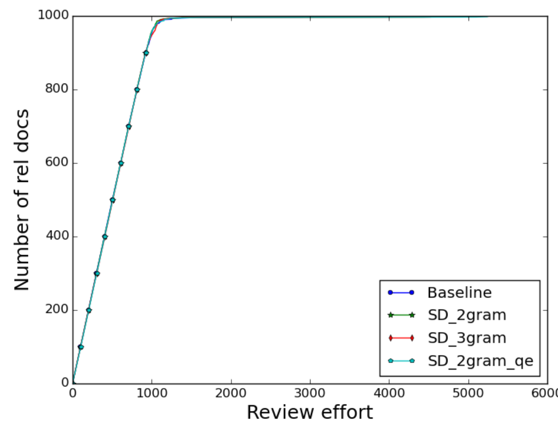

Figure 3.6: Comparison on test topic tr5.

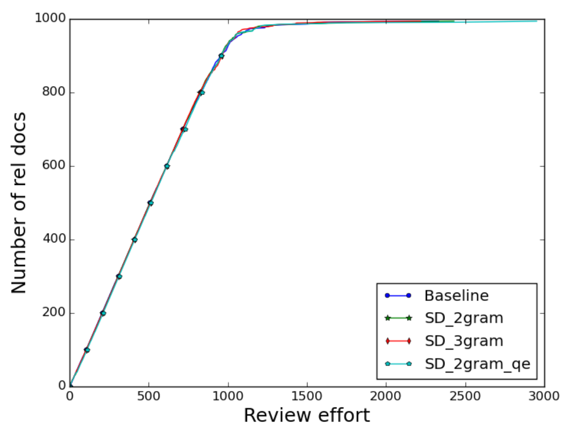

Figure 3.7: Comparison on test topic tr6.

First, I compared different feature engineering methods (unigram, union of unigram and bigram, and union of unigram, bigram, and trigram) detailed in Section 3.4.2. I also tested the value of adding the query expansion method described in Section 3.4.3. The comparison results of different methods on 3 practice topics of 20 Newsgroups and 4 practice topics of Reuters are shown from Figure 3.1 to Figure 3.7. The method SD_2gram_qe is the augmented CAL algorithm described in Section 3.4.4. The method Baseline reproduces the baseline BMI method. The method SD_2gram is the augmented CAL algorithm using the union of unigram and bigram as document features while SD_3gram uses the union of unigram, bigram, and trigram. I compared these four methods on 7 practice topics.

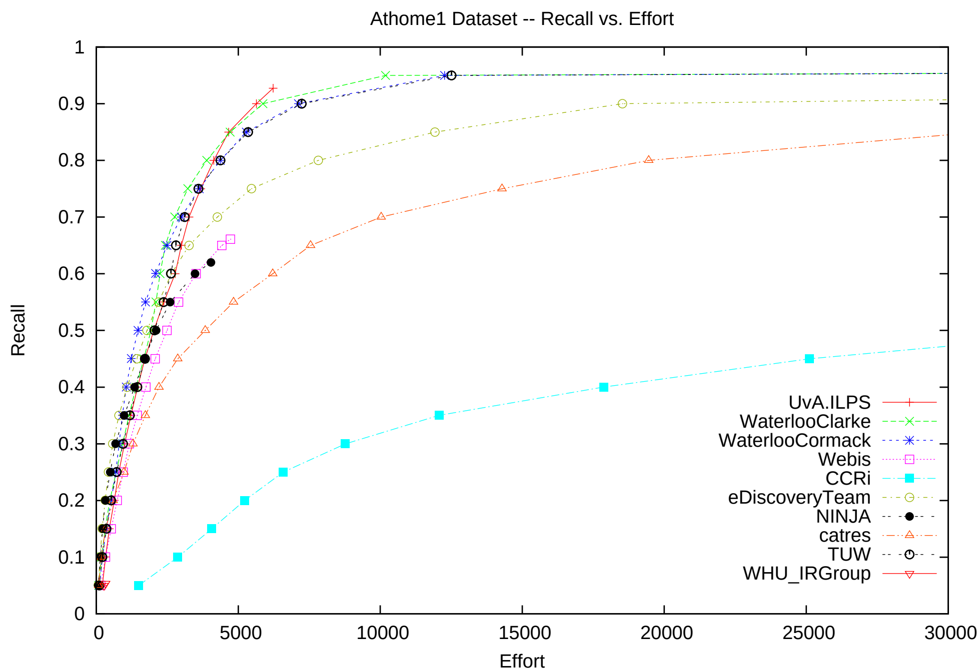

Figure 3.8: The averaged gain curves over 10 topics for submitted runs on dataset Athome1. This Figure is originally from the Total Recall Track 2015 overview paper [Roegiest, Cormack, Grossman and Clarke, 2015].

The gain curves show that SD_2gram_qe can achieve slightly better results compared to other three methods on these practice topics. Therefore, we adopted SD_2gram_qe and submitted our final runs using this method.

We submitted one automatic run (WaterlooClarke) to Total Recall Track 2015 for evaluation on both At-Home and Sandbox tasks. The averaged gain curves from different submitted runs on each data collection are shown from Figure 3.8 to Figure 3.9. The coordinators of Total Recall Track 2015 also submitted a run (WaterlooCormack) using the BMI method [Cormack and Grossman, 2015*a*]. "WaterlooCormack" replicated BMI and represented the baseline result. According to the official evaluation results, no submitted run consistently achieved higher recall than other runs at lower effort. Many runs appeared to have similar results and effectiveness. Some manual or automatic runs achieved higher recall than the BMI run on some topics at the same effort. However, no run consistently improved on the baseline. As shown in the averaged gain curves, our submitted run "WaterlooClarke" has no significant difference with the BMI run "WaterlooCormack". It

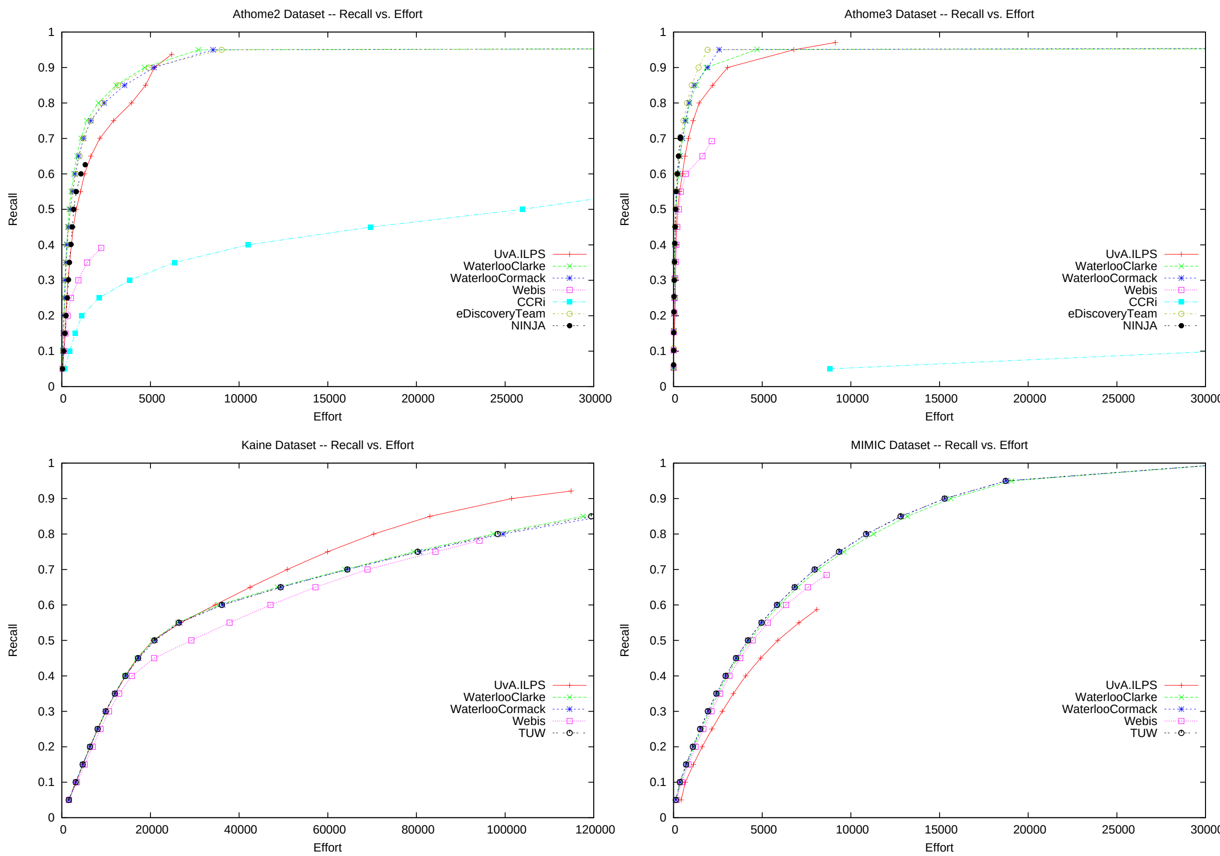

Figure 3.9: The averaged gain curves over 10 topics for submitted runs on datasets Athome2, Athome3, Kaine and Mimic.

is worth mentioning that even manual runs were not able to achieve higher recall than automatic runs at a given amount of assessment effort.

More detailedly, a number of runs can achieve 90% recall within $10,000$ assessments. For topic athome109 on Athome1 test collection, the BMI run and other automatic runs only found limited number of relevant documents in the beginning stage. It took those automatic runs a large amount of effort to find the majority of relevant documents. In contrast, the manual runs were able to find relevant documents more easily and achieved nearly 100% recall using much fewer assessments. In other words, the CAL method or other automatic methods have limitation on retrieving relevant information on certain topics while human assessors can easily break this kind of bottleneck.

Chapter 4

Evaluating Sentence-Level Relevance Feedback for High-Recall Retrieval

In this chapter we applies a novel simulation framework to evaluate whether the time and effort to achieve high recall using continuous active learning (CAL) can be reduced by presenting the assessor with an isolated sentence, as opposed to full document, for relevance feedback. Under the assumption that more time and effort is required to review an entire document than a single sentence, we found that the use of isolated sentence from a document for relevance feedback can reduce the assessment effort without meaningful reduction in recall, compared to state-of-the-art Baseline Model Implementation (BMI) of the AutoTAR Continuous Active Learning method employed in the TREC 2015 and 2016 Total Recall Track [Roegiest, Cormack, Grossman and Clarke, 2015; Grossman et al., 2016].

To simulate the sentence-level relevance feedback, a substantially complete set of relevance labels is needed prior to the simulation study. During the simulation process, the reviewer's assessment to any particular document is determined by requesting these previously determined labels. Furthermore, to simulate the presentation of isolated sentences to the reviewers for relevance feedback, we also require a prior set of relevance label for each sentence in every document, with respect to every topic.

In the current study, we augment four publicly available test collections with sentence-level relevance labels. We use a combination of the existing document relevance labels, new human assessments, and machine-learning method described in Section 4.2 to generate sentence labels for sentences. We use the available document labels to simulate document-level relevance feedback, and the newly created sentence labels to simulate sentence-level relevance feedback. Both sentence-level and document-level relevance feedback strategies are evaluated in terms of document-level recall—the fraction of relevant documents presented in whole or in part to the reviewer—as a function of review effort. The review effort is measured in several different ways. First, we use the total number of assessments rendered by the reviewer. Second, we also estimate the total number of sentences viewed by the reviewer to render those assessments. We assume that the reviewer's actual time and effort is likely to fall somewhere between these two bounds.

Besides choosing whether to present a full document or isolated sentence to the assessor for relevance feedback, it is also necessary to determine how to select the right document or sentence. As a baseline, we used the Baseline Model Implementation (BMI) implementation of the AutoTAR Continuous Active Learning method (CAL) shown in Section 3.4, which repeatedly uses machine learning classifier to select and present the next-most-likely relevant documents to the assessor for labelling. The labelled documents are then added into the training set. Three binary choices were incorporated into BMI: (1) whether to *present* full documents or isolated sentences to the reviewer for relevance feedback; (2) whether to *train* the classifier using full documents or isolated sentences; and (3) whether to *select* the highest-scoring document, and the highest-scoring sentence within that document, or to select the highest-scoring sentence, and the document containing that sentence. We compared and evaluated all eight combinations which were varied on these three binary choices described in Section 4.1.

We inferred that sentence-level feedback might be less accurate than document-level feedback, thus yielding lower recall for a given number of assessments. Nevertheless, sentence-level feedback could be rendered more quickly, potentially yielding higher recall within a given amount of review time and effort. We further inferred that selecting the highest-scoring sentence (as opposed to the highest-scoring document) and/or using reviewed sentences (as opposed to reviewed documents) for training the classifier might help to improve the accuracy and hence the efficiency of sentence-level feedback.

Contrary to our conjecture, we found that sentence-level feedback resulted in no meaningful loss in accuracy shown in Section 4.3. Our results suggest that relevance feedback based on isolated sentences can achieve higher recall with less time and effort, under the assumption that sentences can be assessed, on average, more quickly than full documents.

4.1 Integrate Continuous Active Learning with Sentence-Level Relevance Feedback

BMI implements the AutoTAR CAL method [Cormack and Grossman, 2015*a*], shown in Algorithm 3. The topic statement is regarded as a synthetic relevant document in Step 1. 100 randomly selected documents are labelled as "non-relevant" and added into the training set shown in Step 3. An initial logistic regression classifier is trained on this training set in Step 4. The highest-scoring B documents are selected from the unassessed documents and appended to system output in Steps 6 and 7. The system output records the list of the reviewed documents. The B documents labelled by reviewer are then added to the training set in Step 9. 100 randomly selected documents coded as non-relevant in the training set are replaced by the newly selected 100 random documents in Step 3 and 5. The classifier is retrained using the new training set. The classifier selects the next B highest-scoring not reviewed documents for review in the new batch. This process repeats until enough relevant documents have been found.

We modified BMI to use either sentences or documents at different stages of its process. As part of this modification, we consider the document collections to be the union of documents and sentences, and choose documents or sentences at each step, depending on a configuration parameter. For example, a single document of 100 sentences becomes 101 documents, where 1 document is the original document and the other 100 documents are the document's sentences.

BMI uses logistic regression as implemented by Sofia-ML[1] as its classifier. The logistic regression classifier was configured with logistic loss with Pegasos updates, L2 normalization on feature vectors with $lambda = 0.0001$ as the regularization parameter, AUC optimized training, and $200,000$ training iterations. The features used for training the classifier were word-based TF-IDF:

$$w = (1 + \log(tf)) \cdot log(N/df) \tag{4.1}$$

where w is the weight of the word, tf is the term frequency, N is the total number of documents and sentences, and df is the document frequency where both documents and sentences are counted as documents. The word feature space consisted of words occurring at least twice in the collection and all the words were downcased and stemmed by the Porter stemmer. We do not remove stopwords.

Algorithm 4 illustrates our modified BMI that enables either sentence-level or document-level feedback, training, and ranking. The system output O in Step 6 records the documents

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

ALGORITHM 4: Generic sentence feedback and document feedback algorithm

Step 1. Treat the topic statement as a relevant document and add this document into the training set;
Step 2. Set the initial batch size B to 1;
Step 3. Temporarily augment the training set by adding 100 random documents ($2d$) or sentences ($2s$) from the collection, temporarily labelled "non-relevant";
Step 4. Train the classifier using the training set. Then remove the random documents added in Step 3 from the training set;
Step 5. Derive the top B (best_sent, best_doc) pairs using the classifier. We have two choices $\{3d, 3s\}$ to select the (best_sent, best_doc) pair. The details of the $\{3d, 3s\}$ are shown in Table 4.1;
Step 6. Append the selected B best_doc to system output (coded as O). The system output records the list of best_doc that have been selected by the classifier and labelled by the reviewer;
Step 7. For each of the top B (best_sent, best_doc) pairs execute steps 8 to 10;
Step 8. Present either the best_sent ($1s$) or best_doc ($1d$) in the pair to the reviewer;
Step 9. Receive the relevance assessment l from reviewer;
Step 10. Add either (best_sent, l) as $2s$ or (best_doc, l) as $2d$ to training set;
Step 11. Increase B by $\lceil \frac{B}{10} \rceil$;
Step 12. Repeat steps 3 through 11 until substantially all relevant documents appear in the system output.

that have been assessed by reviewer. The system output also keeps the order of documents judged by reviewer so that we can use the system output to measure the recall achieved at a certain amount of effort. The corresponding human-in-the-loop framework which uses different level of relevance feedback is shown in Figure 4.1.

Steps 3, 5, 8 and 10 in Algorithm 4 involve three binary choices; we explored two possibilities for each choice, for a total of $2^3 = 8$ combinations. The principal choice happens in Step 8: whether to present the best_sent or the best_doc in the pair to the reviewer. We label these alternatives as $1s$ and $1d$, respectively. In support of this choice, it is necessary to decide how to build the training set in steps 3 and 10, and how to use the classifier to identify the top B (best_sent, best_doc) pairs in Step 5. In Step 10, we choose new added training samples either: ($2s$) the best_sent with corresponding label l; or ($2d$) the best_doc with corresponding label l. In step 3, the 100 randomly selected non-relevant training examples are chosen by either: ($2s$) 100 random sentences; or ($2d$) 100 random documents. In Step 5, we choose the (best_sent, best_doc) pair either: ($3s$)

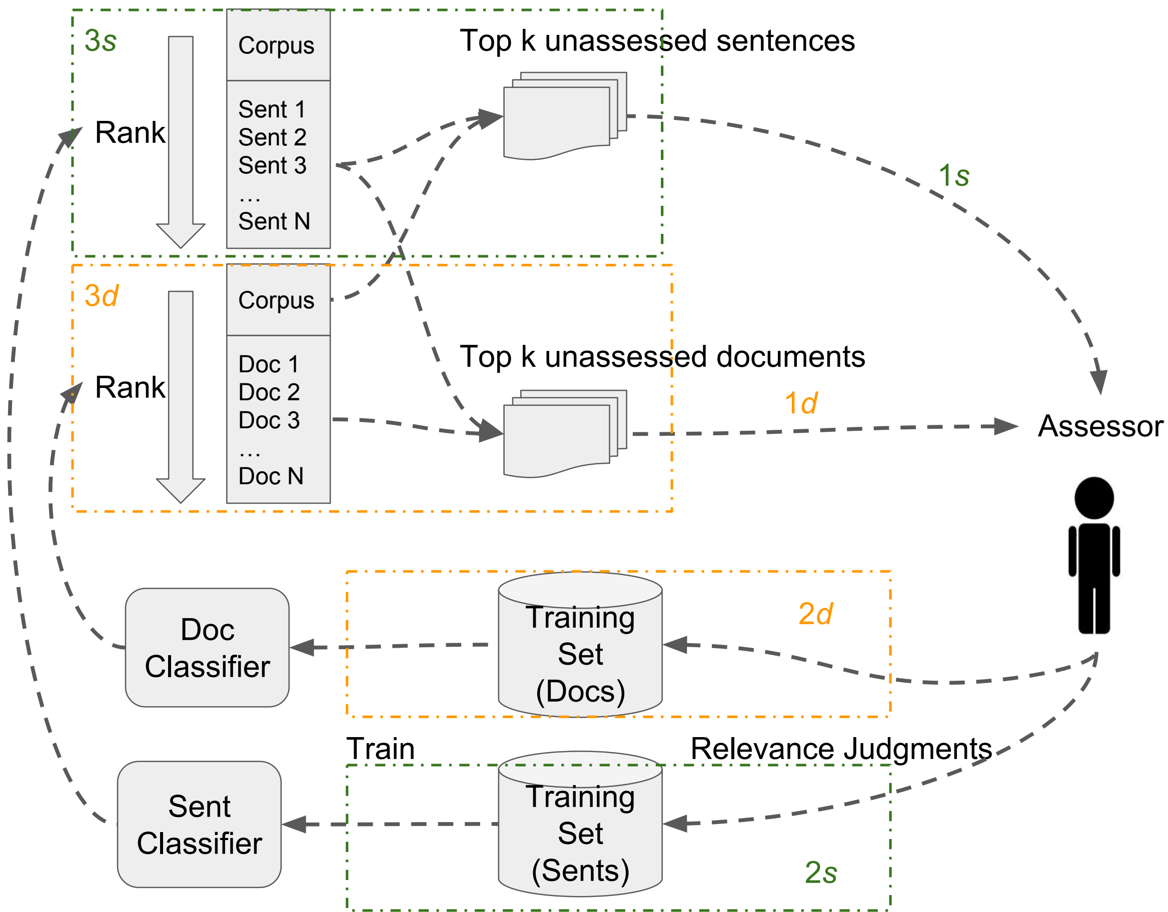

Figure 4.1: The human-in-the-loop CAL framework for sentence feedback and document feedback algorithm.

the highest-scoring sentence contained in any document not yet in system output O, and the document containing that sentence; or ($3d$) the highest-scoring document not yet in system output O, and the highest-scoring sentence within that document. The sentences in ($3d$) were scored by the same classifier that was also used for document scoring. More formally, if we denote system output by O, $3s$ is defined by Equations 4.2 and 4.3:

$$best\_sent = \underset{sent \notin doc \in O}{\operatorname{argmax}} \; Score(sent) \tag{4.2}$$

$$best\_doc = d \mid best\_sent \in d \tag{4.3}$$

while $3d$ is defined by Equations 4.4 and 4.5:

$$best\_doc = \underset{doc \notin O}{\text{argmax}}\ Score(doc) \tag{4.4}$$

$$best\_sent = \underset{sent \in best\_doc}{\text{argmax}}\ Score(sent) \tag{4.5}$$

We investigated three dimensions: either using a sentence or a document for assessment and relevance feedback, training, and ranking. The description of the three dimensions are shown as follows and detailed in Table 4.1:

- **1st Dimension ($1d$ or $1s$)**: Present the best document $1d$ or the best sentence $1s$ from (document, sentence) pair for assessor to review;
- **2nd Dimension ($2d$ or $2s$)**: Add a document $2d$ or a sentence $2s$ from newly judged (document, sentence) pair as new training example;
- **3rd Dimension ($3d$ or $3s$)**: Select the best (document, sentence) pair from either highest-ranking sentence and the document containing that sentence ($3s$); Or highest-ranking document and the highest-scoring sentence within that document ($3d$).

Using documents for each choice of the process (choosing $1d$, $2d$, and $3d$) is our baseline which replicates BMI, except for the use of the union of documents and sentences to compute word features. For simplicity, we use the notation *ddd* to represent this combination of choices, More generally, we use XYZ to denote the choices on three dimensions $1X$, $2Y$ and $3Z$, where $X, Y, Z \in \{d, s\}$. The choices for all the eight combinations are shown in Table 4.1.

4.2 Test Collections

We use four public test collections to compare and evaluate the eight different variations of continuous active learning. We use the three test collections from the TREC 2015 Total Recall track: Athome1, Athome2, and Athome3. We also use the test collection from the TREC 2004 HARD track [Allan, 2003; Voorhees and Harman, 2000]. For each collection, we used NLTK's Punkt Sentence Tokenizer[2] to break all documents into sentences. Corpus

[2]http://www.nltk.org/api/nltk.tokenize.html

Table 4.1: Eight combinations on three binary choices.

#No	Strategy	Present best_doc or best_sent to reviewer ($1d$ or $1s$)	Add (best_doc, l) or (best_sent, l) and 100 random sentences or documents as non-relevant to training set ($2d$ or $2s$)	Select (best_sent, best_doc) pair ($3d$ or $3s$)
1	ddd	best_doc	$2d$: (best_doc, l) and 100 randomly selected documents treated as non-relevant	$3d$: the highest-scoring document not yet in system output, and the highest-scoring sentence within that document.
2	sdd	best_sent	$2d$	$3d$
3	dsd	best_doc	$2s$: (best_sent, l) and 100 randomly selected sentences treated as non-relevant	$3d$
4	ssd	best_sent	$2s$	$3d$
5	dds	best_doc	$2d$	$3s$: the highest-scoring sentence contained in any document not yet in system output, and the document containing that sentence.
6	sds	best_sent	$2d$	$3s$
7	dss	best_doc	$2s$	$3s$
8	sss	best_sent	$2s$	$3s$

statistics for the four collections are shown in Table 4.2. Each document contains on average 15.9, 22.8, 28.4, and 16.3 sentences for Athome1, Athome2, Athome3, and HARD test collections, respectively.

In order to compare sentence-level feedback with document-level feedback strategies, we needed complete relevance labels for all sentences as well as for all documents in the

Table 4.2: Dataset statistics

Dataset	Number of topics	Number of documents	Number of sentences	Number of relevant documents
Athome1	10	290,099	4,616,934	43,980
Athome2	10	460,896	10,493,480	20,005
Athome3	10	902,434	25,622,071	6,429
HARD	25	652,309	10,606,819	1,682

collections.

The TREC 2004 HARD track's collection provided pooled assessments with complete relevance labels for all documents in the pool. In addition, for 25 topics of HARD collection, every relevant document was divided by the TREC assessors into relevant and non-relevant passages identified by character offsets. For the HARD collection, the 25 topics with passage judgments were used for our experiment. We labelled a sentence as relevant if it overlapped with a relevant passage. Sentences that did not overlap with a relevant passage were labelled non-relevant.

For both the HARD track collection and the Total Recall collections, sentences from non-relevant and unassessed documents were labelled as non-relevant. We made such an assumption since a relevant document should contain at least one relevant sentence. If a document is non-relevant, all the sentences in it should be regarded as non-relevant.

The test collections from Total Recall Track 2016 provided complete document-level relevance judgments, i.e., the relevance of every document in the collection is known. Each relevant document is composed of one or more than one relevant sentences and zero or more non-relevant sentences. In order to simulate the sentence-level relevance feedback, we need to know the relevance of each sentence in the collection, with respect to each topic. To label the sentences as relevant or non-relevant the author employed "Scalable CAL" (S-CAL) [Cormack and Grossman, 2016*b*] to build a calibrated high-accuracy classifier that was used to label every sentence within every relevant document. Our total effort to train the S-CAL classifier was to review 610, 453, and 376 sentences, on average, per topic, for each of the three Athome datasets, respectively.

While neither of these methods is able to yield a perfect labelling, their purpose is to simulate human relevance feedback, which is imperfect as well. The internal calibration of our S-CAL classifier indicated its recall and precision both to be above 0.8 ($F_1 = 0.82, 0.87, 0.81$ for Athome1, Athome2, and Athome3, respectively), which is comparable to human accuracy [Cormack and Grossman, 2016b]. We inferred that this derived sentence-level label set would be good enough to verify the effectiveness of sentence-level feedback. Similarly, we inferred that overlap between sentences and relevant passages in the HARD collection would also yield labels that were good enough for this purpose.

Table 4.3: Micro-averaged statistics of generated sentences label set on different datasets.

Dataset	Number of sentences per document	Number of sentences per relevant document	Number of relevant sentences per relevant document	Position of the first relevant sentence in relevant document	Proportion of relevant documents has relevant sentence
Athome1	15.9	18.1	7.8	2.0	0.98
Athome2	22.8	19.4	3.8	5.1	0.97
Athome3	28.4	47.2	7.5	19.1	0.97
HARD	16.3	23.2	11.3	4.0	1.00

The results of our sentence labelling are shown in Table 4.3. The average position of the first relevant sentence in each relevant document is shown in the fifth column, while the distribution of such positions is shown in Figure 4.2. On Athome1, Athome2 and HARD three datasets, more than 50% relevant documents in each dataset have their first relevant sentences located at the first sentences. However, the position of the first relevant sentence in the relevant document is larger than 2 for all the four datasets. In other words, the reviewers need to review at least more than two sentences to find the first relevant sentence in each relevant document under the assumption that reviewer read the document sequentially. The sixth column shows the fraction of relevant documents containing at least one sentence labelled relevant. It shows that nearly every relevant document contains at least one relevant sentence.

4.3 Evaluation Methods

The current study adopts and extends the human-in-the-loop CAL evaluation apparatus which is simulated by the TREC Total Recall track. It has the following process. A standard test collection consisting of a set of documents, topic statements, and relevance assessments (qrels) is provided. The most-likely relevant document is presented to the

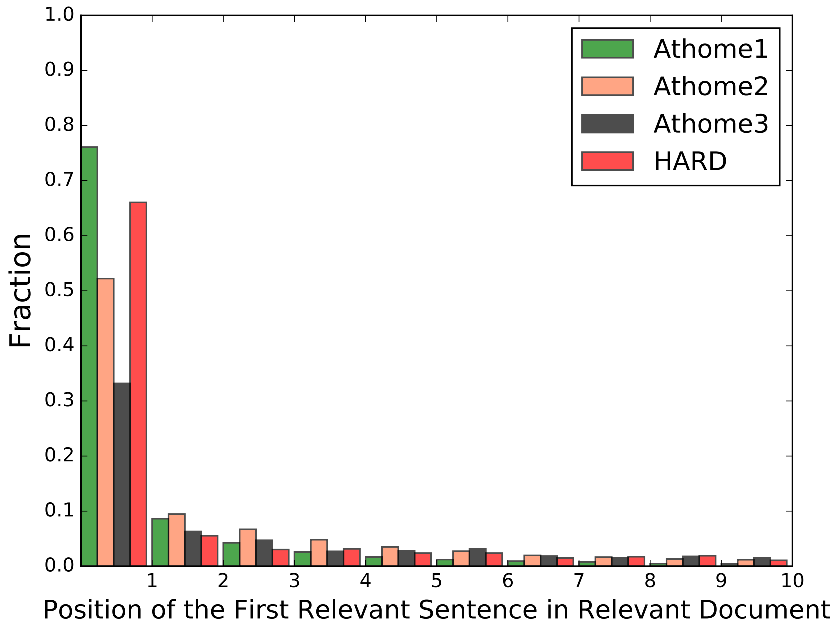

Figure 4.2: The distribution of the position of the first relevant sentence in the relevant documents for different document collections.

reviewer for assessment. The reviewer's relevance assessment to a given document is simulated by consulting the existing qrels. The relevance of the labelled document is then fed back to the machine learning classifier. The machine-learned classifier then ranks the document collection and chooses the next-most-likely-relevant document to present. The process continues until a formal or informal stopping criterion is met, suggesting that substantially all relevant documents have been presented to the reviewer.

To simulate sentence-level feedback it was necessary to extend the evaluation apparatus to incorporate a sentence dataset and sentence qrels. The sentence dataset consists of all sentences extracted from documents in the test collection. The sentence qrels consist of relevance assessment for each sentence. To simulate sentence-level feedback, the apparatus presents to the simulated reviewer a single sentence, as determined by the system under test, and sends the reviewer's assessment to the system, which then selects the next-most-likely-relevant sentence for review. The document containing the already reviewed sentence will not be presented to reviewer for assessment again. Any document in the test collection is reviewed only once. The "system-selected documents" used for evaluation consist of the stream of documents from which the sentences presented to the reviewer were extracted. In our paper, the "system-selected documents" are recorded in the system output (O) mentioned in the Step 6 of Algorithm 4. The same evaluation apparatus is used to simulate document-level feedback, except that here, the system selects a document to the reviewer for assessment, and the reviewer's feedback is simulated by consulting the document qrels. In document-level-feedback mode, the apparatus is operationally the same as the TREC Total Recall apparatus.

Recall is the fraction of relevant documents presented to the reviewer for assessment, with respect to the total number of relevant documents (R), regardless of whether document- or sentence-level feedback is employed. In our experiment, the documents presented to the reviewer are recorded by the system output (O). We measure the recall ($Recall@E$) achieved at a given effort (E) using the Equation 4.6:

$$Recall@E = \frac{|\{O@E\} \cap \{Relevant\ documents\}|}{|\{Relevant\ documents\}|} \tag{4.6}$$

where the $O@E$ is the system output truncated at the effort E. The relevance judgment sets are the gold standard relevance assessment (qrels) provided by the TREC Total Recall 2015 Track and HARD 2004 Track for the corresponding datasets and topics.

The Total Recall Track 2015 and 2016 measured recall as a function of effort, where effort was measured by the number of assessments rendered by the reviewer. Gain curves were used to illustrate the overall shape of the function, and recall at particular effort levels. $aR + b$ were tabulated to measure different levels of effort, where R is the number of relevant documents for a given topic, a is the constant 1, 2, or 4, and b is the constant 0, 100, or 1000. Intuitively, these measures show the recall that can be achieved with effort proportional to the number of relevant documents R, plus some fixed overhead amount b.

In this paper, we call the number of assessments as E_{judge}. This measurement of effort can only provide a rough estimate of review effort. However, as mentioned in Section 2.3.1, the review effort (e.g., time cost or monetary cost) to review a sentence or a document can vary a lot. As mentioned in Section 6.1.1, assessing the relevance of a document usually takes longer time than assessing a single sentence. Therefore, for sentence-level relevance feedback strategies, simply using the total number of documents assessed to measure the total review effort and compare it with that of document-level relevance feedback could be not accurate and realistic. The number of total assessments E_{judge} might not be enough to reflect the real assessment effort to achieve a certain recall.

We can also measure assessment effort as the number of sentences read, i.e., effort $= E_{sent}$. E_{sent} is the number of sentences that must be read by the reviewer to render a judgment. If a simulated reviewer provides an assessment on a single sentence, the reviewer reads one sentence and makes one assessment. When a full document is presented for assessment, we assume that the reviewer read the document sequentially from the beginning to the first relevant sentence and then make one assessment. In this case, E_{sent} is equal to the number of sentences to read till the first relevant sentence in the document. If the document is non-relevant, the assessor needs to read all of the sentences in the document.

Besides E_{judge} and E_{sent}, we also measure recall as a function of effort E, but in this paper, we try to measure effort as a linear combination of the number of assessments rendered by the reviewer E_{judge}, and the number of sentences that must be read by the reviewer to render a judgment E_{sent}.

The ratio of effort required to make an assessment to the effort required to read a sentence is not necessarily 1.0. To vary and explore different ratios of effort, we express effort, E_λ, as a linear combination of E_{judge} and E_{sent}:

$$E_\lambda = (1 - \lambda) \cdot E_{judge} + \lambda \cdot E_{sent} \tag{4.7}$$

where E_{judge} is the number of assessments and E_{sent} is the number of sentences read. At one extreme, we only care about the number of assessments, i.e., $E_0 = E_{judge}$. At the other extreme, we only count the effort of reading sentences, i.e., $E_1 = E_{sent}$. For sentence-level feedback, $E_{judge} = E_{sent} = E_\lambda$, regardless of λ, since the number of sentences needed to read for a given sentence is just 1.

For single assessment on each document d, the number of assessments on d is $E_{judge} = 1$. We can simplify the assessment effort defined in Equation 4.7 for a single document d as $E_\lambda = 1 + \lambda \cdot (E_{sent} - 1)$. If the $E_{sent} > 1$ for the document d, then $E_\lambda > 1$. If the number of sentences needed E_{sent} for reviewing this document d increases, the E_λ also increases.

4.4 Results

4.4.1 Results based on E_{judge}

We compared the sentence-level feedback strategies with the document-level feedback strategies on three different dimensions—in total, eight combinations shown in Table 4.1. As explained in Section 4.3, we measure the performance as recall versus effort. We can measure effort as the number of assessments (judgments) made by the reviewer, i.e., effort $= E_{judge}$.

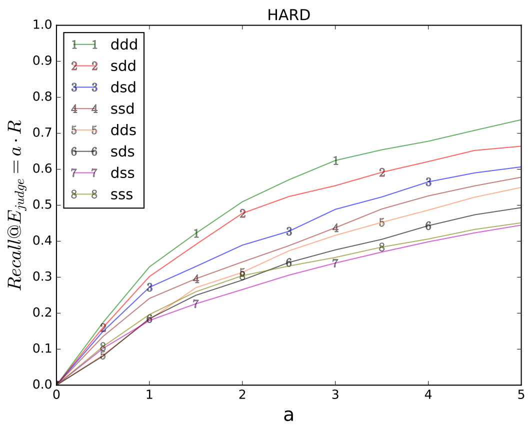

Figure 4.3: Recall at $E_{judge} = a \cdot R$ for varying a on HARD.

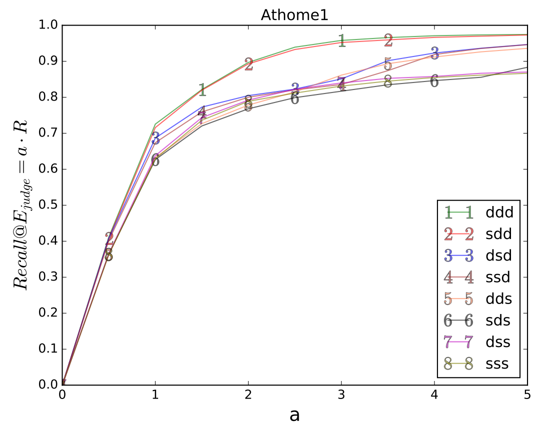

Figure 4.4: Recall at $E_{judge} = a \cdot R$ for varying a on Athome1.

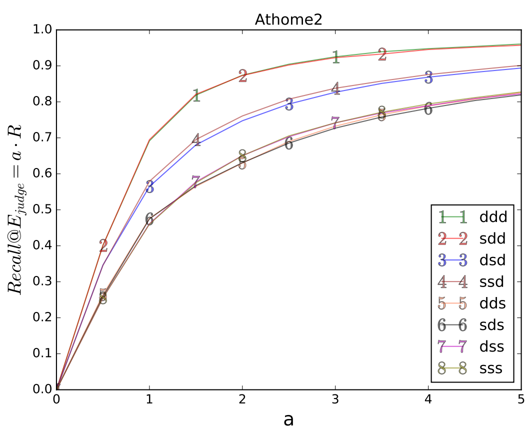

Figure 4.5: Recall at $E_{judge} = a \cdot R$ for varying a on Athome2.

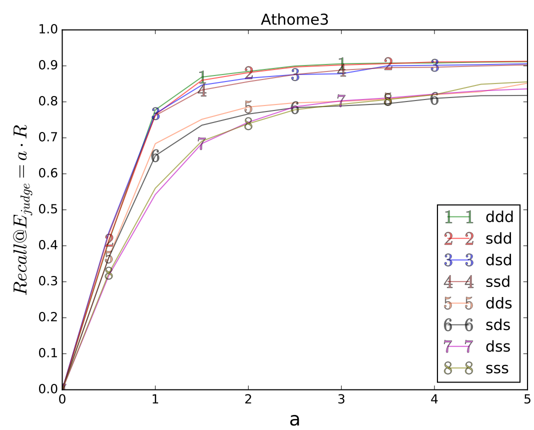

Figure 4.6: Recall at $E_{judge} = a \cdot R$ for varying a on Athome3.

Figures 4.3 show recall vs. effort on the HARD test collection. Figure 4.3 measures effort as a function of the number of judgments (E_{judge}), where the horizontal axis reports judgments in multiples a of the number of relevant document R. For example, $a \cdot R$ documents, where $a = 2$ means that twice as many judgments have been made as there are relevant documents. We vary $a \in \{1, 2, 3, 4, 5\}$ to measure the recall achieved at different level of effort. The corresponding plots for three Athome collections are found at the end of the paper in Figures 4.4–4.6.

Table 4.4: Recall at $E_{judge} = R$ for different strategies on different datasets. We bold the greater value if the difference in recall between *sdd* and *ddd* is statistically significant. The overall is the average result over all the 55 topics from all the four datasets.

Dataset	Effort	ddd	sdd	dsd	ssd	dds	sds	dss	sss
Athome1	1R_Judge	0.73	0.72	0.69	0.67	0.63	0.63	0.64	0.63
Athome2	1R_Judge	0.69	0.69	0.56	0.58	0.48	0.47	0.46	0.46
Athome3	1R_Judge	0.78	0.76	0.77	0.76	0.68	0.65	0.54	0.56
HARD	1R_Judge	0.34	0.31	0.28	0.25	0.19	0.19	0.19	0.20
Overall	1R_Judge	0.55	0.54	0.50	0.48	0.41	0.41	0.38	0.39

Table 4.5: Recall at $E_{judge} = 2R$ for different strategies.

Dataset	Effort	ddd	sdd	dsd	ssd	dds	sds	dss	sss
Athome1	2R_Judge	0.90	0.89	0.80	0.80	0.78	0.77	0.79	0.79
Athome2	2R_Judge	0.87	0.87	0.75	0.76	0.63	0.63	0.65	0.65
Athome3	2R_Judge	0.88	0.88	0.87	0.86	0.79	0.77	0.74	0.74
HARD	2R_Judge	0.53	0.50	0.41	0.36	0.33	0.30	0.28	0.32
Overall	2R_Judge	0.72	0.71	0.62	0.60	0.55	0.53	0.52	0.54

Table 4.6: Recall at $E_{judge} = 4R$ for different strategies.

Dataset	Effort	ddd	sdd	dsd	ssd	dds	sds	dss	sss
Athome1	4R_Judge	0.97	0.97	0.92	0.92	0.91	0.85	0.86	0.85
Athome2	4R_Judge	0.95	0.95	0.87	0.88	0.79	0.78	0.79	0.79
Athome3	4R_Judge	0.91	0.91	0.90	0.90	0.82	0.81	0.82	0.82
HARD	4R_Judge	0.71	0.65	0.59	0.55	0.51	0.46	0.41	0.42
Overall	4R_Judge	0.83	0.81	0.76	0.74	0.69	0.65	0.64	0.64

In general, when effort is measured in terms of judgments only (E_{judge}), we find that the training on and selecting documents to be superior to other methods regardless whether the reviewer judged documents (*ddd* strategy) or sentences (*sdd* strategy), across all eight combinations, for all four datasets, for all a. We also find that training on sentences with the selection of documents (*dsd* and *ssd*) strategies to be worse than the strategies that training on documents and selecting documents (*ddd* and *sdd*) on all datasets, but superior to the other four strategies: *dds*, *dss*, *sds*, and *sss*. The overall comparison of judgments effort for all the eight combinations is that $\{ddd|sdd\} > \{dsd|ssd\} > \{dds|sds|dss|sss\}$.

These results suggest that training using documents and selecting the highest-ranking document from the document-rank list to review (*ddd* and *sdd*) will lead to superior results over other strategies, regardless of whether sentences or documents are presented to the reviewer for feedback. At the same time, the choice of using sentences (*sdd*) or documents (*ddd*) for relevance feedback has very little impact on the recall that can be achieved for a given number of assessments.

Table 4.7: recall[*sdd*]-recall[*ddd*] at effort $= a \cdot E_{judge}$ (95% Confidence interval).

Dataset	a=1	a=2	a=4
Athome1	(-0.025, 0.006)	(-0.012, 0.003)	(-0.009, -0.0003)[3]
Athome2	(-0.008, 0.014)	(-0.005, 0.003)	(-0.007, 0.002)
Athome3	(-0.043, 0.016)	(-0.015, 0.008)	(-0.005, 0.011)
HARD	(-0.074, 0.020)	(-0.071, 0.007)	(-0.122, 0.009)
Overall	(-0.037, 0.006)	(-0.034, 0.002)	(-0.056, 0.003)

The actual recall achieved by each strategy at multiples of R is reported in Table 4.4 ($1R$), Table 4.5 ($2R$), and Table 4.6 ($4R$). In each table, we compare the *ddd* and *sdd* methods and if the difference in recall is statistically significant, we bold the greater value. We measure statistical significance with a two-sided, Student's t-test and significance is for p-values less than 0.05. For example, in Table 4.6, when effort is equal to the four times of number of relevant documents ($4R$) and measured by the number of assessments (4R_Judge) on Athome1, the *ddd* (recall=0.97) and *sdd* (recall=0.97) methods are different at a statistically significant level.

The most interesting observation to be made from Tables 4.4, 4.5, and 4.6 is that when effort is measured in number of judgments E_{judge}, *sdd* and *ddd* are usually equivalent. For Athome1, Athome2, and Athome3, both *sdd* and *ddd* can achieve nearly 0.90 recall when $E_{judge} = 2R$. In other words, the precision of both strategies are approximately 0.45 on these three datasets. However, for HARD dataset, the effectiveness of *sdd* and *ddd* for achieving high recall is lower. *ddd* can only achieve 0.71 recall at $E_{judge} = 4R$. Nevertheless, there is still no significant difference between *sdd* and *ddd* on HARD dataset.

Correspondingly, we also show the confidence interval of the difference between *ddd* and *sdd* for different effort measurements E_{judge} with various values of a in Table 4.7.

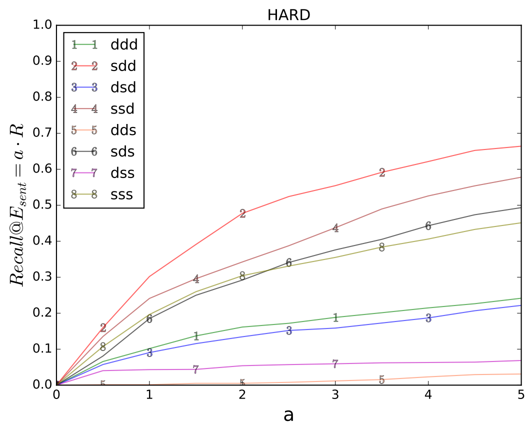

Figure 4.7: Recall at $E_{sent} = a \cdot R$ with varying a on HARD.

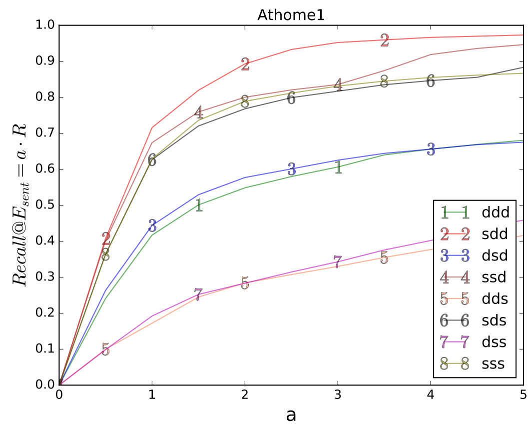

Figure 4.8: Recall at $E_{sent} = a \cdot R$ with varying a on Athome1.

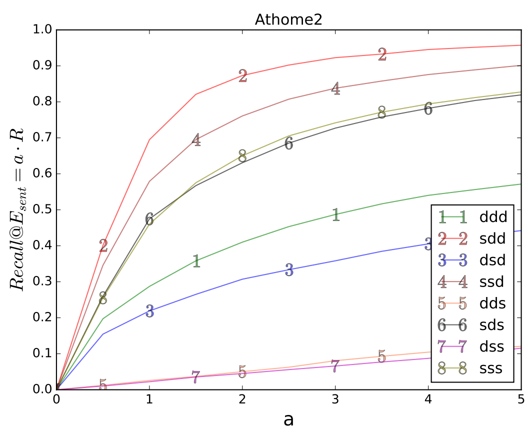

Figure 4.9: Recall at $E_{sent} = a \cdot R$ with varying a on Athome2.

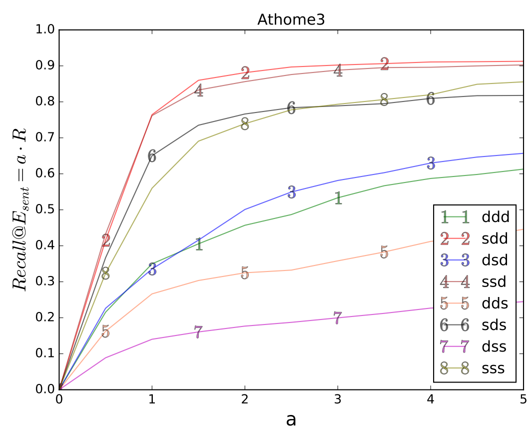

Figure 4.10: Recall at $E_{sent} = a \cdot R$ with varying a on Athome3.

4.4.2 Results based on the E_{sent} and E_λ

Figures 4.7 show recall vs. effort for the HARD test collection. It measures effort as a function of the number of sentences read (E_{sent}). We also vary a from $1, 2, 3, 4, 5$ to compare recall achieved at different level of effort E_{sent}. The corresponding gain curves

[3]The mean difference between recall[*sdd*] and recall[ddd] equals 0.0046 and $p = 0.037$ at effort $= 4 \cdot E_{judge}$ on Athome1.

of recall at different E_{sent} for Athome1, Athome2, and Athome3 datasets are shown in Figures 4.8 – 4.10.

When effort is measured in terms of sentences read only (E_{sent}), all of the sentence-level feedback strategies in which reviewer judges documents $\{sdd, ssd, sds, sss\}$ achieve much higher recall than the document-level feedback strategies in which reviewer judges sentences $\{ddd, dsd, dds, dss\}$ for a given level of effort, as measured in terms of the number of sentences reviewed. Among the four sentence-level feedback strategies, *sdd* is superior, and the relative effectiveness among the sentence-based strategies is consistent with the result when effort is measured by the number of assessments. The overall ranking of four sentence-level feedback strategies evaluated by number of sentences read is $\{sdd\} > \{ssd\} > \{sds|sss\}$. What this means is that for essentially the same number of judgments, we can achieve the same level of recall by only judging the best sentence from a document — we do not need to examine the entire document to judge its relevance.

Table 4.8: Recall at $E_{0.5} = R$, and $E_{sent} = R$ for different strategies on different datasets. We bold the greater value if the difference in recall between *sdd* and *ddd* is statistically significant. The overall is the average result over all the 55 topics from all the four datasets.

Dataset	Effort	ddd	sdd	dsd	ssd	dds	sds	dss	sss
Athome1	1R_0.5	0.48	**0.72**	0.51	0.67	0.24	0.63	0.26	0.63
	1R_Sent	0.42	**0.72**	0.44	0.67	0.17	0.63	0.19	0.63
Athome2	1R_0.5	0.36	**0.69**	0.27	0.58	0.05	0.47	0.04	0.46
	1R_Sent	0.29	**0.69**	0.22	0.58	0.03	0.47	0.02	0.46
Athome3	1R_0.5	0.40	**0.76**	0.41	0.76	0.30	0.65	0.16	0.56
	1R_Sent	0.35	**0.76**	0.34	0.76	0.27	0.65	0.14	0.56
HARD	1R_0.5	0.15	**0.31**	0.13	0.25	0.01	0.19	0.05	0.20
	1R_Sent	0.11	**0.31**	0.09	0.25	0.00	0.19	0.04	0.20
Overall	1R_0.5	0.29	**0.54**	0.28	0.48	0.11	0.41	0.11	0.39
	1R_Sent	0.24	**0.54**	0.22	0.48	0.09	0.41	0.08	0.39

Table 4.8 ($1R$), Table 4.9 ($2R$), and Table 4.10 ($4R$) show the actual recall achieved by each method at different effort. These tables report effort as E_{sent} and a equal combination of number of judgments and number of sentences read ($recall@E_\lambda$, where $\lambda = 0.5$). If the difference between *ddd* and *sdd* in recall is statistically significant, we also bold the greater value. We measure statistical significance with a two-sided, Student's t-test and significance is for p-values less than 0.05.

We also calculate the 95% confidence interval for the difference of $recall@E_{0.5} = a \cdot R$ between *ddd* and *sdd*. We find that $recall@E_\lambda = a \cdot R$ is significantly better for *sdd* than

Table 4.9: Recall of different strategies at $E_{0.5} = 2R$ and $E_{sent} = 2R$

Dataset	Effort	ddd	sdd	dsd	ssd	dds	sds	dss	sss
Athome1	2R_0.5	0.63	**0.89**	0.63	0.80	0.36	0.77	0.37	0.79
	2R_Sent	0.55	**0.89**	0.58	0.80	0.28	0.77	0.28	0.79
Athome2	2R_0.5	0.51	**0.87**	0.38	0.76	0.10	0.63	0.08	0.65
	2R_Sent	0.41	**0.87**	0.31	0.76	0.05	0.63	0.05	0.65
Athome3	2R_0.5	0.55	**0.88**	0.59	0.86	0.39	0.77	0.22	0.74
	2R_Sent	0.46	**0.88**	0.50	0.86	0.32	0.77	0.18	0.74
HARD	2R_0.5	0.21	**0.50**	0.18	0.36	0.02	0.30	0.07	0.32
	2R_Sent	0.17	**0.50**	0.14	0.36	0.01	0.30	0.06	0.32
Overall	2R_0.5	0.40	**0.71**	0.37	0.60	0.16	0.53	0.15	0.54
	2R_Sent	0.33	**0.71**	0.32	0.60	0.12	0.53	0.12	0.54

Table 4.10: Recall at $E_{0.5} = 4R$ and $E_{sent} = 4R$

Dataset	Effort	ddd	sdd	dsd	ssd	dds	sds	dss	sss
Athome1	4R_0.5	0.72	**0.97**	0.71	0.92	0.49	0.85	0.55	0.85
	4R_Sent	0.66	**0.97**	0.66	0.92	0.38	0.85	0.40	0.85
Athome2	4R_0.5	0.64	**0.95**	0.51	0.88	0.18	0.78	0.18	0.79
	4R_Sent	0.54	**0.95**	0.41	0.88	0.10	0.78	0.09	0.79
Athome3	4R_0.5	0.67	**0.91**	0.72	0.90	0.50	0.81	0.30	0.82
	4R_Sent	0.59	**0.91**	0.63	0.90	0.41	0.81	0.23	0.82
HARD	4R_0.5	0.29	**0.65**	0.26	0.55	0.05	0.46	0.08	0.42
	4R_Sent	0.22	**0.65**	0.19	0.55	0.02	0.46	0.07	0.42
Overall	4R_0.5	0.50	**0.81**	0.47	0.74	0.24	0.65	0.22	0.64
	4R_Sent	0.43	**0.81**	0.40	0.74	0.17	0.65	0.16	0.64

ddd for all values of a when $\lambda = 0.5$. We show the confidence interval of the difference between *ddd* and *sdd* for different effort measurements E_{sent} and E_{λ} with various values of a in Tables 4.11 and 4.12.

To get a better sense of when *sdd* becomes superior to *ddd*, we varied λ from 0 to 1 by step size 0.05 and plotted in Figure 4.11 the 95% confidence interval for the difference of $recall@E_{\lambda} = a \cdot R$ between *ddd* and *sdd*. As can be seen, once the cost of reading sentences starts to have some weight where $\lambda = 0.05$, *sdd* becomes superior to *ddd*. The recall[*sdd*]-recall[*ddd*] became larger with the increase of λ.

For single assessment on each document d, we can simplify the effort $E_{\lambda=0.05}$ for doc-

Table 4.11: recall[sdd]-recall[ddd] at effort = $a \cdot E_{sent}$ (95% Confidence interval).

Dataset	a=1	a=2	a=4
Athome1	(0.178, 0.42)	(0.181, 0.508)	(0.107, 0.514)
Athome2	(0.308, 0.508)	(0.352, 0.574)	(0.266, 0.545)
Athome3	(0.292, 0.537)	(0.244, 0.605)	(0.148, 0.499)
HARD	(0.121, 0.279)	(0.222, 0.41)	(0.297, 0.516)
Overall	(0.242, 0.348)	(0.307, 0.428)	(0.306, 0.442)

Table 4.12: recall[sdd]-recall[ddd] at effort = $a \cdot E_{0.5}$ (95% Confidence interval).

Dataset	a=1	a=2	a=4
Athome1	(0.121, 0.344)	(0.127, 0.393)	(0.041, 0.445)
Athome2	(0.210, 0.360)	(0.227, 0.401)	(0.163, 0.390)
Athome3	(0.250, 0.373)	(0.246, 0.394)	(0.178, 0.349)
HARD	(0.092, 0.225)	(0.193, 0.365)	(0.249, 0.445)
Overall	(0.194, 0.290)	(0.247, 0.356)	(0.238, 0.365)

ument d as $E_{\lambda=0.05} = 1 + 0.05 \cdot (E_{sent} - 1)$. As mentioned in Table 4.3, the position of the first relevant sentence in the relevant document is always larger than 2.0. Based our assumption that the reviewer read the document sequentially from the beginning to the first relevant sentence, we can infer $E_{sent} \geq 2.0$. To make this more concrete, if the number of sentences reviewed E_{sent} for d is more than 1, *sdd* can use less effort than *ddd* to achieve the same level of recall. In other words, if the time to judge a document is substantively more than judging a sentence, *sdd* is more effective than *ddd*.

4.5 Conclusion

This simulation study suggests that an active learning method can identify a single sentence from each document that contains sufficient information for a user to assess the relevance of the whole document. The best-performing active learning method selected the highest-scoring sentence from the highest-scoring document for assessment, based on a model trained using entire documents whose labels were determined exclusively from a single sentence.

If we compare the recall achieved by different strategies at a given number of assessments, there is no significant difference between the best-performing sentence-level rele-

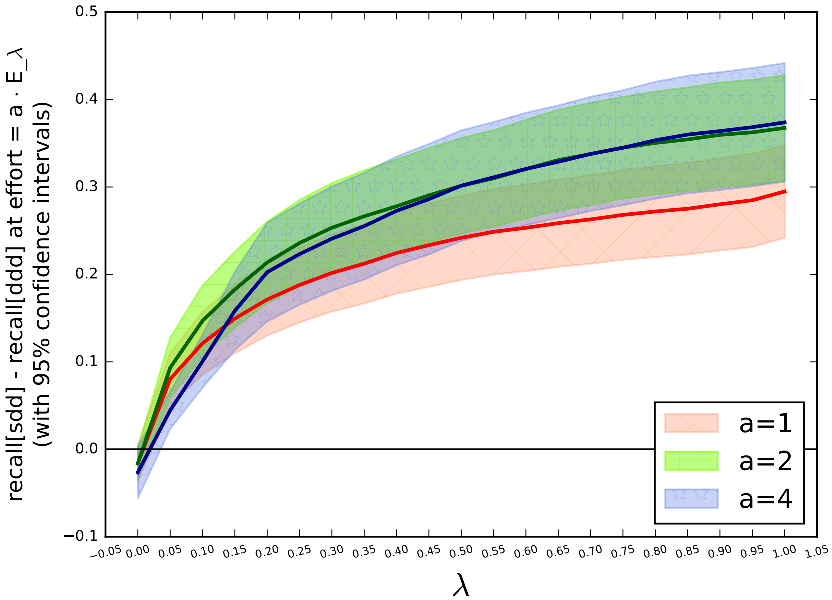

Figure 4.11: recall[sdd]- recall[ddd] at $E_\lambda = aR,$ where $a \in \{1, 2, 4\}$ by varying λ from 0 to 1 by step size 0.05 (95% Confidence interval). $E_\lambda = E_{judge}$ where $\lambda = 0$ and $E_\lambda = E_{sent}$ where $\lambda = 1$. With the increase of λ, recall[sdd] became significantly larger than recall[ddd] for all values of a.

vance feedback strategy and the best-performing document-level relevance feedback strategy. Under the weak assumption that the user can review a sentence more quickly than an entire document, the results of our study suggest that a system in which only sentences were presented to the user would achieve very high recall more quickly than a system in which entire documents were presented.

However, the relevance feedback process in this study was simulated. The synthetic labels used to simulate the relevance feedback on sentences were imperfect, but of comparable quality, according to the calibrated recall and precision, to what has been observed for human users [Voorhees, 2000]. We tried to measure the review effort needed to achieve high recall using several different evaluation apparatuses, which were also imperfect.

Chapter 5

Effective User Interaction for High-Recall Retrieval

In the previous Chapter 4, the simulation studies suggest that even a single extracted sentence may be adequate for CAL to perform well. In addition, past research has shown that human assessors are able to assess the relevance of documents faster by judging shorter document excerpts (e.g., extractive summaries) compared to judging full documents [Smucker and Jethani, 2010]. In addition, with regard to the accuracy of judgments, there is no significant loss for judging short document excerpts in place of judging full document. However, to our best knowledge, no existing controlled user study has been conducted to study whether just using document excerpts for relevance feedback in CAL is able to achieve high recall. In order to test the hypothesis that judging short document excerpts can reduce assessment time and effort to achieve high recall, we conducted a 50-person controlled user study. We designed a high-recall retrieval system (HiCAL) on the basis of the BMI implementation of continuous active learning (CAL). The HiCAL system could display either short document excerpts or full documents for human assessors to judge. In addition, we tested the value of adding a search engine into CAL. In the controlled user experiment, participants were asked to try to find as many relevant documents as possible within one hour.

In this experiment, we hypothesize that users will be able to find a larger number of relevant documents if they judge short excerpts instead of full documents within a limited time frame. In addition, we examine the effects of allowing users to compose their own queries and search documents (interactive search and judging) from search engine. Interactive search may help users find relevant documents when the CAL system has trouble finding relevant documents, Those judged document from interactive search can then be fed back into the CAL algorithm to build a better classifier and further find other relevant documents.

We base our high-recall retrieval system on Cormack and Grossman's state-of-the-art autonomous technology-assisted review (AutoTAR) method described in Algorithm 3. AutoTAR is a version of Continuous Active Learning (CAL) [Cormack and Grossman, 2014], which is an iterative relevance feedback process.

Our experimental system provides a configurable version of CAL with different options. One version of CAL can only show a paragraph-length document excerpt for judging. The other versions of CAL show the document excerpt by default and further allow the user to click to view the full document. In addition, our system could be configured to only allow users to use CAL for judgments. And our system could also be configured to allow users to switch to a search engine to retrieve relevant documents.

Based on these different configurable options, our CAL experimental system had in total four variations to support the 2×2 factorial experiment. One factor was the display of the document in the CAL component: a document excerpt alone or an excerpt plus the ability to click and view the full document. The other factor was whether or not the users were allowed to use a search engine. Any relevance judgments made from the search were fed back to the CAL's machine-learned classifier. Although BMI has shown its superior effectiveness in many tasks, the comparison between a CAL system alone and a system that combines CAL and manual searching has not been conducted in a controlled user study prior to this paper, to the best of our knowledge.

We had 50 participants involved in the experiment using our HiCAL system. For each of system variations, the participants need to find as many relevant documents as possible for a given search topic within one hour. Both the search topics and documents were from the TREC 2017 Common Core Track [Allan et al., 2017]. To evaluate performance of different variations, we used several performance measures to compare various aspects of different high-recall applications. We observed that:

- For the primary target of finding a larger number of relevant document or achieving higher recall, CAL with only paragraph-length excerpts outperformed the version of CAL that only allowed users to view full documents.
- Giving users the ability to interactively query a search engine, did not help users find a larger number of relevant documents. For some evaluation measures, the ability to conduct searches even hurt performance.
- Any value of the ability to view full documents or interactively search was offset by the significant time cost of using these time-consuming interactive features.

In the remainder of the chapter, we detail our experiment, present and discuss our results, and then conclude the chapter.

5.1 HiCAL: A System for High-Recall Retrieval

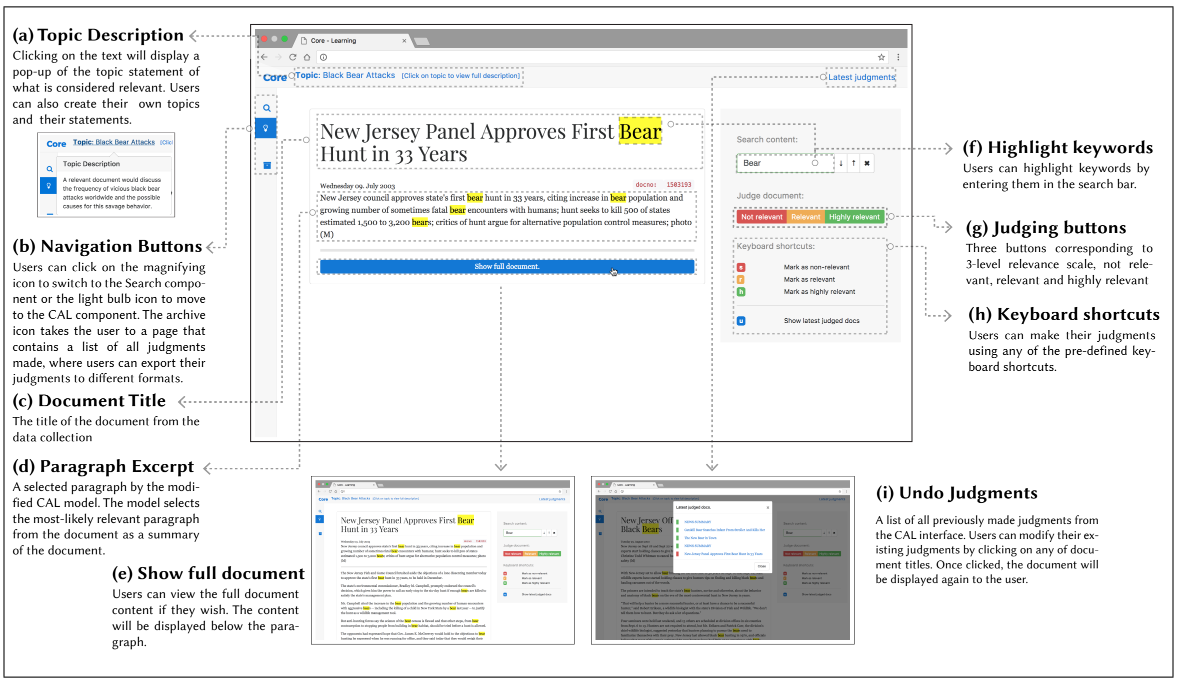

Figure 5.1: CAL user interface in the HiCAL system. The title, date, document id and a specific paragraph of the document is shown to user for judging. The user can click on the "Show full document" button to view the full document. The "Latest judgments" button enables users to review their previous 10 assessments and modify their judgments. Three judgment buttons are provided for making relevance judgments. A keyword highlight feature is provided where the user can enter keywords to highlight them.

In order to conduct the controlled user study, we implemented a high-recall-information-retrieval system named HiCAL [1]. The core part of our HiCAL system is an implementation of continuous active learning (CAL). As mentioned in the Section 2.2.2, CAL is an iterative

[1]https://github.com/hical

relevance feedback method by which the user continuously judge documents selected by a machine learned classifier. CAL selects the most likely relevant documents for assessment based on the order of relevance probability.

The CAL system has two possible configurations. In the first configuration, CAL shows a document's title, date, and a selected paragraph from the document. This configuration is shown in Figure 5.1. In the second configuration, CAL is the same as the first but allows the user to click to view the entire document. The full document is displayed below the paragraph excerpt by clicking to view the full document For both configurations, once the user has determined the document's relevance, the user can then click the three judgment buttons on the right hand side to submit the relevance judgment. After receiving the judgment, CAL retrains its relevance model, reranks the data collection, and then displays the next most likely relevant unassessed paragraph or document to user. 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.

As we detail in Section 5.1, we carefully implemented CAL and improved it efficiency so that there was no noticeable delay from submitting a judgment to receiving the next document to judge. Therefore, the users were able to review the stream of documents continuously without any delay.

As shown in Figure 5.1, a 3-level relevance scale: non-relevant, relevant, and highly-relevant is provided for relevance judgments. As noted by Harman [2011][Section 2.4.3, page 39], NIST assessors prefer a three-level judgment scale over binary decisions (i.e., relevant and non-relevant) because it makes decision making easier. In our analyses of results, both relevant and highly-relevant judgments were treated the same as relevant. We also provide keyboard shortcuts for judging in addition to the buttons. "s" is the keyboard shortcut to label non-relevant document. "r" is the keyboard shortcut to label relevant document. "h" is the keyboard shortcut to label highly-relevant document. A keyword highlight feature is also provided so that user can use the *"Ctrl + F"* shortcut to highlight the input keywords. User can separate multiple keywords by spaces to highlight each of them simultaneously.

In some cases, a user might make a judgment mistake and want to change a previous CAL judgment, our interface provides means for the users to view and modify their latest 10 CAL judgments. In the back-end, our system also records all the previous judgments made from the users.

Our HiCAL system can show CAL alone or can also provide a search engine for user to query. Figure 5.2 shows the search engine user interface in HiCAL. The users can compose their own queries and control the number of returned documents (10, 20, 50, or 100) with

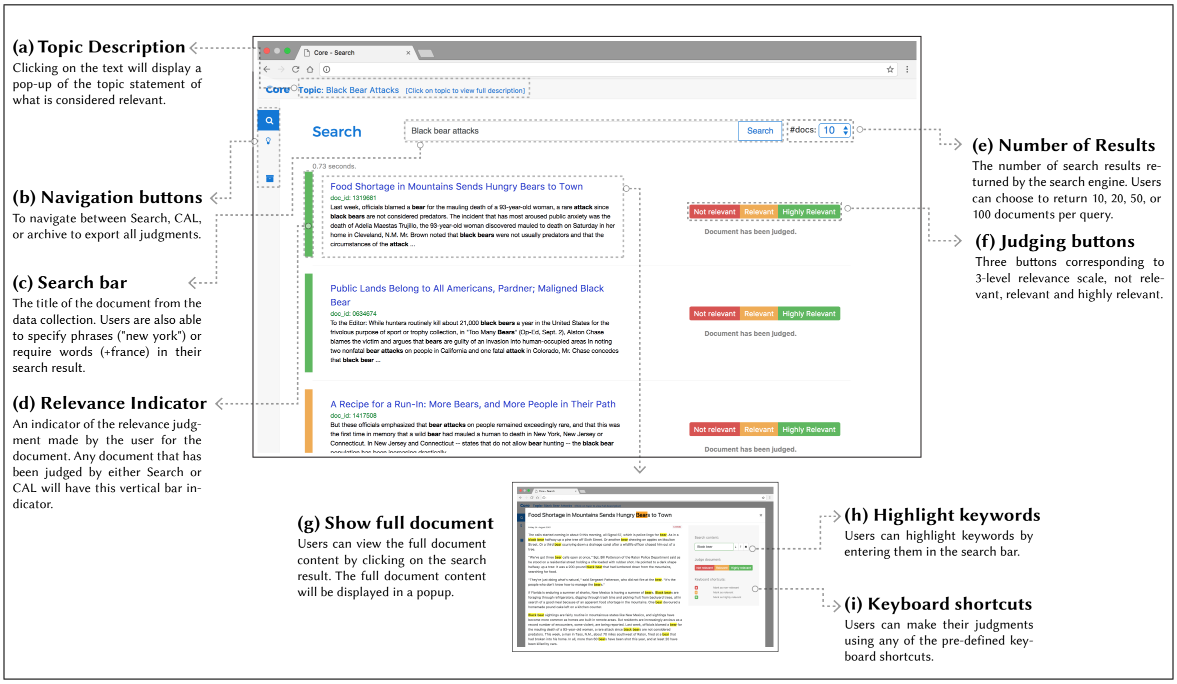

Figure 5.2: The search user interface in the HiCAL system. Users may judged documents directly from the search results or via clicking on a result to view and judge the full document. The interface has a description of the topic, a search bar, an option to select the number of results returned (default is 10), and three judgment buttons. The first result shown in the SERP is assessed “Highly Relevant”, the second is “Relevant” and the third result is “Not relevant“. Pagination is not supported for the SERP.

10 results being the default. Similar to some commonly used commercial search engine (i.e., Google or Bing), a search engine results page (SERP) is shown to user after retrieving the documents. Users can directly judge the relevance of a document based on its snippet from the SERP. Any already judged document instantly shows the user’s judgment in the SERP, and the user can freely change their previous judgments if so desired. In addition, users are also able to click on each result from the SERP to view the full document content. When clicking on a result, a full document view interface also pops up. Users are provided with the same judging interface and keyword highlighting tool as in the CAL interface. From the search interface, users have the freedom to choose which documents to judge or not. In addition, similar to other search engine, we also provide search operators for helping users to compose their specific queries. Users can specify phrases in their queries with double

ALGORITHM 5: 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 labelled document d to the training set;
Step 8. Repeat steps 2 through 7 until some stopping criteria is satisfied.

quotes ("") and can require the presence of a word with a plus sign (+).

When the system variation includes search interface, users are able to freely switch between CAL interface and search interface at any time. Two buttons on the top left corner of the interface can be clicked to switch between different interfaces.

Judgments collected from the search interface are fed into the same set of judgments made from the CAL interface. Thus, the machine learning classifier in CAL will be retrained based on all existing judgments. Moreover, the CAL interface will not show already judged documents to the user. While search may help find new relevant documents, it might also help CAL when CAL seems to get stuck and fail to find new relevant documents [Cormack and Grossman, 2015*a*].

We next detail the implementations of CAL and the search engine.

Implementation of paragraph-level CAL

For our CAL implementation, we modified the CAL algorithm used in the baseline model implementation (BMI) [Roegiest, Cormack, Grossman and Clarke, 2015] to enable the paragraph-level relevance feedback. The corresponding framework of paragraph-level CAL algorithm is shown in Figure 5.3. The rankings were performed on a paragraph-level instead of documents. In each iteration of CAL, the classifier ranks all the paragraphs and selects the highest-scoring paragraph from unassessed documents for assessor to judge. It should be noted that the training set is still built upon the documents instead of paragraphs. The

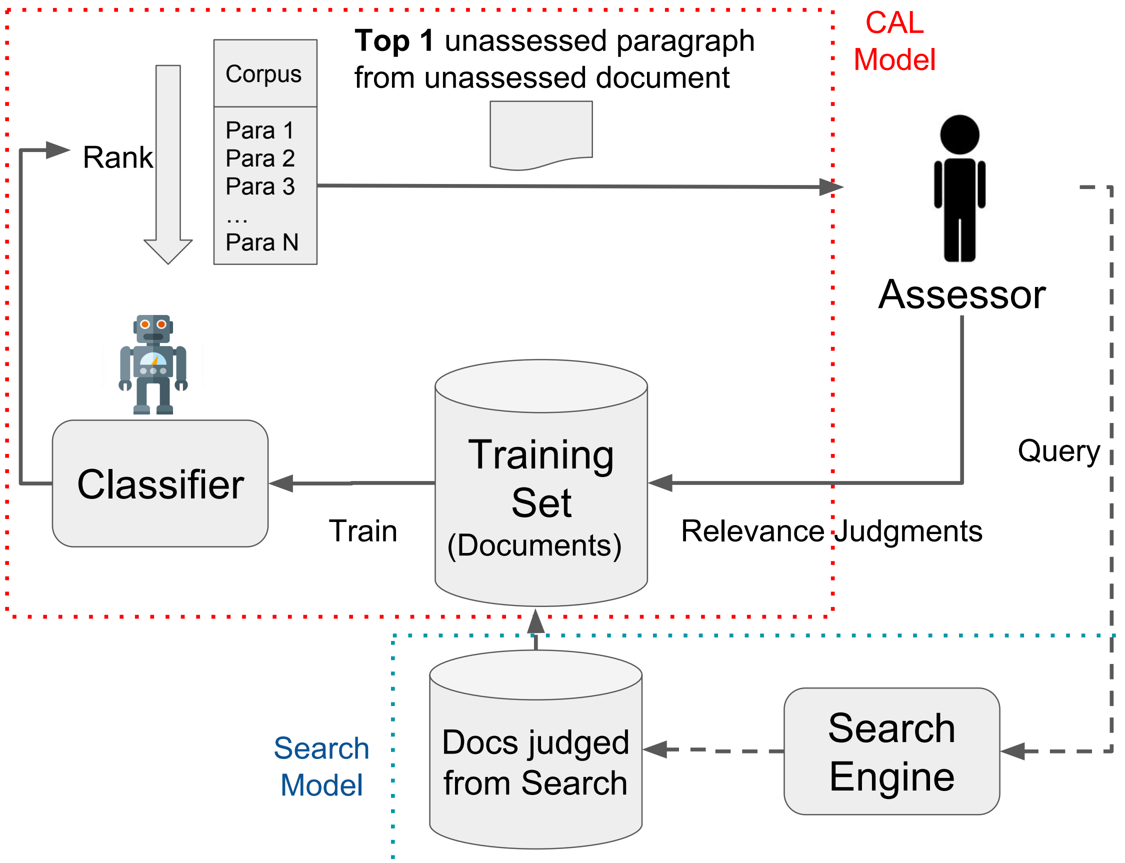

Figure 5.3: The paragraph-level relevance feedback Continuous Active Learning framework.

label of judged paragraph is used to label the document containing this paragraph. The details of the modified algorithm are listed in Algorithm 5.

For each document, we extracted the paragraphs wrapped by the $\langle p \rangle \langle /p \rangle$ tags. In total, around 30 million paragraphs were extracted from the collection, with an average of 16.7 paragraphs per document. The document's title, date, and ID were also extracted for logging and displaying purposes. The title, date, and ID of a document are shown in CAL and search interface by default.

BMI used the unigram word-based *tf-idf* as document feature vectors for training the classifier and ranking documents. 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. In out experiments, we keep the stopwords.

$$\textit{tf-idf} = (1 + \log(\textit{tf})) \cdot \textit{log}(N/\textit{df}) \tag{5.1}$$

As shown in the Equation 5.1, We calculated the *tf-idf* weight for each word in paragraphs and documents separately. *tf* is the term frequency, N is the total number of documents in the corpus, 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. We use the popular SVM-light sparse data format to store the *tf-idf* features.

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:

$$\begin{cases} \textit{tf-idf}_{t \in d} = \frac{\textit{tf-idf}_t}{\sqrt{\sum\limits_{t \in d} \textit{tf-idf}_t{}^2}} \\ \textit{tf-idf}_{t \in p} = \frac{\textit{tf-idf}_t}{\max\{20, \sqrt{\sum\limits_{t \in p} \textit{tf-idf}_t{}^2}\}} \end{cases} \tag{5.2}$$

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 5 is the concatenation the title and description of the topic. This topic statement is treated as a synthetic relevant document for training the initial classifier. Note that the classifier is trained on the documents and then ranks on the paragraphs in order to select the paragraph most likely to be relevant. An assessment on a paragraph p is considered to be the same as the assessment on document d it is part of.

The original BMI algorithm is implemented in Bash, which is suitable for simulations but inefficient for practical use. In addition to the algorithmic modifications, we reimplemented BMI in C++. The details of the implementation is described in [Abualsaud, Ghelani, Zhang, Smucker, Cormack and Grossman, 2018].

Training the classifier and scoring the entire data collection is resource intensive. The original BMI implementation performed this step after receiving a batch of judged judgments. The batch size of each iteration increased exponentially. By doing so, the computation time for training and ranking can be saved. Our new C++ implementation is

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

able to efficiently train the model and re-score the whole data collection whenever a new assessment is made from the user.

The original BMI implementation suffered from the heavy reliance on file I/O and sub-optimal intermediate operations. The classifier needs to load the document or paragraph features from hard disk for every iteration. To enable fast processing and use of efficient intermediate data structures, we stored all the feature vectors of paragraphs and documents in memory, and parallelized the computations across paragraphs and documents.

A key difference between Algorithm 5 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 unassessed highest-scoring paragraph to the assessor. Meanwhile, classifier retraining and paragraph rescoring are operated in the background. Under our experiment setup, the steps 3 through 5 can be executed 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.

Implementation of Search

For the search engine, we processed the LDC New York Times Annotated Corpus [Sandhaus, 2008] by extracting its guid, title, date, and text body from each document. 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 [Munoz and Nagarajan, 2001] packaged as part of an early version the Wikipedia Miner software [Milne, 2014].

We indexed each document's title and body using Indri [Strohman et al., 2005] and then stemmed words with the Krovetz stemmer. The retrieval model uses Indri's default parameters. To build the document snippets, we retrieve the top 2 scoring sentences from a document, concatenate them, and then truncate them to a maximum of 75 words.

The details of implementing the search interface of HiCAL system is described in [Abualsaud, Ghelani, Zhang, Smucker, Cormack and Grossman, 2018].

5.2 Experimental Setup

In this section, we describe our experiment settings in detail. We next describe the search topics and document collection, the study design, the study procedure, and other details of the experiment including how we measured performance and determined statistical significance.

5.2.1 Search Topics and Documents

We used the TREC 2017 Common Core Track [Allan et al., 2017] test collection for our search topics and documents. The Core Track provided in total 250 topics. 50 of them were assessed by NIST assessors and were provided with NIST qrels (gold standard). We used the 50 NIST assessed topics as opposed to the full set of 250 topics. The track's task was ad-hoc retrieval of documents from the New York Times dataset [Sandhaus, 2008], which includes over 1.8 million news articles.

Submitted runs were either manual or automatic runs, based on whether manual intervention is used. Runs without any type of human intervention in the query construction process are considered to be *Automatic* runs, while *Manual* runs involve human intervention (often in the form of manual judgment of documents). If an automatic run involves the use of judgments from previous tracks (i.e., Robust Track 2004), it is considered an "Automatic-Routing" run.

The topics used were originated and modified from the TREC 2004 Robust Track [Voorhees et al., 2005]. This allowed teams to train relevance 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 2017 [Zhang, Abualsaud, Ghelani, Ghosh, Smucker, Cormack and Grossman, 2017]. We ourselves used an early variant of our high-recall system to find and label relevant documents. Based on our own judgments, we submitted several manual runs. According to our usage experience with this prototype system, we modified it for the controlled user experiment that we report in this paper. We collected relevance judgments for 50 topics using only the first 10 study participants. The preliminary results from 10 users were submitted to the track as a run. By submitting this run, we hoped to increase the chance that the documents that our participants reported as relevant would also be judged by the NIST assessors. And these user found relevant documents can be part of the construction of the test collection. To reduce issues of bias, we were careful to make sure the first 10 study participants used each of the system variations. We finished running the full experiment with the remaining 40 participants after the track submission deadline. The judgments from the remaining 40 users were not used to create any of our submitted runs. We are careful to exclude our submitted runs and other related runs from our university when it matters to the analysis of experiment results in Section 5.3.2.

When we displayed search topics to participants on the system interfaces, we used hand edited versions that combined the topic's description and narrative. The reason for doing this both is to make the topic more clear and to shorten the amount of text displayed to the participant. Regardless of the retrieval system variation used by the participant, the participants all saw the same hand edited topic descriptions.

For the topic statement used in P-CAL for training the initial classifier mentioned in Step 1 of Algorithm 5, we concatenate the title and description provided by NIST as the topic statement instead of our hand edited topic descriptions.

5.2.2 Study Design

Table 5.1: 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

50 participants used four different variations of our high-recall retrieval system to find as many relevant documents as possible within one hour. All these system variations incorporated a modified continuous active learning (CAL) model derived from the TREC Total Recall Track's baseline model implementation (BMI). In this study, two factors were investigated and each of them had two levels, i.e., a 2×2 factorial design. The first factor determined whether participants using the CAL interface of the system would judge a paragraph-length excerpt of a document or given the ability to click to view the full documents. The other factor determined whether or not search was made available to the participants. Judgments made from the interactive search were fed into training set. The CAL system then could use these judgments from both search and CAL interfaces to learn the relevance model.

Table 5.1 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;
- **CAL-D&Search**: CAL with full documents and search

Each participant completed 5 tasks, and each of these 5 tasks was associated with a unique search topic. For 4 tasks among the 5 tasks, the participant used one of the four system variations as per Table 5.1 to find as many relevant documents as possible within one hour. For the fifth task, the participant judged the relevance of 60 sampled documents. These 60 document were randomly selected based on their likelihood of being relevant as determined by a relevance model trained base on our own judgments. We call this as the ***reference*** treatment. For this treatment, we showed participants the full document and the participants had to judge the document's relevance one by one until finishing judging all 60 documents. This judgment mode is similar to many traditional relevance judging tasks. There was no time limit for this reference treatment. We use this treatment to compare user behaviour on a traditional relevance judging task to user behaviour on the other four treatments. The results of this comparison is described in Chapter 6. By the end of the experiment, each system variation had been applied once to each of the 50 topics.

In order to cover all 5 treatments on each topic, we created a balanced study design as follows. We first divided the 50 topics into 10 blocks of 5 topics each. For each block 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 each cell of the squares, and then the squares were randomized. After running the experiment, we discovered that the topics were not randomly shuffled into groups of 5 but were instead assigned to groups in their numeric order. While there might exist some association between topics and their given number, we do not think this lack of randomization is a concern.

By balancing the design, we ensured that:

- Each user was assigned five tasks such that no two tasks had the same topic or treatment. Therefore, each user covered all the 5 treatments.
- Each topic was paired with all 5 treatments. Therefore, across all users, each topic was covered by 5 different treatments.

5.2.3 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 signing their consent form to participate in the study, each participant went through an in-person tutorial. The tutorial covers the installation of our HiCAL 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 of graded relevance which includes non-relevant, relevant, and highly relevant [Voorhees, 2001*a*]. Following the convention of TREC relevance, we told participants that a document should be labelled as relevant if any portion of it is relevant. We also told participants to be consistent in their judgments and not to adjust their notion of relevance. We warned participants to not just rely on keywords for making judgments. In some cases a document might contain a lot of keywords, it could be non-relevant.

We picked two topics from the TREC 2004 HARD Track [Allan, 2004] to give participants practice making graded relevance judgments. For one topic, participants judged 6 documents (i.e., 2 relevant documents, 2 highly relevant documents, and 2 non-relevant documents). The participants discussed with the researcher any differences between their judgments and the NIST judgments. For the second topic, the same process was followed, but only a paragraph-length excerpt from document was shown to the participants.

When it came to assessment effort, we asked participants to "work as fast as possible while maintaining your accuracy." We made clear that 4 tasks were requiring 1 hour of work and a timer in the backend was used to record their active judgment time. 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 and we would check their judgment quality. 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 help participants familiarize with the system. Both search and CAL interfaces were thoroughly explained. Each participant used both interfaces to find relevant documents during the practice task. One of the non-NIST judged Common Core topics was used 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 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 five assigned tasks on their own. We asked participants to try and finish all 5 tasks within 5-7 days. During the study, participants could choose to take a break or continue their progress whenever they wanted. We encouraged the participants to try finishing one task without take any break in the middle.

The participants were able to work on their own computers in whatever locations or environment they preferred. We made this choice mainly because 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. Allowing participants to work on their own also gives us a sense of how crowd-sourced workers might perform at this task.

Because participants work on their own computer, our system is engineered to monitor their activity and only count their active working time. 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 return to the task. The timer continues to record time until the participant close the pop-up dialog box. We did not count those 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 one hour's worth of active work has been reached. Unfortunately, our software allowed some participants to work in excess of one hour on some tasks. To make sure we only measured performance within 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 the full user study, they first answered a demographics questionnaire. Then they performed their 5 search tasks (see Section 5.2.2). Each task included a pre- and a post-task questionnaire. After completing all five tasks, we had participants answer an exit questionnaire to collect their feedback and overall experience. Once finished, participants returned to be paid $100 for their participation.

5.2.4 Participants

Before conducting the 50 person full study, we completed a pilot test with two participants to discover 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.

5.2.5 Performance Measures

As discussed in the Section 2.1, there are many tasks that require high-recall retrieval. We consider that they can be categorized into two classes. 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, the documents judged by the user as relevant are considered to be the result set. Although there may exist some values in examining the documents judged non-relevant, our study participants were trying to avoid finding non-relevant documents. In addition, some users only mark relevant documents from search engine and they do not provide any useful non-relevant judgments.

Tasks such as eDiscovery and systematic review usually contain two passes of relevance judging. The first pass could be conducted by someone qualified to identify relevant material, such as junior assessors. The second pass would be conducted by an expert who examines the judgments from the first pass and makes a final determination about which documents are relevant. For example, in systematic review, a lead researcher might guide graduate students to the task of finding all relevant research papers. The graduate students give the lead researcher the documents judged relevant. The lead researcher then examines each document and makes the final decision on which documents are relevant and should be included in the review.

Both eDiscovery and systematic review tasks want to find all relevant documents. Any missing relevant document could result in legal issues for eDiscovery or could affect the conclusions made by a systematic review. Based on the two passes of review process, a good first performance measure is simply the number of relevant documents found and reported by the user, U_{rel}. Given that different search topics have different numbers of relevant documents, it can be helpful to normalize the number of relevant documents user found. Therefore, we use recall as our normalized measure of performance. Recall is the fraction of all relevant documents found by a user according to the gold standard qrels:

$$recall = \frac{|U_{rel} \cap R|}{|R|}, \tag{5.3}$$

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 assessors (gold standard). The NIST assessors act as the experts who determine the final relevance of those user judged relevant documents.

In the cases where two passes are needed to find relevant documents, each non-relevant document that is labelled as relevant by the first pass wastes the time of the final reviewer in the second pass. Therefore, we measure the precision of the set of documents returned by the first pass as another useful measure:

$$precision = \frac{|U_{rel} \cap R|}{|U_{rel}|}, \tag{5.4}$$

F_1 is a useful standard measure that combines both recall and precision and captures the tradeoff between them:

$$F_1 = \frac{2 \times recall \times precision}{recall + precision}. \tag{5.5}$$

The task of IR test collection construction, in most cases, also requires to find all relevant documents. Assessment error made in judging and missing relevant documents affect not only the values of effectiveness measures, but also affect the ability to correctly rank different retrieval systems. We collect the runs submitted to the 2017 TREC Common Core track. We eliminate the runs we submitted to the track as well as related runs from another group at our university [Zhang, Abualsaud, Ghelani, Ghosh, Smucker, Cormack and Grossman, 2017; Grossman and Cormack, 2017]. We compare the ranked list of these runs using the relevance judgments produced by our study participants to the ranked list produced with the NIST qrels. For measuring the retrieval effectiveness or quality of a run, we use mean average precision (MAP).

As described in Section 2.3.1, the most common measures of ranking correlation are Kendall's rank correlation coefficient, τ, and Yilmaz, Aslam and Robertson [2008]'s τ_{AP}. The τ_{AP} measure places more weights on high scoring runs. We use Urbano and Marrero [2017]'s implementation of τ_{AP}.

In some cases, it is possible to rank systems well enough but produce scores that are very different from the score produced by the NIST qrels. To measure the difference between MAP scores from user judgments with MAP scores from NIST qrels, we compute the root mean squared error (RMSE):

$$RMSE = \sqrt{\frac{\sum_i^n (nist.map_i - t.map_i)^2}{n}}, \tag{5.6}$$

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

5.2.6 Statistical Significance and Modeling

We used generalized linear mixed-effects models (GLMMs), as implemented in the lme4 [Bates et al., 2015] package in R [R Core Team, 2014], to measure the statistical significance of our results. The GLMM is an extension of generalized linear models to include both fixed and random effects (hence mixed models). We treat our study participants and the search topics as random effects. The independent variables (factors) of our experiment were fixed effects. The two factors were whether CAL was with only paragraph-length excerpts or with the option to view full documents, and whether or not a search engine was available. The dependent variables are the various performance measures described in Section 5.2.5. 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 separately. We then compare these two models using a likelihood ratio test that reports a p-value.

5.3 Results

In the study, participants used four treatments of our high-recall retrieval system to find as many relevant documents as possible within one hour. On the condition that our study participants only able to work for one hour, we did not expect them to achieve high recall on average. The most essential analysis 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 assessment or whether the CAL component would not only show the excerpt but 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 from the search engine are used to train the relevance model in 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 5.1). As for the ***reference*** treatment, we regard this treatment as a traditional relevance judging task. We describe the comparison of user behaviour on this treatment with other 4 treatments in Chapter 6.

5.3.1 Main Results

Table 5.2 is a key/primer to help understand our tables of performance measures for those unfamiliar with this style of reporting. The table format mirrors the 2×2 factorial experi-

Table 5.2: Key/primer for reading Tables 5.3 and 5.5.

<table>
<tr><td rowspan="2">Search Available</td><td colspan="2">CAL types</td><td colspan="2" rowspan="2">Marginal means (search)</td></tr>
<tr><td>Full doc available</td><td>Paragraph Excerpt only</td></tr>
<tr><td>No</td><td>CAL-D Average</td><td>CAL-P Average</td><td>Average without Search</td><td rowspan="2">p value (Search vs. No Search)</td></tr>
<tr><td>Yes</td><td>CAL-D&Search Average</td><td>CAL-P&Search Average</td><td>Average with Search</td></tr>
<tr><td rowspan="2">Marginal means (CAL types)</td><td>Average of Full doc available</td><td>Average with only Paragraph excerpt</td><td colspan="2" rowspan="2">Overall Mean</td></tr>
<tr><td colspan="2">p value (Full doc vs. Paragraph)</td></tr>
</table>

ment design of Table 5.1. For each combination of factors, we report the mean performance over 50 search topics. 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, the overall mean of all the 4 treatments is reported. In our analysis, we are particularly curious about the effect each factor has on the performance measures, As described in Section 5.2.6, 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 or U_{rel}. Table 5.3a shows these results. In this case, both factors yield statistically significant differences. Only showing a paragraph-length excerpt in CAL can help find significantly larger number of relevant documents being reported by participants. Likewise, a CAL-alone system without search interface is significantly superior than a CAL system that includes search interface. Participants using CAL-P found on average 97.9 relevant documents within one hour, which is a almost 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 junior reviewers to find relevant documents, and a second pass where the relevant documents from the first pass are further verified by an expert. Table 5.3b reports a similar scenario in which the first pass is by our study participants and the second pass is verified by the

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

NIST assessors. If a document is labelled as relevant by both our participants and NIST assessors, this document is a user found NIST relevant document. If a document is not assessed by the NIST assessors, we assume it to be non-relevant. Shown in the Table 5.3b, CAL with paragraph-length excerpts has statistically significant better performance over CAL with the option to show full documents ($p < 0.001$). The availability of search engine hurts performance. However, it is not a statistically significant effect ($p = 0.065$). We observe that search slightly helped CAL with full document available while it caused a large decrease in performance for CAL with paragraphs.

Given that different topics have different number of relevant documents, we should normalize each topic according to the number of relevant documents on it. Otherwise, the number of relevant documents found would be skewed to the topics which have a high proportion of relevant documents. We did the normalization by computing recall, which Table 5.3c 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 ($p = 0.04$). As with the number of user found NIST relevant document, search has helped CAL with documents and hurt CAL with paragraphs in terms of recall.

5.3.2 Ranking of IR Systems

Table 5.4: Performance measures for the task of test collection construction (see Section 5.2.5). 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 5.1) 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]

Another important use of high-recall retrieval systems is to evaluate different IR systems using the relevance judgments. Each set of judgments produced by our four system treatments was used to score the TREC 2017 Common Core runs. Each run is scored

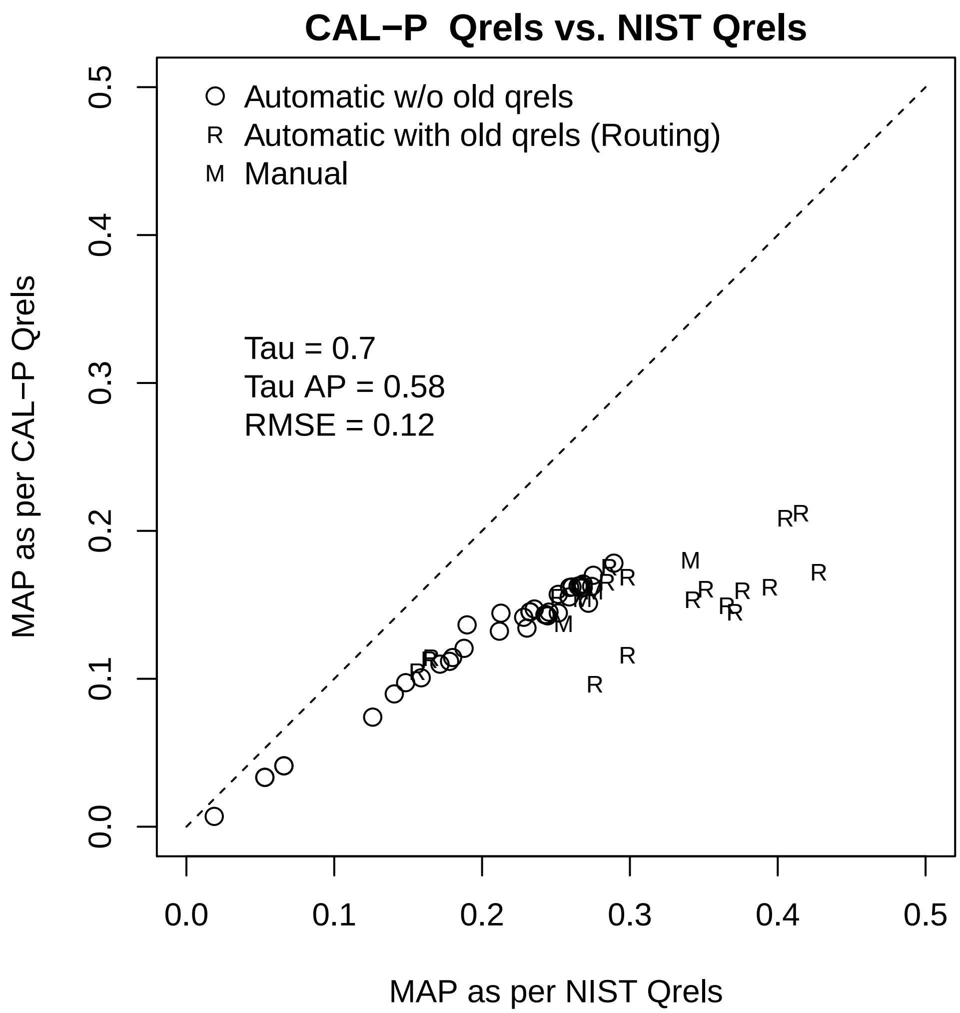

Figure 5.4: MAP evaluated by qrels from CAL-P compared against MAP evaluated by NIST qrels over runs from the TREC 2017 Common Core Task.

using mean average precision (MAP). We excluded our runs and related runs from our university [Zhang, Abualsaud, Ghelani, Ghosh, Smucker, Cormack and Grossman, 2017; Grossman and Cormack, 2017]. Table 5.4 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 paragraphs and without search (CAL-P) performed the best at ranking the IR systems with the highest τ and τ_{AP} scores. The corresponding scatter plots com-

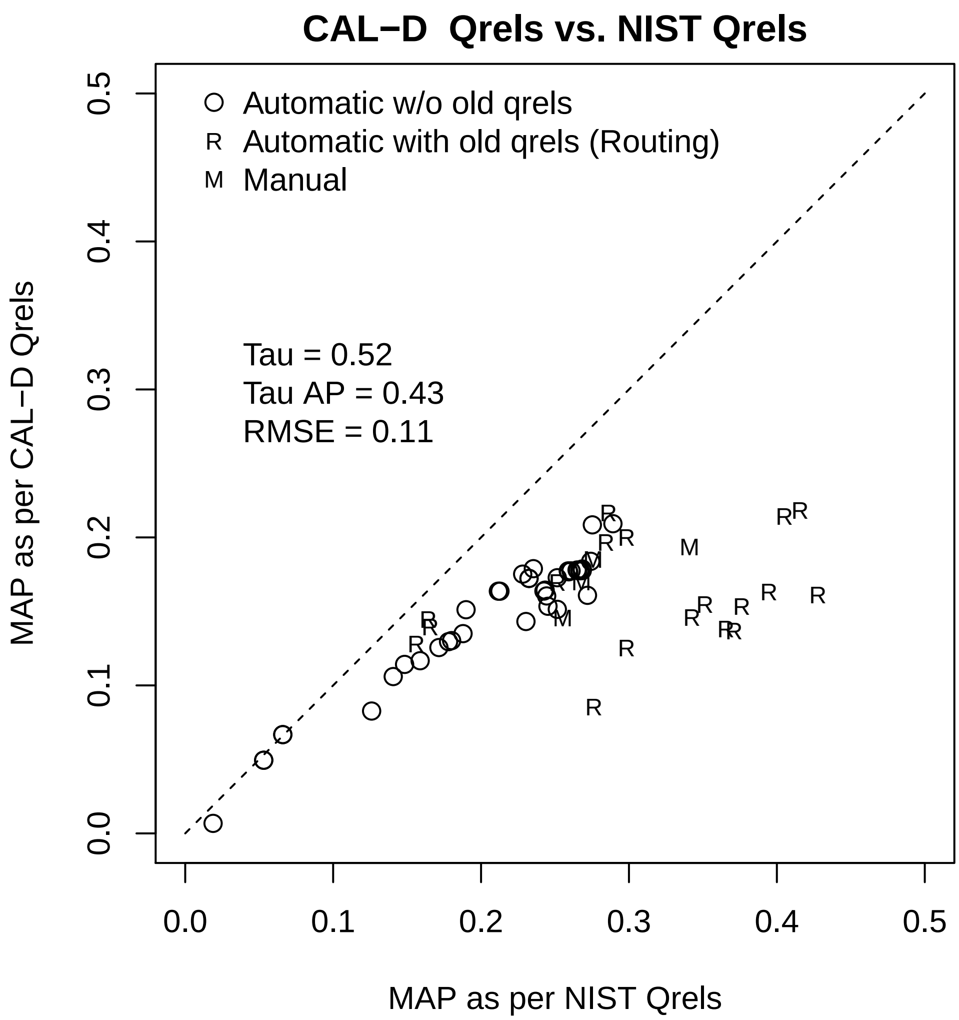

Figure 5.5: MAP evaluated by qrels from CAL-D compared against MAP evaluated by NIST qrels over runs from the TREC 2017 Common Core Task.

paring the MAP scores measured from different judgment sets are shown from Figure 5.4 to Figure 5.8.

5.3.3 Secondary Results

Besides the primary results we report in Section 5.3.1, we also observe some other interesting results. They might not directly contribute to our main finding. But they reflect

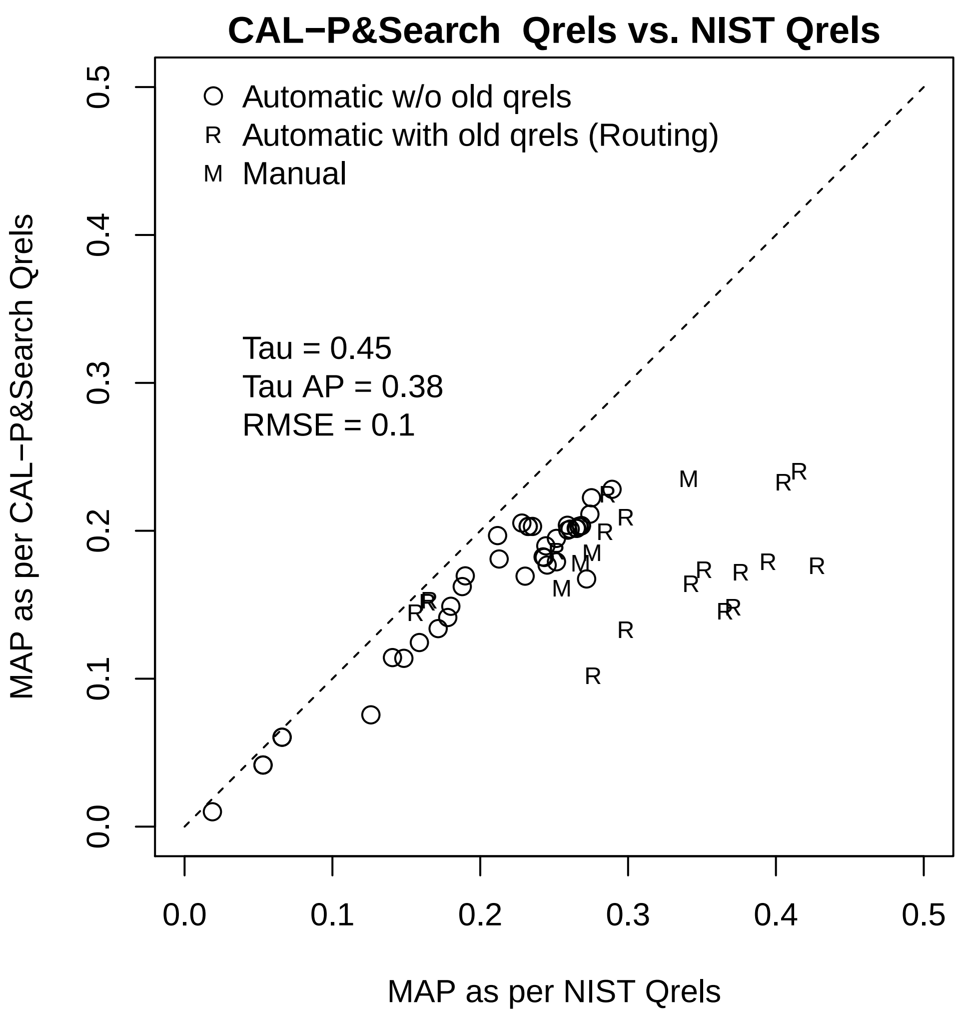

Figure 5.6: MAP evaluated by qrels from CAL-P&Search compared against MAP evaluated by NIST qrels over runs from the TREC 2017 Common Core Task.

user assessment behaviours from other perspective.

Table 5.5a reports the mean precision for each treatment, but these figures need to be explained with caution. The Figure 5.9 showed that precision of CAL with paragraphs, and CAL with documents were effectively the same, but Table 5.5a shows CAL with full documents available has a better precision at a statistically significant level. The issue here is that precision is being measured over different sets and amounts of judgments. In high-recall retrieval systems, the relevant documents would be easier to find in the beginning

Search Available	CAL types: Full doc available	CAL types: Paragraph only	Marginal means (search)	
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: Full doc available	CAL types: Paragraph only	Marginal means (search)	
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: Full doc available	CAL types: Paragraph only	Marginal means (search)	
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 5.5: The secondary results of comparing four system variations. We have marked with a $*$ the differences that are significant at $p < 0.05$.

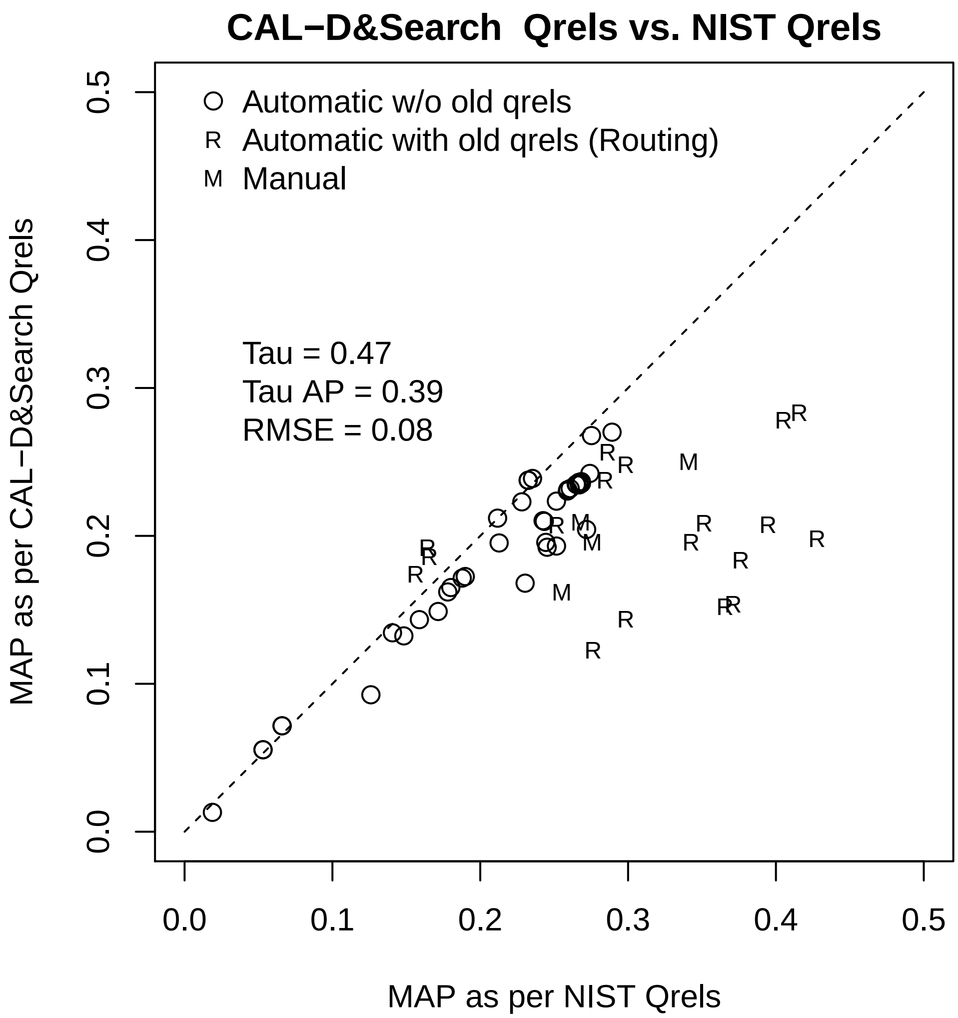

Figure 5.7: MAP evaluated by qrels from CAL-D&Search compared against MAP evaluated by NIST qrels over runs from the TREC 2017 Common Core Task.

stage and be harder as the retrieval process continues. CAL with paragraphs allows users to make judgments much faster and thus explore deeper of the documents collection. As the prevalence of relevant documents experienced by the user decreases, it is possible to expect the user to falsely assess non-relevant documents as relevant at higher rates.

To more fairly compare the precision of different treatments, we report the mean precision measured on a reduced judgment set in Table 5.5c. We consider only the first k documents that the participant reported as relevant as the reduced judgment set. For

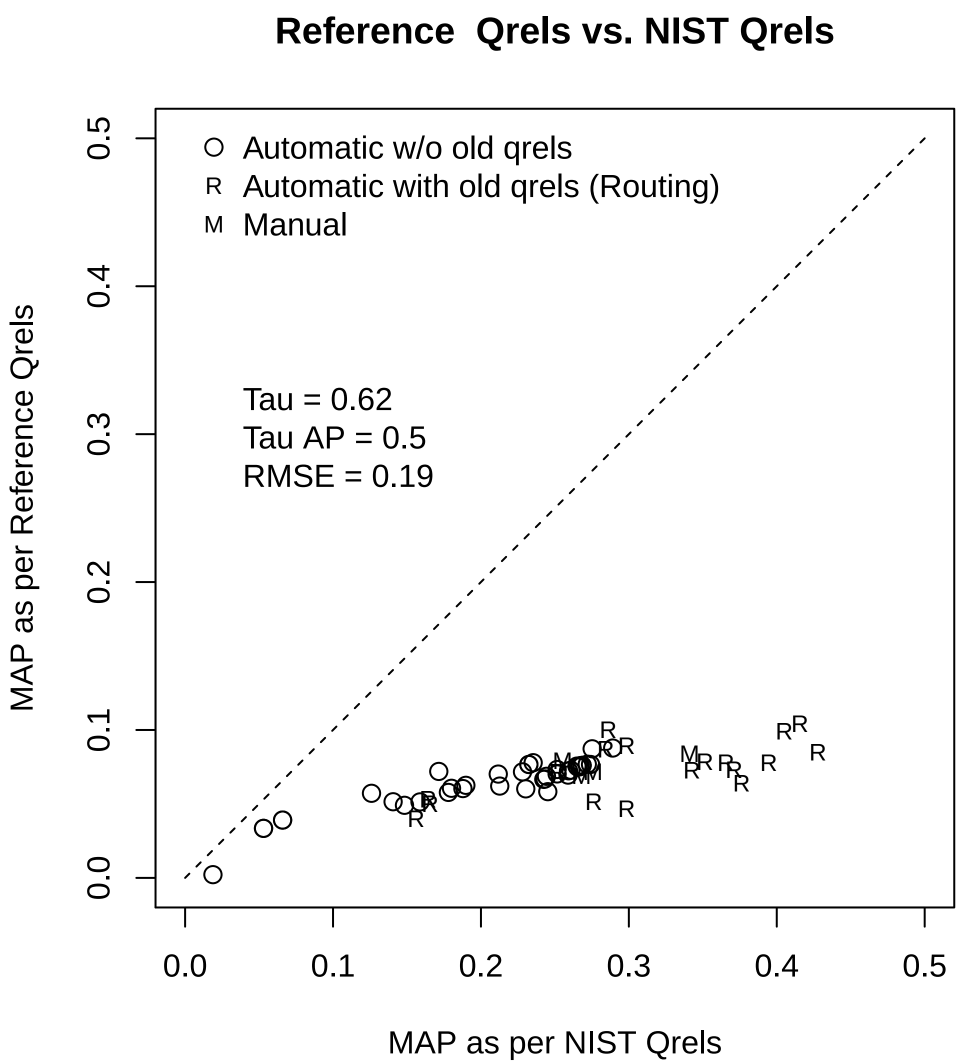

Figure 5.8: MAP evaluated by qrels from reference treatment compared against MAP evaluated by NIST qrels over runs from the TREC 2017 Common Core Task.

a given topic, k is the minimum total number of relevant documents that participants reported across all the four treatments. We noticed there was no statistically significant difference for precision at k judgments with or without the option to use search engine, and with or without the ability to view full documents.

To consider the tradeoff between recall and precision, we report the average F_1 in Table 5.5b. CAL with paragraphs is better than CAL with documents at a statistically significant level. While search improves precision, the increase of precision is not able to offset the loss of recall. As with recall, search improves the F_1 of CAL with documents but hurts CAL with paragraphs, and there is a statistically significant interaction effect between CAL and search ($p = 0.03$).

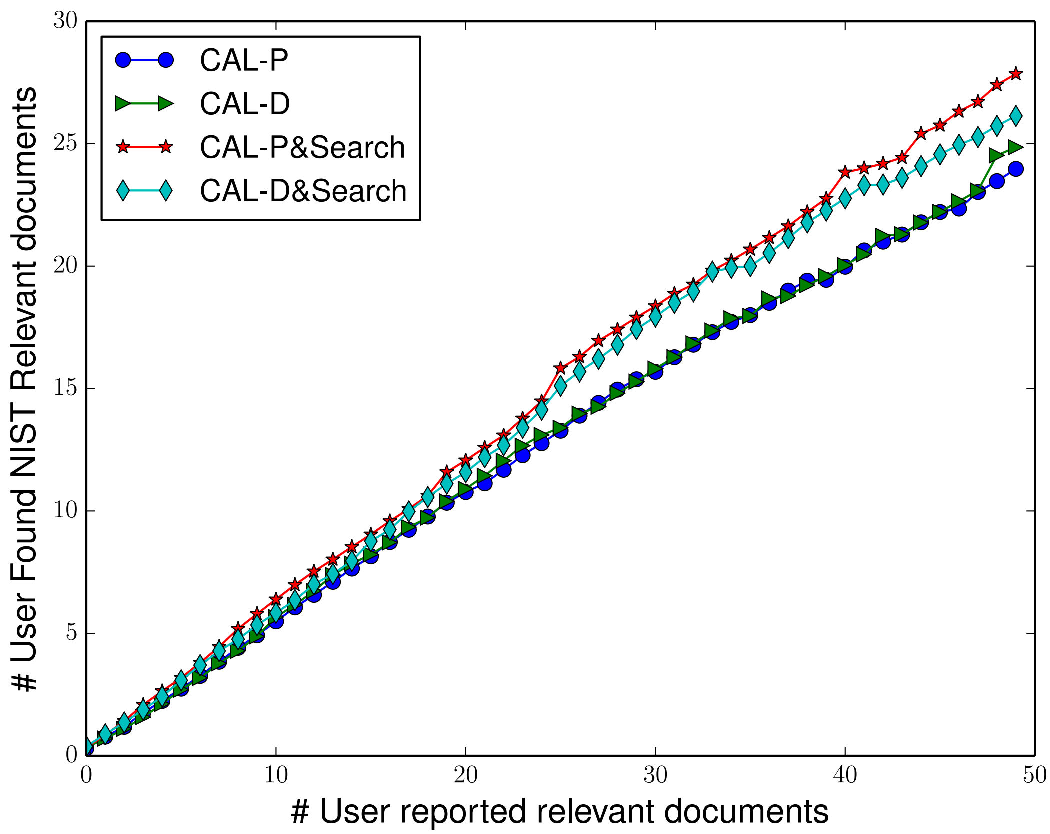

Figure 5.9: Average number of participant found NIST relevant documents vs. number of self-reported relevant documents for the first 50 self-reported relevant documents.

Since the option to use search engine 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 excessive wrong assessments during search or because of the lower rate of judgments caused by search. 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 relevant documents increases, there are fewer participants in the average. Figure 5.9 shows this analysis. The slope of each curve in Figure 5.9 is the precision of each treatment.

In Figure 5.9, the first thing noticeable is that both CAL-P and CAL-D have almost the same precision (the slope of curve). As participants report finding relevant documents, the fraction of NIST relevant documents are nearly the same for CAL-P and CAL-D. The second thing we observe is that when search is available, the precision performance improves regardless of what type of the CAL is. 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.

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

Figure 5.4 to Figure 5.7 show the scatter plots comparing the macro average precision (MAP) scores evaluated by relevance set from four different treatments with the MAP scores evaluated by the gold standard NIST qrels. We report common measures of ranking correlation: Kendall's τ, τ_{AP}, and root mean squared error (RMSE) for MAP scores. Different from τ, τ_{AP} places more weight on high scoring runs [Yilmaz, Aslam and Robertson, 2008; Urbano and Marrero, 2017]. RMSE in Equation 5.6 measures the difference between MAP scores $t.map$ from each treatment's judgments with MAP scores $nist.map$ from NIST qrels.

From Figure 5.4, we found that MAP scores from the CAL-P judgment set achieved the highest τ and τ_{AP} rank correlation with the MAP scores produced from NIST qrels. This result is consistent with the results in Table 5.3 where we found that CAL with documents is worse than CAL with paragraphs in terms of number of relevant document found. Nevertheless, MAP scores produced by the CAL-D&Search had the lowest RMSE compared to MAP scores from NIST qrels. The addition of using search engine appeared to help run scores on average align with the scores produced with the NIST qrels (lower RMSE). We hypothesize that search may be able to find some high value relevant documents that CAL had not yet found within on hour. All system treatments 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.

It it worth to note that the judgment set from ***reference*** treatment can yield a quite high correlation with NIST qrels shown in Figure 5.8. Both $\tau = 0.62$ and $\tau_{AP} = 0.5$ are higher than that of CAL-D, CAL-D&Search, and CAL-P&Search. However, as defined in our experiment, only 60 documents were sampled for judgments in ***reference*** treatment. We hypothesis that these 60 sampled documents could represent the distribution of relevant documents well. Therefore, it can effectively evaluate different retrieval systems and yield reasonable rankings. We leave the test of this hypothesis for future work.

5.4 Conclusion

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

High-recall information retrieval (HRIR) makes high reliance on the user. A state-of-the-art HRIR system 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 a larger number of relevant documents with one hour than other versions of our system that gave the participants more freedom to examine full document content and to search for relevant documents. There may be cases in which users refuse to be so limited and require that they 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 variation 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.

Chapter 6

Assessing Behaviour in High-Recall Retrieval

In Chapter 5, we conducted a 50-person controlled user study to test the hypothesis that assessing shorter document excerpts will find a larger number of relevant documents within a limited time frame than assessing full documents. We also tested the value of integrating a search engine with CAL. We found that users were able to find a significantly larger number of relevant documents by assessing shorter document excerpts. The ability to use a search engine and to view full documents slowed users down and resulted in lower performance. In this chapter, we dig into user behaviour to try to understand how these user interactions slow down users' assessments. With this aim, we compared the time spent on assessing each document under different system variations. In some treatments of our experiment, users were given the freedom to use a search engine or to view the full document. Investigating the usage and behaviour of using a search engine and viewing full documents can help us better understand how users utilize these features and spend their time. After the user study, we also asked for users' feedback on each system-provided feature (e.g., keyword highlighting, judgment shortcuts or search interface). Analysis of the users' subjective feedback regarding these system features can help us understand users' preferences, and further improve our high-recall retrieval system.

6.1 Evaluating User Assessment Behaviour

6.1.1 Assessment Speed

In the previous section, the experimental results suggested that participants would find a greater number of relevant documents by judging the relevance of documents on the basis of viewing only paragraph-length excerpts rather than full documents. In addition, we found that a CAL system without search performs as well as or better than a CAL system augmented with interactive searching. We can infer that allowing participants to view full documents slows down their rates of making assessments. Likewise, because the number of relevant documents found decreases when search is available, we can also infer that people on average find relevant documents at a slower rate via searching than via CAL.

In this section, we further measure the time spent on each judgment and the time required to find each relevant document using different treatments. Table 6.1a shows the time required to make a single assessment using four system variations and the reference treatment. For each topic, we calculated the time spent on a document by dividing one hour by the number of judgments using a specific treatment. Then, for each treatment, we averaged the time over 50 topics. We report both the median time and the average time with a 95% confidence interval over 50 topics.

The results show that the time cost to judge a document using CAL-P is the lowest within four system variations. Using CAL-P, participants took an average of only 22.7 seconds to submit each relevance judgment while they took an average of 56.8 seconds using CAL-D. Adding search to CAL can further slow down the rate of finding relevant documents. With CAL-P&Search, participants spent an average of 35.4 seconds on each judgment. CAL-D is so slow that adding search does not even slow it down further, using CAL-D&Search, participants spent 54.1 seconds on each judgment. The median time per single assessment shows the same trend as the mean time.

We also measured the judgment time participants spent on assessing documents from the reference set. The average time spent on reference set is close to the time spent using CAL-D or CAL-D&Search. Each judgment using the reference treatment took an average of 50.0 seconds. The judgment mode of reference treatment is similar to that of traditional judging, in which an assessor takes around one minute to make a single judgment.

Table 6.1b shows the median and mean time required to find a user-reported relevant document. For each topic, we calculated the time by dividing one hour by the number of user-reported relevant documents and then, for each topic, averaged the time over 50 topics. The results show that using CAL-P, participants spent the lowest median time

Treatment	Median time (s)	Mean time (s)
CAL-P	19.2	22.7 [18.6, 26.8]
CAL-D	48.4	56.8 [45.0, 68.6]
CAL-P&Search	32.2	35.4 [29.2, 41.6]
CAL-D&Search	48.3	54.1 [43.7, 64.5]
Reference	44.9	50.0 [40.6, 59.3]

(a) Time to make a judgment.

Treatment	Median time (s)	Mean time (s)
CAL-P	48.0	97.8 [28.0, 167.5]
CAL-D	90.1	122.2 [90.7, 153.6]
CAL-P&Search	65.5	92.2 [66.9, 117.5]
CAL-D&Search	92.3	112.0 [91.1, 132.9]
Reference	136.5	187.2 [148.7, 225.7]

(b) Time to find a user-reported relevant document.

Treatment	Median time (s)	Mean time (s)
CAL-P	128.6	337.0 [134.3, 539.7]
CAL-D	200.6	557.2 [298.3, 816.0]
CAL-P&Search	138.7	351.8 [156.0, 547.6]
CAL-D&Search	163.6	238.6 [177.1, 300.1]
Reference	256.1	498.3 [349.0, 647.6]

(c) Time to find a NIST relevant document.

Table 6.1: Median and mean time per judgment/relevant document using different treatments.

(48.0 seconds) to find a user-reported relevant document. Adding the ability to view full documents (CAL-D) resulted in a judgment time of 90.1 seconds and using a search engine (CAL-P&Search), 65.5 seconds. However, the resulting mean time using each treatment differs from the median time. The CAL-P&Search produced the lowest mean time (92.2 seconds) for finding a user-reported relevant document while CAL-P's mean time was 97.8 seconds. If we take the 95% confidence interval into consideration, we find that the time differences between all these four treatments are not significant. The big difference between mean time and median time ($mean > median$) suggests that the distribution of time is skewed to one side. In this case, some participants took a large amount of time to find user-reported relevant documents.

We also measured the time required to find a NIST relevant document. For each topic, we calculated the time by dividing one hour by the number of user-found NIST relevant documents and then, for each treatment, averaged the time over 50 topics. Table 6.1c shows the median and mean time results. Similar to the results in Table 6.1b, CAL-P achieves the lowest median time (128.6 seconds). Using CAL-D took 200.6 seconds and using CAL-P&Search took 163.6 seconds. The resulting mean time using each treatment differs from the median time. But there is still no statistically significant difference in the mean time to find a NIST relevant document using different treatments.

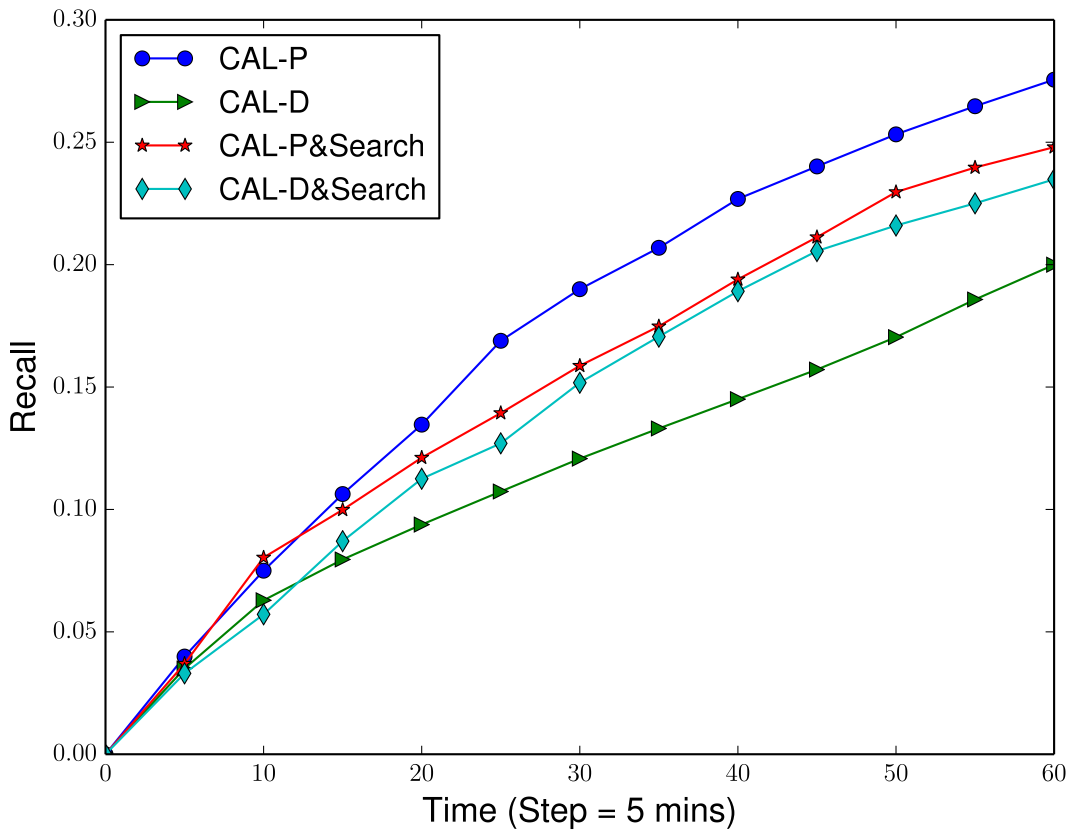

Figure 6.1: Judging time (minutes) vs. recall, using different treatments.

Figure 6.1 shows the recall achieved at different points of time during one hour using different treatments. Every 5 minutes, we measured the recall achieved on each topic and then, for each treatment, averaged the recall at a given point in time over all 50 topics.

The results show that CAL-P always helps participants achieve higher recall within one hour compared to other system variants. CAL-P&Search and CAL-D&Search rank second and third, respectively. CAL-P&Search performs slightly better than CAL-D&Search. In the first 10 minutes, CAL-P&Search achieved almost the same effectiveness as CAL-P and even outperformed CAL-P from the 5- to 10-minute mark. In contrast, CAL-D has the lowest recall throughout almost the whole process. CAL-D achieved higher recall than CAL-D&Search in first 10 minutes but became less effective after 10 minutes. The ability to view full documents slowed down the rate of finding relevant documents.

Combining the results of Figure 6.1 and Figure 5.9, we can infer that CAL-P is the most effective system variant for helping participants find relevant documents within a limited time frame. Conducting searches does increase the precision of participants' judgments. However, search takes more time to find a relevant document than simply judging short document excerpts selected by CAL. For the purpose of achieving high recall within a limited time frame, the higher precision of judgments achieved by using searches does not cannot help participants achieve higher recall.

6.1.2 Usage of Viewing Full Documents in the CAL Model

Our experimental results support our hypothesis that users working for a given amount of time would find a greater number of relevant documents by viewing only paragraph-length excerpts rather than full documents. The only difference between CAL-P and CAL-D is that CAL-D allows the user to view the full document in addition to the paragraph-length document excerpt. Given that we expect judgment quality based on full documents to be as good as or even better than judgment quality based on paragraphs, it is apparent that allowing assessors to view full documents slows down their rate of judging. In this section, we analyze user behaviour when viewing full documents in the CAL model of CAL-D and CAL-D&Search. To do so, we measured the frequency with which users viewed full documents in CAL and to what extend viewing full documents slowed down assessment rates compared to just assessing short document excerpts.

As shown in Table 6.2, we found that users on average made 73% (CAL-D) and 63% (CAL-D&Search) of total judgments using CAL by viewing the full document. This shows that users like to view the full document in most cases. For those assessments based on viewing the document excerpt only in CAL, users spent on average 13.2 (CAL-D) and 12.3 seconds (CAL-D&Search) on assessing each excerpt. In contrast, for the assessments made by clicking to view the full document in CAL, users spent on average 52.7 (CAL-D) and 44.1 (CAL-D&Search) seconds on making a judgment. In short, viewing a full document cost significantly more time than viewing just a paragraph-length excerpt. We also measured the decision time users took from reading an excerpt to clicking on the view full document button. Decision time was on average 12.5 (CAL-D) and 9.9 (CAL-D&Search) seconds. Time spent on making this decision was almost the same as the time spent on each document excerpt, which means that users read the document excerpt first and then decide whether or not they need to further read the full document.

Treatment	Fraction of total judgments from CAL by viewing full doc	Avg. time judging each paragraph from CAL model	Avg. time judging full documents from CAL model	Avg. time to click on view full document button on CAL model
CAL-D	0.73 [0.64, 0.82]	13.2 [9.2, 17.2]	52.7 [44.9, 60.5]	12.5 [8.0, 17.0]
CAL-D&Search	0.63 [0.53, 0.73]	12.3 [9.2, 15.3]	44.1 [37.0, 51.3]	9.9 [7.3, 12.6]

Table 6.2: Usage of viewing full document in CAL model for CAL-D and CAL-D&Search.

6.1.3 Usage of Search

Our results show that CAL without search performs as well as or better than a CAL system augmented with interactive search and judging. Because 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 interactive search than via CAL. In this section, we analyze search usage on the CAL-P&Search and CAL-D&Search treatments. We try to understand the frequency of using interactive search when search is available, and how the use of search slows down the overall assessment rate.

Treatment	Fraction of judgments by using search	Fraction of time on search	Avg. time per judgment from search	Avg. time per judgment from CAL
CAL-P&Search	0.29 [0.21, 0.37]	0.40 [0.31, 0.49]	60.2 [49.9, 70.5]	24.2 [20.6, 27.8]
CAL-D&Search	0.39 [0.30, 0.48]	0.44 [0.34, 0.53]	64.7 [52.2, 77.2]	47.1 [37.1, 57.1]

Table 6.3: Frequency of using search in CAL-P&Search and CAL-D&Search. Comparison of assessment time per document between using search and using CAL.

Under the treatments CAL-P&Search and CAL-D&Search, a search interface was provided to users. By summing up the active time spent 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, as shown in Table 6.3. Accordingly, users made 29% (CAL-P&Search) and 39% (CAL-D&Search) of total judgments using search. We also found that 9 participants in CAL-P&Search and 10 participants in CAL-P&Search did not make any judgments using search.

We also measured separately the average time of making a judgment using the search interface and of using the CAL interface. The average time for making an assessment using search is 60.2 seconds (CAL-P&Search) and 64.7 seconds (CAL-D&Search). In contrast, the time to make an assessment by judging paragraph-length excerpts from CAL is significantly shorter, only 24.2 seconds for CAL-P&Search. When the option to view full documents is available, the time to make an assessment from CAL is 47.1 seconds (CAL-D&Search), which is still shorter than assessment time using search. From these results, we can infer that each judgment using search takes more assessment time than judgments made from document excerpts or from full documents selected by CAL.

Treatment	# Unique Queries	# Switches	# CAL Sessions	# Search Sessions
CAL-P&Search	5.1 [3.2, 6.9]	4.8 [3.5, 6.0]	3.0 [2.4, 3.6]	2.8 [2.2, 3.5]
CAL-D&Search	4.5 [3.2, 5.8]	3.8 [2.8, 4.8]	2.5 [2.0, 3.0]	2.3 [1.8, 2.9]

Table 6.4: Search interface usage for CAL-P&Search and CAL-D&Search.

As shown in Table 6.4, users on average made 5.1 (CAL-P&Search) and 4.5 (CAL-D&Search) unique queries using the search engine within one hour. They made 4.8 (CAL-P&Search) and 3.8 (CAL-D&Search) switches between the CAL interface and the search interface. We regard a user staying on and using one interface after each switch as one session. Users used the CAL interface and search interface for 3.0, and 2.8 sessions, respectively, in CAL-P&Search, and 2.5 (CAL) and 2.3 (search) sessions, in CAL-D&Search.

For the judgments made using the search interface, we classified two types, according to their sources. The first type of judgment is made directly from the SERP. Users judge the document snippet returned by the search engine and then make a judgment directly on SERP. We found that users on average made 15.2 (CAL-P&Search) and 16.3 (CAL-D&Search) judgments on SERP. As shown in Figure 5.2, users are also allowed to view the full document by clicking the search result on SERP. Therefore, the second type of judgment is the assessment made by clicking on the result from SERP and viewing the entire document. Users made 17.0 (CAL-P&Search) and 19.8 (CAL-D&Search) judgments by viewing the full document returned by SERP.

Figure 6.2 shows the fraction of participants who used search at a given timestamp for treatments CAL-P&Search and CAL-D&Search. At every minute mark, we recorded the

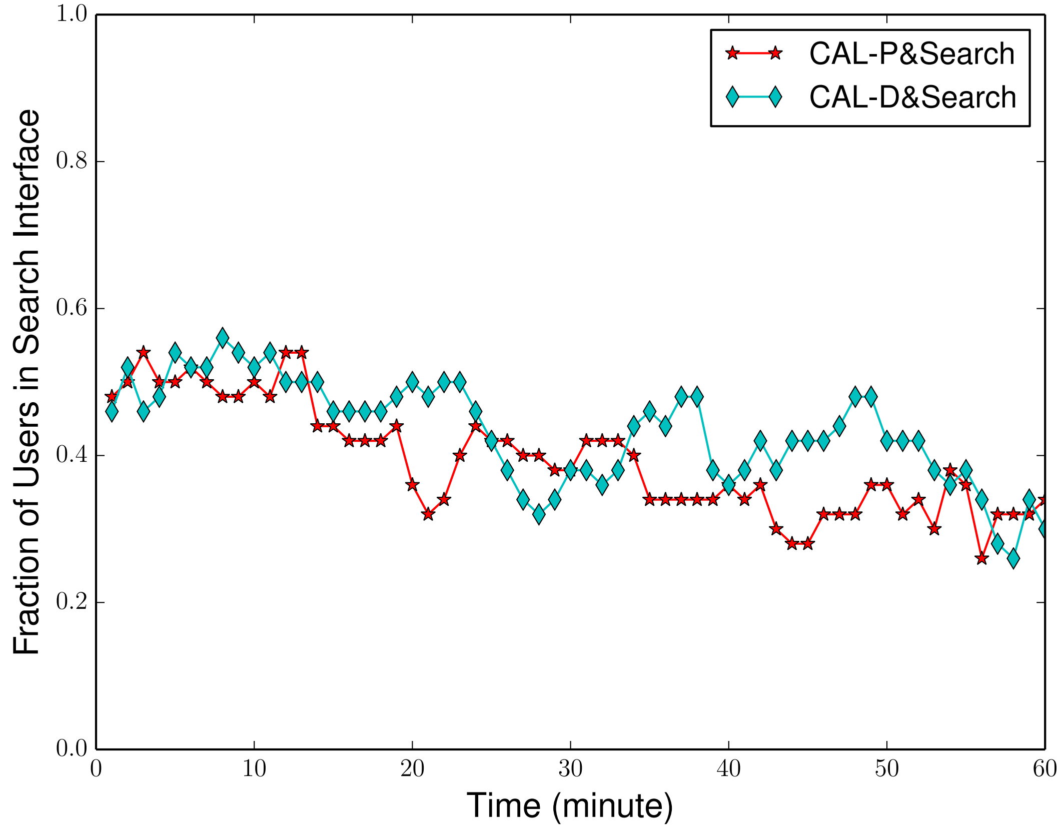

Figure 6.2: The fraction of participants using search during the one hour task.

fraction of participants using search. Then we averaged the fractions over all 50 participants. The results show that users conducted searches during the whole one-hour task. At the beginning of each task, around 50% of participants were using search to make judgments. The fraction of participants using search drops as time increasing. At the end of each task, only around 30% of participants were using search. Participants used the search interface slightly more on CAL-D&Search compared to CAL-P&Search.

6.2 Measuring Judging Performance

In our experiment, we used the assessments from NIST assessors to evaluate the judgments made by users using different treatments. While we know that NIST assessors sometimes make mistakes [Smucker and Jethani, 2011*b*], by comparing different treatments with the same standard, we eliminated the effect of such mistakes. In the first pass, the users found and judged documents using each of our different high-recall system variations. In the second pass, NIST assessors rejudged the documents selected during the first pass and made the final decisions on the labels. We found that some differences exist between NIST assessors' judgments and users' judgments. Users and NIST assessors label documents differently. Judgment performance using different treatments can be compared with the NIST judgments. In this section, we measured the judgment performance of each treatment based on the NIST judgments. We tried to understand which factors in our experiment affected judgment performance.

Participant	NIST judgment	
	Relevant (Pos.)	Non-Relevant (Neg.)
Relevant	TP = True Pos.	FP = False Pos.
Non-Relevant	FN = False Neg.	TN = True Neg.

Table 6.5: Confusion matrix based on judgments from users and NIST assessors.

The task of relevance assessment can be viewed as a subtask of signal detection yes/no theory. The true positive rate (hit rate) shown in Equation 2.8 and false positive rate (false-alarm rate) shown in Equation 2.8 are two well-established methods of measuring judgment performance. In our case, true positive rate or TPR measures the proportion of NIST assessors labelled relevant (positives) that are correctly identified by participants. TPR has the same value as recall, which is equal to $|TP|/(|TP|+|FN|)$, where $|TP|+|FN| = |Pos|$. The false positive rate or FPR measures the proportion of NIST assessors labelled non-relevant (negatives) that are wrongly labelled by participants as positive. FPR is equal to $|FP|/(|FP| + |TN|)$, where $|FP| + |TN| = |Neg|$. TP, TN, FN, and FP are defined in Table 6.5.

Based on TPR and FPR, we can model the user's ability to discriminate between non-relevant and relevant documents. We use the measure d' to characterize the assessor's ability to discriminate [Smucker and Jethani, 2011*a*]:

$$d' = z(TPR) - z(FPR) \tag{6.1}$$

Search Available	CAL types		Marginal means (search)	
	Full doc available	Paragraph only		
No	0.43	0.39	0.41	$p = 0.097$
Yes	0.37	0.36	0.37	
Marginal means (CAL types)	0.40	0.38	Overall Mean	
	$p = 0.454$		0.39	

(a) False Positive Rate (FPR).

Search Available	CAL types		Marginal means (search)	
	Full doc available	Paragraph only		
No	1.10	0.94	1.02	$p = 0.122$
Yes	1.29	1.07	1.18	
Marginal means (CAL types)	1.19	1.01	Overall Mean	
	$p = 0.055$		1.10	

(b) User's ability to discriminate (d').

Search Available	CAL types		Marginal means (search)	
	Full doc available	Paragraph only		
No	-0.29	-0.06	-0.18	$p = 0.669$
Yes	-0.25	-0.11	-0.18	
Marginal means (CAL types)	-0.27^{*}	-0.08^{*}	Overall Mean	
	$p = 0.004$		-0.18	

(c) User's criterion c.

Table 6.6: We have marked with $*$ the differences that are significant at $p < 0.05$.

where z is the inverse of the cumulative distribution function of the normal distribution which converts the TPR or FPR to a score. A higher d' value indicates a greater ability to discriminate. Users with different TPR and FPR might have the same d'. For example, User A with a TPR of 0.73 and a FPR of 0.35 has the same $d' = 1$ as User B with a TPR of 0.89 and a FPR of 0.59 [Smucker and Jethani, 2011*a*].

We also computed user's criterion c, as shown in Equation 6.2. For this calculation, the user selects a criterion for relevance judgment. When the user's confidence regarding relevance is above the criterion, the document is labelled relevant; otherwise, it is labelled non-relevant.

$$c = -\frac{1}{2}(z(TPR) + z(FPR)) \tag{6.2}$$

where z is the same score function used in Equation 6.1. A negative criterion means that users are more likely to make false positives in order to avoid missing relevant documents. In other words, users with negative c are more liberal in making judgments. A positive criterion signifies that users are conservative in making assessments in an attempt to avoid false positives.

For the computation of d' and c, rates of 0 or 1 for TPR or FPR will lead to infinities. The values of 0 or 1 will be very probable when these rates are estimated based on very small samples. We follow a standard correction used by Smucker and Jethani [2011*a*] to avoid infinities. Estimated TPR (eTPR) and estimated FPR (eFPR) are defined in Equation 6.3 and Equation 6.4, respectively. For computing d' and criterion c, we used eTPR and eFPR instead of TPR and FPR.

$$eTPR = \frac{|TP| + 0.5}{|TP| + |FN| + 1} \tag{6.3}$$

$$eFPR = \frac{|FP| + 0.5}{|FP| + |TN| + 1} \tag{6.4}$$

TPR has the same value as recall. The results of recall or TPR are shown in Table 5.3c. We calculated the FPR of the judgments produced using each treatment. As shown in Table 6.6a, CAL-D has the highest FPR and CAL-P&Search has the lowest. CAL with paragraphs has lower FPR compared with CAL with full documents, but not at a statistically significant level. Likewise, CAL with the ability to search helps reduce FPR but not at a significant level.

The results of d' yielded by different treatments are shown in Table 6.6b. Of the four treatments, CAL-D&Search has the highest d' (1.29) and CAL-P has the lowest d' (0.94). The ability to search and the option to view full documents both helped improve the ability to discriminate d' but not at a statistically significant level.

The results of criterion are shown in Table 6.6c. We found that all four treatments yielded negative criterion values. CAL-D has the lowest c and CAL-P has the highest c . The ability to view full documents significantly reduced the criterion of users. CAL with full documents has c of -0.27 while CAL with document excerpts has c of -0.08. Users are more liberal in making assessments when they are given the ability to view full documents. The option of using search does not make a difference on criterion.

6.3 System Recall

Search Available	CAL types		Marginal means (search)	
	Full doc available	Paragraph only		
No	33.1	58.7	45.9*	$p = 0.009$
Yes	35.1	45.7	40.3	
Marginal means (CAL types)	34.1	52.2*	Overall Mean	
	$p < 0.001$		43.2	

(a) System retrieved NIST relevant documents.

Search Available	CAL types		Marginal means (search)	
	Full doc available	Paragraph only		
No	0.26	0.42	0.34	$p = 0.009$
Yes	0.30	0.37	0.34	
Marginal means (CAL types)	0.28	0.39*	Overall Mean	
	$p < 0.001$		0.34	

(b) System recall $recall_{sys}$.

Table 6.7: Performance achieved from system side.

We observed that the factors of viewing full documents and using search yielded effects on user performance of judgment in aspects such as relevance criterion and ability to discriminate. User criterion and ability to discriminate further affect the number of relevant documents found and the recall achieved. Using Equation 5.3, we measured the recall of each system variation by computing the number of user-found NIST relevant documents. We found that some relevant documents retrieved by the system were judged as non-relevant by users. These false negatives do not contribute to recall.

We used the system recall $recall_{sys}$ defined in Equation 6.5 to measure the performance of each system variation by minimizing the assessing effect of users. We tried to compute and compare which system variation retrieves and presents the greatest number of NIST relevant documents to users. Although the user judgment plays an important role in the relevance feedback process, we calculated system recall in order to minimize effects of user judgment on the evaluation of system effectiveness.

$$recall_{sys} = \frac{|S_{ret} \cap R|}{|R|} = \frac{|TP| + |FN|}{|R|} \tag{6.5}$$

where S_{ret} is the set of documents retrieved by each system variation and presented to users and R is the set of NIST-defined relevant documents. In the CAL interface, the documents in S_{ret} are returned by the CAL model. In the search interface, the documents in S_{ret} are returned by the search engine and judged by assessors. If a document is returned by our search engine but not assessed by a user, it is not included in S_{ret}. $S_{ret} \cap R$ is the set of system-retrieved NIST relevant documents.

The results for system-retrieved NIST relevant documents and system recall are shown in Table 6.7a and Table 6.7b, respectively. Of the four system variations, CAL-P is able to find a greater number of system-retrieved NIST relevant documents and achieve higher recall. Both the ability of viewing full documents and the ability to search significantly reduced the number of system-retrieved NIST relevant documents found. CAL with paragraphs achieved significantly higher system recall than CAL with full documents. However, search neither helped nor hurts system recall.

6.4 Analysis of User Feedback and Preference

In our controlled study, participants were required to answer a questionnaire after each task. We asked about their overall experience using the different system variations and their preference regarding different system features.

In this section, we compare user feedback regarding the different variants of our system. At the end of the study, we asked participants which system variant they preferred. 48% of study participants preferred CAL-D&Search over the more restrictive variants. In other words, our participants wanted full control of a highly interactive system. This preference runs counter to the finding that their performance was highest when their interactions were limited to producing relevance judgments on paragraph-length excerpts.

We asked participants for their feedback on each of the system features in Table 6.8. We used a 5-point scale to rate each feature. The results are shown in Figure 6.3. The keyword highlighting feature is the most popular, with 86% of users indicating that it was somewhat useful or very useful. Users preferred to use most system features.

Table 6.8: Features of the HiCAL system rated by participants.

Feature	Description	Example
Keyword Highlighting	Keyword search within a document or paragraph	Figure 5.1f, 5.2h
Judgment Shortcuts	Keyboard shortcuts for submitting relevance judgments	Figure 5.1h, 5.2i
Search Interface	Ability to use a search engine to find documents in addition to the learning interface	Figure 5.2
Topic Description	Display of topic statement of what is considered relevant	Figure 5.1a, 5.2a
Undo Judgments	Ability to review recent judgments and change judgment	Figure 5.1i
Full Document Content	Ability to view a full document rather than merely a paragraph summary	Figure 5.1e
Advance Search	For the search engine, the ability to specify phrases ("new york") or require words (+france)	Figure 5.2c

Table 6.9: Percentage of participants preferring a given system variant.

Treatments	Percentage of participants
CAL-P	16%
CAL-D	26%
CAL-P&Search	10%
CAL-D&Search	48%

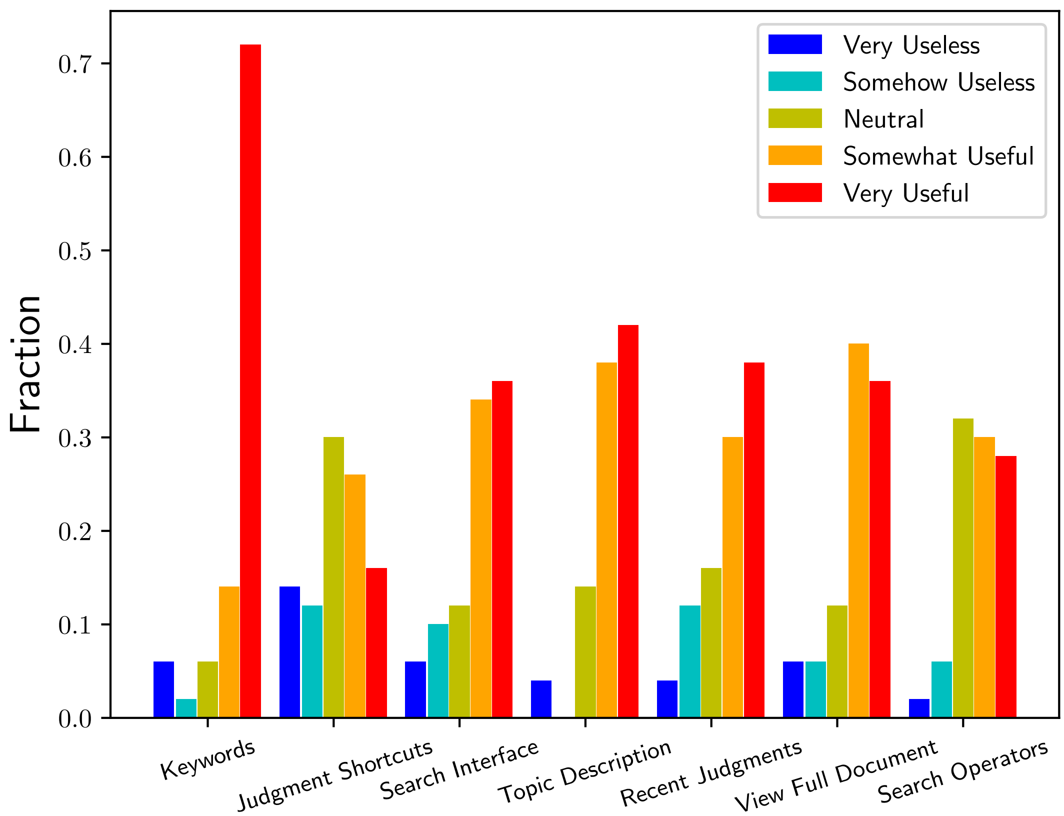

Figure 6.3: Percentage of user preference for different system features.

Chapter 7

Volume Estimation Using Sampling Strategy

When to stop assessment is also a crucial problem for high-recall retrieval task. Given a large dataset and a limited assessment budget, assessing every single document to achieve high recall is not practical and realistic. In the CAL process, we find that it becomes even harder to find relevant documents as the number of CAL iterations increase. In many cases, knowing the number of relevant documents in the dataset helps assessors to determine when enough relevant documents have been found. And stopping further assessments helps reduce excessive unnecessary judgments. One way to address this problem is to estimate the volume of relevant documents in the dataset. We refer to this issue as the volume estimation problem.

This chapter describes a method to accurately and efficiently estimate the number of relevant documents in a collection for a given topic. How would we do this both *accurately*, such that our estimate is as close as possible to the actual number, and *efficiently*, with as little assessment effort as possible? The existing active learning techniques for high-recall retrieval provide a baseline. We can count the number of assessments to find all the relevant documents via continuous active learning methods. Alternatively, we could just randomly sample from the collection to estimate the prevalence and infer the volume. The question is: can we do better than either approach?

The contribution of this paper is the development and evaluation of a technique for volume estimation based on continuous active learning and sampling. The intuition of our method contains two main steps. The first step is to use active learning methods to find all the "easy-to-find" relevant documents. And then the second step is to use sampling methods to estimated the number of relevant documents in the remaining collection. The switch point between the first step and the second step is called as "switchover" point or knee point, which can be understood as the "knee" in an effort vs. recall gain curve. The idea is to take advantage of active learning to find all the easy-to-find relevant documents until the knee point, and then use sampling techniques to estimate the remainder of the collection.

We use a simple and effective technique for detecting this knee point and explore three different sampling methods past the knee. We conduct experiments on several TREC provided datasets and a collection of tweets. The results show that our best strategy yields more accurate estimation (with the same assessment effort) than several alternatives.

7.1 Volume Estimation Approaches

Suppose we would like to estimate the number of relevant documents for a given topic in a particular data collection consisting of D documents.

A simple and naïve approach might be to randomly sample (without replacement) documents from the collection and assess them for relevance. We can approximate this as a Bernoulli process, for which the volume estimate R_T is:

$$R_T = (R_E/E) \cdot D \tag{7.1}$$

where R_E is the number of relevant documents after examining a total of E documents. This algorithm is not complete since that we still need to know how many samples E to draw before sampling. We discuss how to tackle this issue in our paired experimental methodology Section 7.2.

An alternative simple way is to use an existing active learning based high-recall retrieval method such as the BMI (Section 3.4). By this way, we simply *find* all the relevant documents, which is a trivial way to determine the volume. However, the problem of using continuous active learning is that the BMI does not provide an established stopping criterion.

By relying on the effectiveness of BMI, we can first apply BMI to find and judge A documents. As shown in the results of TREC Total Recall track, the BMI based methods were able to find relevant documents with high precision in the beginning stage. During this human-in-the-loop relevance feedback process, we first explore some fraction of the collection that has a higher rate of relevant documents. And we discover R_A documents among A judgments. This value R_A provides a lower bound on the number of relevant documents in the collection. We can ensure that this data collection contains at least R_A relevant documents. However, the problem is that we don't know how many relevant documents there are in the residual collection which has not been reviewed. Even so, we can still estimate an upper bound using the rate at which we're finding relevant documents in the BMI process, and extrapolate to the remainder of the collection. However, this makes a inaccurate estimation since that active learning method usually rank documents more likely to be relevant before documents less likely to be relevant. Therefore, the rate of relevant documents in the active learning process is usually much higher than the rate in the residual collection. Therefore, such an inference would yield an unrealistically large overestimate.

One solution to extrapolate the residual collection is sampling. This requires to further answer two questions: First, how to determine the knee point (i.e., the value of A)? Second, how to sample the residual collection and estimate the prevalence after the knee point? We tackle these two questions in turn.

Let us suppose that we sample S documents to be judged: we can estimate the prevalence of relevant documents and then infer the total number of relevant documents left in the residual collection R_S. We can estimate the total number of relevant documents by adding this value to R_A, the number of relevant documents we found during the active learning process.

The contribution of this approach is the development of this two-phase volume estimation technique that integrate sampling with active learning method. Below, we describe simple techniques for setting A, the amount of assessment effort to expend in active learning process, and S, the amount of effort to expend in random sampling.

7.1.1 Find the Knee

In our approach, we employ the BMI described in Section 3.4 augmented by the following the knee-finding method proposed by Cormack and Grossman [2016*a*]. In each iteration, the BMI selects a batch of documents for human assessments. The batch size is exponentially increasing as the number of iterations growing. After receiving relevance feedback for each batch of documents, we can measure the gain curve based on the number of relevant documents found. In this case, the y axis of the gain curve is the number of relevant documents found.

At the end of each iteration of BMI, we have two numbers that correspond to the total number of relevant documents found and the total number of documents judged so far.

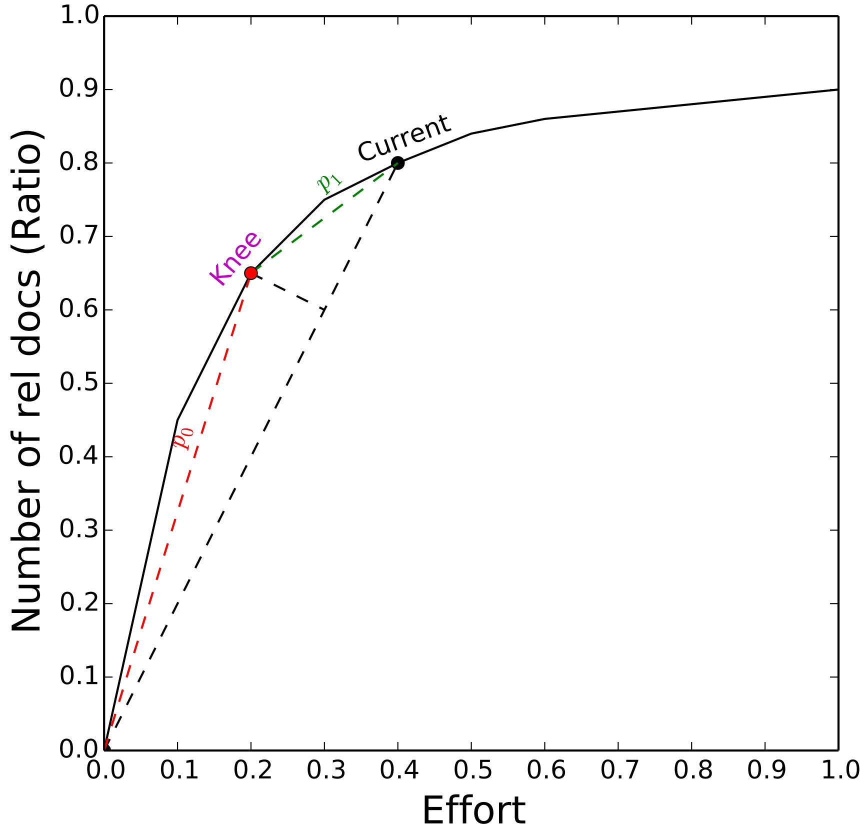

Figure 7.1: Detection of the knee point in the gain curve.

Figure 7.1 shows how we detect the knee point. A candidate knee point is selected as follows: find a point on the gain curve with maximum perpendicular distance from a line between the origin point $(x = 0, y = 0)$ and the current point of the curve. Let p_0 be the slope of the line from the origin point to the candidate knee point, p_1 be the slope of the line from the candidate knee point to the current point, and the slope ratio $\rho = \frac{p_0}{p_1}$. The candidate knee point can be determined if these two criteria are satisfied:

1. the number of documents examined so far from the active learning process exceeds 1000 (to ensure that the active learning process has surpassed the beginning "ramp up" stage);

2. $\rho > 6$, if at least 150 relevant documents have been retrieved; or $\rho > 156 - r$, if $r < 150$ relevant documents have been retrieved.

The second clause in the second criterion is a special case for handling topics with low prevalence of relevant documents. The parameters for this technique were tuned on a private dataset [Cormack and Grossman, 2016*a*].

Note that to be precise, we don't actually discover the knee point until we've passed the actual knee point (the current point is always beyond the knee point), but the intuition nevertheless holds. The actual switchover point where we stop active learning process and start sampling corresponds to the point where we discovered the knee. However, we still use "stopping at the knee" to explain our method.

7.1.2 Sampling Strategies

Based on the knee-finding algorithm described in the previous section, we apply the BMI until the knee point has been found. As noted, we actually stop at the "current" point shown in Figure 7.1. At that point, we have judged A documents and found R_A relevant documents. The next question is: how to extrapolate the residual collection that we have not yet explored? In our experiment, three sampling strategies are presented:

Negative Binomial Sampling. In this approach, document are sampled from the residual collection until M relevant documents have been found. This sampling process requires us to judge S documents. Each sample can be characterized as a Bernoulli trial, and thus the sampling process can be modeled by a negative binomial distribution (or Pascal distribution). Each Bernoulli trial has two potential outcomes that are success and failure. In our experiment, success represents that the sampled document is judged relevant and failure means that sampled document is judged non-relevant. Under this interpretation, the minimum variance unbiased estimator for $\hat{p}$, the probability of success (i.e., probability of a document being relevant) is given as:

$$\hat{p} = \frac{r-1}{r+k-1} = \frac{M-1}{S-1} \tag{7.2}$$

where r is the number of relevant documents found (which we set to M), and k is the number of non-relevant documents in our sequence of observations [Johnson et al., 2006]. Note that the total number of judgments in sampling $S = r + k$.

From Equation 7.2, our estimate of the total number of relevant documents, R_T, is as follows:

$$R_T = R_A + (D - A)\frac{(M-1)}{(S-1)}, \text{ for } M > 1. \tag{7.3}$$

In our experiments, we tried varying $M \in \{2, 4, 8\}$. Higher values of M reduce the variance, but at the same time cost more assessment effort. The total effort is $A+S$ for this approach, where S is the total number of judged documents to find M relevant documents.

The Horvitz-Thompson Estimator. In some cases, some topics have only few relevant documents. It might require a lot of assessment effort to find M relevant documents in the residual collection where the prevalence of relevant documents is low. Therefore, using negative binomial sampling for estimating those low-prevalence topics could be expensive. An alternative method is to use the classifier trained in BMI to score all documents in the residual collection, thus ranking all remaining documents in the descending order of relevance. The classifier in BMI is derived from the current point, and thus it is trained upon all the existing judgments. The more-likely-to-be-relevant documents are ranked prior to the less-likely-to-be-relevant documents.

We apply a standard estimation method called the Horvitz-Thompson Estimator (HT estimator) [Tillé, 2006] to estimate the total superpopulation in a stratified sample. First, we compute a distribution over all documents in the residual collection. The probability of each document being sampled is proportional to its probability of relevance (as estimated by the classifier in BMI). This renormalized distribution is referred as the inclusion probability. Defined in Equation 7.4, π_i refers to the probability that document i will be sampled. In our experiment, we use the relevance score $score_i$ of document i generated from the logistic regression classifier which is also used in BMI. We normalize the relevance score to $[0, 1]$ via $\pi_i = 1/(1 + exp(-score_i))$. The Horvitz-Thompson Estimator estimates the total number of relevant documents R_T:

$$R_T = R_A + \sum_{i=1}^{n} \pi_i^{-1} Y_i \tag{7.4}$$

where Y_i is an indicator variable for relevance for each of the n sampled documents. The value of Y_i is 1 (relevant) or 0 (non-relevant) for a given document i. R_A is the number of relevant document found in the active learning phase. Note that this method does not form a complete algorithm because the stopping criterion is still missing. The HT estimator does not tell us how many samples n we need to draw. Here again, we address this issue in our paired experimental methodology in the below section.

Stratified Sampling. To address the downside of the HT estimator where a well established stopping criterion is needed, a novel stratified sampling strategy is used. This stratified sampling method also ranks the residual collection based on the relevance score after when the knee point is detected. This sampling approach proceeds in iterations: in the i-th iteration, we randomly sample $K^S = 1000$ documents from the next top ranking $K = 10,000$ documents and let assessors judge those documents. Suppose we find R_i relevant documents, and thus we can then estimate the proportion of relevant documents for i-th iteration is (R_i/K^S). We can further infer that there are $K \cdot (R_i/K^S)$ in the top K ranked documents. We then proceed to the next iteration and sample another K^S documents from the *next* K top ranked documents. The above process repeats until we cannot find any more relevant document from K^S sampled documents. Then, the total number of relevant documents can be estimated as:

$$R_T = R_A + \sum_{i=1}^{n} K \cdot (R_i/K^S) \tag{7.5}$$

where n is the total number of iterations. R_A is the number of relevant documents found in the active learning process. The total effort expended is $A + n \cdot K^S$, where A is the number of assessments from the active learning part.

We name this strategy as stratified sampling because we select samples from each "strata" of K documents, and using the estimated prevalence to infer the number of relevant documents for each strata. We keep repeating this process until we find no more relevant documents from the selected samples, which allows us to have a stopping criteria and exit early for low prevalence topics to avoid excessive assessments.

7.2 Experimental Setup

To evaluate and compare our different volume estimation strategies, we used public test collections from the TREC 2015 Total Recall Track [Roegiest, Cormack, Grossman and Clarke, 2015]. More specifically, we used three collections: the (redacted) Jeb Bush Emails (called "Athome1"), consisting of 290k emails from Jeb Bush's eight-year tenure as the governor of Florida (10 topics); the *Illicit Goods* dataset (called "Athome2") collected for the TREC 2015 Dynamic Domain Track, consisting of 465k documents from a web crawl (10 topics); and the *Local Politics* dataset (called "Athome3") collected for the TREC 2015 Dynamic Domain Track, consisting of 902k documents from various news sources (10 topics). For each topic, a complete set of relevance assessment is provided by Total Recall Track coordinators. The relevance assessment on documents of these test collections are derived using continuous active learning method described in the track overview paper [Roegiest, Cormack, Grossman and Clarke, 2015]. The details of these test collection are listed in Chapter 4. In our experiment, some strategies require assessors to judge the relevance of sampled documents. We simulate the relevance judgment process. The relevance of a given document is determined by the set of relevance labels.

For the purposes of our study, the provided test collection and corresponding evaluation methodology has been sufficiently validated for assessing the effectiveness of high-recall tasks in Total Recall Track 2015. Therefore, these test collections are suitable for our volume estimation problem. Finally, as a validation set, we evaluated our techniques on the Twitter collection described by Bommannavar et al. [2016], who *exhaustively* annotated nearly 800k tweets from one day in August 2012 with respect to four topics: Apple (the technology company), Mars (the planet), Obama, and the Olympics. This test collection exactly satisfied our goal: how much relevant information is there on social media about a particular topic?

As discussed, the random sampling and the HT estimator approaches are not complete estimation algorithms since they both lack a stopping criterion. In other words, they do not tell us how many samples to draw. In contrast, the advantage of negative binomial sampling and stratified sampling is that each of them have a stopping criterion, where the total effort can be determined. Therefore, the random sampling and the HT estimate are compared to negative binomial sampling (N.B.) and stratified sampling in a **paired setup**. where we evaluate and compare these techniques at the same level of effort (number of assessments). This models an A/B testing scenario in which we have two parallel assessors judge the sampled documents from two different techniques at exactly the same pace. When one technique terminates, we also stop the other one. At that stop point, we ask the questions that how do the two estimates compare and which estimate is closer to the actual number of relevant documents?

Our experimental procedure is as follows: for each topic in a collection, we ran our estimation technique (either negative binomial sampling or stratified sampling) and computed the total number of assessments. We then ran a paired experiment with either random sampling or the HT estimator (or both) using exactly the same level of effort. We recorded the estimated volume for all techniques. For each collection, we report the average (relative) error across all topics and the root mean square error between the estimated volume and the actual number of relevant documents. The Athome1 collection was used as our training set, on which we ran 50 trials of the above procedure to characterize the variation of estimates. The Athome2, Athome3, and Twitter collections were used as held-out test sets—we report the results of a single trial.

7.3 Results

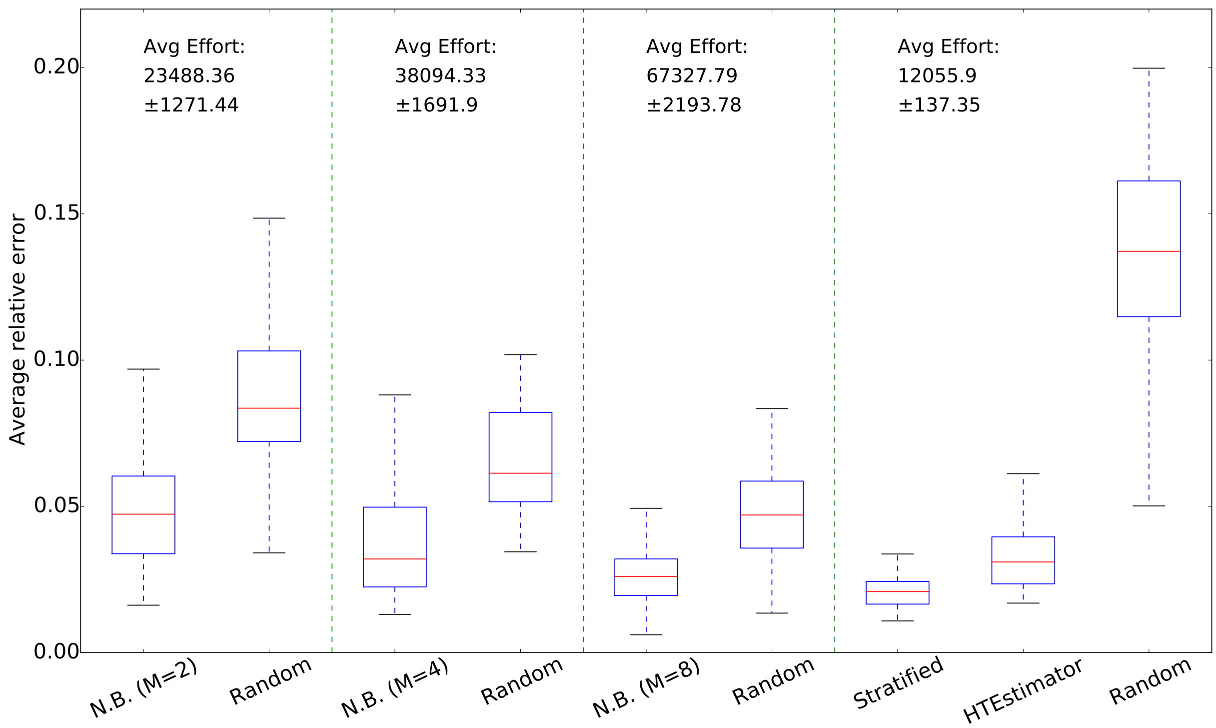

Figure 7.2: Box-and-whiskers plot characterizing 50 trials of each of our techniques on the Athome1 collection.

The results of 50 trials of our experimental procedure on Athome1 test collection are shown in Figure 7.2. The average relative error and overall distribution across the 50 trials str characterized by a standard box-and-whiskers plot. We compared negative binomial sampling (N.B.), $M = \{2, 4, 8\}$, with random sampling using the paired approach described above. For each comparison, we first evaluate negative binomial sampling and compute the total number of effort to accomplish the sampling process. Then, the random sampling approach is severed as a baseline method and compared with negative binomial sampling at the same level of effort. d We also compared stratified sampling with the HT estimator and random sampling using exactly the same procedure. Each of these comparisons is shown by grouped bars (separated by dashed lines) in the Figure 7.2.

As expected, the negative binomial sampling approach becomes more accurate and robust with increasing values of M (but meanwhile requires correspondingly more effort).

For reference, the entire Athome1 collection contains 290k documents. When using negative binomial sampling with $M = 8$, we on average need to examine nearly a quarter of the collection. However, we find that negative binomial sampling is more accurate than random sampling at the same level of effort with respect to all different M values.

We also compare the stratified sampling, HT estimator, and random sampling using the same level of effort. It is obvious that the stratified sampling approach is superior to all other techniques. On average, stratified sampling requires nearly half as much effort as negative binomial sampling with $M = 2$ but yields much more accurate estimates. In fact, stratified sampling provides more accurate estimates than negative binomial sampling with $M = 8$, at about one fifth of the effort. Stratified sampling also performs better than both the HT estimator and random sampling at the same level of effort.

Results on Athome2 and Athome3 test collections are shown in Table 7.1. Since these comprise our held-out test data, we only report the results of a single trial instead of 50 trials. In the table, rows are grouped together in terms of different techniques at the same level of effort. For example, the rows marked "= sample" denote accuracy of random sampling with the same number of judged documents as the corresponding negative binomial or stratified condition. There exists sampling variability for our different strategies. In our single trial, we observe greater error with $M = 8$ than with $M = 4$ using negative binomial sampling. This observation is not inconsistent with the results in Figure 7.2. The upper bound of negative binomial sampling results has higher relative error than the lower bound of random sampling results.

Overall, the results on Athome2 (465k documents) are consistent with the results from Athome1, our training set. Negative binomial sampling becomes more accurate and robust with increasing M and is more accurate than random sampling with the same level of effort. In contrast, our stratified sampling technique provides comparable low error rate but at far less effort, beating both the HT estimator and random sampling methods.

In terms of effort relative to the size of the collection, it appears that the Athome2 topics are a bit more "difficult" compared to Athome1. However, our stratified sampling approach actually requires *less* effort ($8,363$ judged documents) on a larger collection—Athome2 compared to Athome1 which takes on average $12,055.9$ judgments.

Results of negative binomial sampling method on the Athome3 collection, which contains 902k documents, are quite poor. Table 7.2 shows the reason: for each topic in that collection, we list the total number of relevant documents, the effort used in the active learning part of our procedure, and the number of relevant documents found at that point. For five of the topics (those in bold), active learning helped find either all or nearly all the relevant documents, which means that our termination criterion for negative binomial

Measure	Avg Effort	Avg Relative Error	Root Mean Square Error
Athome2			
Neg. Binomial ($M = 2$)	80925	0.016	0.021
= sample	80925	0.094	0.122
Neg. Binomial ($M = 4$)	122527	0.014	0.023
= sample	122527	0.052	0.062
Neg. Binomial ($M = 8$)	181407	0.015	0.020
= sample	181407	0.045	0.060
Stratified	8363	0.026	0.042
= HTEstimator	8363	0.051	0.070
= sample	8363	0.410	0.621
Athome3			
Neg. Binomial ($M = 2$)	482237	0.041	0.105
= sample	482237	0.045	0.079
Neg. Binomial ($M = 4$)	546379	0.011	0.030
= sample	546379	0.042	0.073
Neg. Binomial ($M = 8$)	597489	0.023	0.058
= sample	597489	0.032	0.064
Stratified	3168	0.053	0.113
= HTEstimator	3168	0.100	0.200
= sample	3168	0.867	1.119
Twitter			
Neg. Binomial ($M = 2$)	24160	0.261	0.233
= sample	24160	0.222	0.240
Neg. Binomial ($M = 4$)	39162	0.106	0.090
= sample	39162	0.046	0.041
Neg. Binomial ($M = 8$)	40295	0.007	0.036
= sample	40295	0.179	0.181
Stratified	22687	0.047	0.048
= HTEstimator	22687	0.093	0.092
= sample	22687	0.170	0.218

Table 7.1: Results of various volume estimation techniques on the Athome2, Athome3, and Twitter collections.

Topic	Rel Docs	Knee Stop Effort	RelAtKnee
athome3089	**255**	1105	**254**
athome3133	**113**	1105	**112**
athome3226	2094	3478	2022
athome3290	**26**	2316	**26**
athome3357	629	1526	599
athome3378	**66**	1105	**66**
athome3423	76	1232	40
athome3431	1111	1232	1106
athome3481	2036	3478	1924
athome3484	**23**	1105	**23**

Table 7.2: Relevant documents identified and effort when BMI terminates for Athome3.

sampling (e.g., with $M = 2, 4, 8$) is never met. And hence, the sampling process forces us to examine *the entire collection* and try to find relevant document that does not exist. In contrast, with stratified sampling we examine 1000 of the top 10,000 ranked documents, find zero relevant, and terminate. The stratified sampling only takes 3,168 judgments to stop and achieves comparably low error rate.

Of course it would be reasonable to add in a termination condition in the negative binomial sampling case (e.g., stop after K documents if we haven't found a single relevant one). But then we're just duplicating the initial iteration of the stratified sampling approach. We leave this idea for future work.

The bottom group of Table 7.1 shows our results on the Twitter collection. Once again, we report results from a single trial. Overall, the findings are consistent with the other collections: our stratified sampling technique clearly yields more accurate estimates than all other techniques while requiring less effort. This gives us some degree of confidence that our algorithms, developed on email (Athome1), can be generalized to entirely different collections (e.g., tweets). We have no explanation as to why negative binomial sampling with $M = 4$ gives worse estimates than comparable random sampling, or why comparable random sampling with $M = 8$ gives such poor results. We purposely decided against error analysis in order to preserve the sanctity of this validation set.

7.4 Conclusion

Estimating the number of relevant documents in a collection is critical to the high-recall retrieval problem. Our results reflect that actually *finding* the relevant documents is a good approach to *counting* and then estimating the total volume. However, our best strategy reflect that we should first identify the "easy to find" documents and then extrapolate on the rest collection via sampling. Among different sampling strategies, stratified sampling yields more accurate and robust results with the same level of effort. Our approach establishes a baseline for future work on an important real-world application.

Chapter 8

Conclusion and Future Work

In this thesis, we have investigated judging short document excerpts for relevance feedback in continuous active learning to achieve high recall while reducing assessment effort. We designed and implemented a high-recall retrieval system, called HiCAL. We evaluated the effectiveness of the HiCAL system by separately conducting a simulation study and a controlled user study. We also explored factors that could affect the effectiveness of achieving high recall, such as giving users the ability to search and view full documents. We also investigated the problem of when to stop assessment during the continuous active learning process to avoid excessive assessment.

8.1 Summary

We first approached the high-recall retrieval task by participating in the Total Recall Track 2015 [Roegiest, Cormack, Grossman and Clarke, 2015]. The Total Recall Track organizers provided participants with a version of AutoTAR called the BMI, and used the BMI as the baseline method. We augmented the BMI implementation with seed document selection, feature engineering, and query expansion. On the basis of this modified BMI, we submitted corresponding runs to the Total Recall Track for evaluation. The results indicated that no submitted runs were able to consistently beat the BMI over different datasets. The superiority of BMI in high-recall tasks was further validated on Total Recall Track 2016 [Grossman et al., 2016] and CLEF 2017 eHealth lab task [Anagnostou et al., 2017].

We also investigated using document excerpts as relevance feedback in CAL to achieve high recall. In our simulation study, we integrated sentence-level relevance feedback into

BMI. For each document selected by BMI, the simulated assessor reviewed a single sentence instead of the full document to make the assessment. In addition, we compared sentence-level feedback strategies with document-level feedback strategies on three different dimensions. The three dimensions are (1) let the assessor assess a single sentence or the full document to label the document; (2) select the best sentence or the best document from the highest-scoring document or the highest-scoring sentence, respectively; (3) retrain the classifier by adding the labelled document or the labelled sentence into the training set. For each dimension, there exists a binary choice, and thus, in total, eight combinations. The results of this simulation experiment show that the sentence-level relevance feedback method achieves almost the same level of recall as the document-level relevance feedback method based on the same number of judgments (the number of documents reviewed).

To validate the hypothesis that reviewing document excerpts for relevance feedback can reduce assessment time and effort for achieving high recall, we conducted a 50-person controlled user study. We designed and implemented a high-recall retrieval system (HiCAL), which applies continuous active learning (CAL). 50 users used different variants of the HiCAL system to find as many relevant documents as possible within one hour. The HiCAL system could display either full documents or short document excerpts for relevance assessment. In addition, the HiCAL system integrated a search engine with CAL, so that users could use interactive search and judging to find relevant documents. The results from the user study reveal that users can find a larger number of relevant documents by viewing document excerpts within one hour. The option to view full documents and to use interactive search slowed down users' assessments and resulted in a lower number of relevant documents.

Finally, we investigated how to estimate total volume of relevant documents for a specific topic. We combined CAL with a sampling method to perform estimation. First, CAL was used to find as many relevant documents as possible until the knee point at which the majority of relevant documents had been found. Then, sampling methods were used to sample documents and estimate the proportion of relevant documents in the residual collection. We compared several sampling methods and found that the stratified sampling method yields the most accurate and robust estimation at any level of effort.

8.2 Future Work

8.2.1 User Study for Higher Recall within Longer-Time Span

For Chapter 5, we conducted a controlled user study, in which 50 users judged as many relevant documents as possible within one hour using our HiCAL system. The best performed strategy —CAL-P—achieved on average 0.27 recall over 50 topics within one hour. Apparently, it is difficult to achieve very high recall within just one hour of work. Therefore, we could let participants work longer and judge more documents for each task, thus potentially achieve higher recall. Then, we could compare different methods by measuring the amount of time required to achieve a certain level of high recall.

The reason for not being able to achieve high recall could also be due to the assessment disagreement between participants and NIST assessors. We measured recall according to the number of documents labelled relevant by both participants and NIST assessors. If a NIST-labelled relevant document is labelled as non-relevant by our assessor (False Negative), this document is not contributing to recall. Therefore, if there exist False Negatives, high recall cannot be achieved even when all the relevant documents have been retrieved and presented to the assessors. We leave answering this question for future work.

8.2.2 Use and Evaluation of Highly Relevant Documents

In our user study, participants were able to label relevant documents as highly relevant or relevant. We did not distinguish the highly relevant documents from the relevant documents in our experiments. However, highly relevant documents can be used as extra information to help improve the CAL model, since they usually contain stronger signals about relevance than the relevant documents. In each iteration of CAL, we could adjust the weights for highly relevant documents when retraining the classifier. We could also modify the pairwise loss function in logistic regression when comparing highly relevant documents with relevant documents. A pair is correct if the score for a highly relevant document is higher than that of a relevant document. If a relevant document is ranked higher than a highly relevant document, we would penalize this pair.

On the other hand, if the NIST assessors were to provide the graded relevance labels for documents, we could also discriminate highly relevant documents from relevant documents when evaluating the results. In some cases, it might be more tolerable to miss some relevant documents than to omit highly relevant documents. Therefore, we could reward the system which is able to find a larger number of highly relevant documents, and penalize a system that omits many relevant documents. How to assign different weights to highly relevant documents and to relevant documents to evaluate recall remains a problem.

8.2.3 Search vs. CAL in the Long Run

Interactive search and judging can significantly improve precision according to our user study. More specifically, we found that search helped participants more precisely find relevant documents compared to CAL, especially in the first stage of assessment. However, participants spent more time to find relevant documents using search. Therefore, there is a trade-off between precision and time cost when using search to find relevant documents. We already know that CAL usually reaches a plateau when the proportion of relevant documents begins to drop significantly. CAL might get stuck at a particular point and be unable to find any new relevant documents for a long time. At that point, search might be able to help the assessor find relevant documents and explore more relevance space. We assume that suitable switches between search and CAL might help achieve high-recall retrieval.

8.2.4 Effects of Judgments on Ranking Correlation of Runs

In Section 5.3.2, we found that judgments from search reduced the RMSE of MAP scores compared to NIST qrels when scoring the TREC 2017 Common Core runs [Zhang, Abualsaud, Ghelani, Ghosh, Smucker, Cormack and Grossman, 2017; Abualsaud, Cormack, Ghelani, Ghenai, Grossman, Rahbariasl, Smucker and Zhang, 2018]. But judgments without search achieved higher τ compared to the MAP scores from NIST qrels. We hypothesize that search can help participants find some special relevant documents that cannot be found by CAL. These high-value relevant documents might help align the MAP scores and produce closer MAP scores compared to NIST qrels.

In addition, we found that no treatments produce close MAP scores for the Routing runs (automatic runs using existing old qrels). Some relevant documents found by Routing runs might not be found by our system; thus judgments from human assessors cannot rank those runs correctly. These questions demand further analysis.

References

Abualsaud, M., Cormack, G. V., Ghelani, N., Ghenai, A., Grossman, M. R., Rahbariasl, S., Smucker, M. D. and Zhang, H. [2018], UWaterlooMDS at the TREC 2018 Common Core Track, *in* 'TREC'.

Abualsaud, M., Ghelani, N., Zhang, H., Smucker, M. D., Cormack, G. V. and Grossman, M. R. [2018], A system for efficient high-recall retrieval, *in* 'The 41st International ACM SIGIR Conference on Research and Development in Information Retrieval', SIGIR '18, ACM, New York, NY, USA, pp. 1317–1320.

Ai, Q., OConnor, B. and Croft, W. B. [2018], A neural passage model for ad-hoc document retrieval, *in* 'European Conference on Information Retrieval', Springer, pp. 537–543.

Allan, J. [1995], Relevance feedback with too much data, *in* 'Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval', SIGIR, ACM, New York, NY, USA, pp. 337–343.

Allan, J. [2003], HARD Track Overview in TREC 2003 High Accuracy Retrieval from Documents, *in* 'TREC'.

Allan, J. [2004], HARD Track Overview in TREC 2004 High Accuracy Retrieval from Documents, *in* 'TREC'.

Allan, J. [2005], HARD Track Overview in TREC 2005 High Accuracy Retrieval from Documents, *in* 'TREC'.

Allan, J., Kanoulas, E., Li, D., Gysel, C. V., Harman, D. and Voorhees, E. [2017], TREC 2017 Common Core Track Overview, *in* 'TREC'.

Anagnostou, A., Lagopoulos, A., Tsoumakas, G. and Vlahavas, I. P. [2017], Combining inter-review learning-to-rank and intra-review incremental training for title and abstract screening in systematic reviews, *in* 'Working Notes of CLEF 2017 - Conference and Labs of the Evaluation Forum, Dublin, Ireland, September 11-14, 2017.'.

Aslam, J. A., Pavlu, V. and Savell, R. [2003], A unified model for metasearch, pooling, and system evaluation, *in* 'Proceedings of the twelfth international conference on Information and knowledge management', ACM, pp. 484–491.

Baeza-Yates, R., Ribeiro, B. d. A. N. et al. [2011], *Modern information retrieval*, New York: ACM Press; Harlow, England: Addison-Wesley,.

Bagdouri, M., Webber, W., Lewis, D. D. and Oard, D. W. [2013], Towards minimizing the annotation cost of certified text classification, *in* 'Proceedings of the 22nd ACM international conference on Conference on information & knowledge management', ACM, pp. 989–998.

Baron, J. R., Lewis, D. D. and Oard, D. W. [2006], TREC 2006 Legal Track Overview, *in* 'TREC'.

Bates, D., Mächler, M., Bolker, B. and Walker, S. [2015], 'Fitting Linear Mixed-Effects Models Using lme4', *Journal of Statistical Software* **67**(1), 1–48.

Bendersky, M. and Kurland, O. [2010], 'Utilizing passage-based language models for ad hoc document retrieval', *Information Retrieval* **13**(2), 157–187.

Blair, D. C. [1996], 'STAIRS redux: Thoughts on the STAIRS evaluation, ten years after', *JASIS* **47**(1), 4–22.

Blair, D. C. [2002], 'Some thoughts on the reported results of TREC', *Information processing & management* **38**(3), 445–451.

Blair, D. C. and Maron, M. E. [1985], 'An evaluation of retrieval effectiveness for a full-text document-retrieval system', *Communications of the ACM* **28**(3), 289–299.

Bommannavar, P., Lin, J. and Rajaraman, A. [2016], Estimating topical volume in social media streams, *in* 'SAC', pp. 1096–1101.

Borden, B. B. [2010], 'The demise of linear review', *Williams Mullen E-Discovery Alert* .

Buckley, C. and Robertson, S. [2008], Relevance feedback track overview: TREC 2008, *in* 'TREC'.

Büttcher, S., Clarke, C. L. and Cormack, G. V. [2016], *Information retrieval: Implementing and evaluating search engines*, Mit Press.

Callan, J. P. [1994], Passage-level evidence in document retrieval, *in* 'Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval', Springer-Verlag New York, Inc., pp. 302–310.

Carroll, L. [2013], 'The Grossman-Cormack glossary of technology-assisted review', *Federal Courts Law Review* **7**(1).

Carterette, B. [2009], On rank correlation and the distance between rankings, *in* 'Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval', ACM, pp. 436–443.

Carterette, B., Allan, J. and Sitaraman, R. [2006], Minimal test collections for retrieval evaluation, *in* 'Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval', SIGIR '06, ACM, New York, NY, USA, pp. 268–275.

Clarke, C. L. and Cormack, G. V. [1996], Interactive substring retrieval (MultiText experiments for TREC-5)., *in* 'TREC'.

Clarke, C. L., Cormack, G. V. and Burkowski, F. J. [1995], Shortest substring ranking (MultiText experiments for TREC-4), *in* 'TREC', Vol. 4, pp. 295–304.

Cormack, G. V. [2007], TREC 2007 Spam Track Overview, *in* 'TREC'.

Cormack, G. V., Clarke, C. L. and Buettcher, S. [2009], Reciprocal rank fusion outperforms condorcet and individual rank learning methods, *in* 'Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval', ACM, pp. 758–759.

Cormack, G. V., Clarke, C. L., Palmer, C. R. and To, S. S. [1997], Passage-based refinement (MultiText experiments for TREC-6), *in* 'TREC', Citeseer, pp. 303–319.

Cormack, G. V. and Grossman, M. R. [2014], Evaluation of machine-learning protocols for technology-assisted review in electronic discovery, *in* 'Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval', ACM, pp. 153–162.

Cormack, G. V. and Grossman, M. R. [2015*a*], 'Autonomy and reliability of continuous active learning for technology-assisted review', *CoRR* **abs/1504.06868**.

Cormack, G. V. and Grossman, M. R. [2015*b*], Waterloo (Cormack) participation in the TREC 2015 Total Recall Track., *in* 'TREC'.

Cormack, G. V. and Grossman, M. R. [2016*a*], Engineering quality and reliability in technology-assisted review, *in* 'Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval', ACM, pp. 75–84.

Cormack, G. V. and Grossman, M. R. [2016*b*], Scalability of continuous active learning for reliable high-recall text classification, *in* 'Proceedings of the 25th ACM International on Conference on Information and Knowledge Management', ACM, pp. 1039–1048.

Cormack, G. V. and Grossman, M. R. [2017], Technology-Assisted Review in Empirical Medicine: Waterloo Participation in CLEF eHealth 2017, *in* 'Working Notes of CLEF 2017 - Conference and Labs of the Evaluation Forum, Dublin, Ireland, September 11-14, 2017.'.

Cormack, G. V. and Grossman, M. R. [2018], Beyond pooling, *in* 'The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval', ACM, pp. 1169–1172.

Cormack, G. V., Grossman, M. R., Hedin, B. and Oard, D. W. [2010], Overview of the TREC 2010 legal track, *in* 'TREC'.

Cormack, G. V. and Mojdeh, M. [2009], Machine Learning for Information Retrieval: TREC 2009 Web, Relevance Feedback and Legal Tracks., *in* 'TREC'.

Cormack, G. V., Palmer, C. R. and Clarke, C. L. [1998], Efficient construction of large test collections, *in* 'Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval', ACM, pp. 282–289.

Cormack, G. V., Zhang, H., Ghelani, N., Abualsaud, M., Smucker, M. D., Grossman, M. R., Rahbariasl, S. and Ghenai, A. [2019], Dynamic sampling meets pooling, *in* 'SIGIR'.

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K. and Harshman, R. [1990], 'Indexing by latent semantic analysis', *Journal of the American society for information science* **41**(6), 391–407.

Di Nunzio, G. M. [2018], A study of an automatic stopping strategy for technologically assisted medical reviews, *in* 'European Conference on Information Retrieval', Springer, pp. 672–677.

Goeuriot, L., Kelly, L., Suominen, H., Névéol, A., Robert, A., Kanoulas, E., Spijker, R., Palotti, J. and Zuccon, G. [2017], CLEF 2017 eHealth evaluation lab overview, *in* 'International Conference of the Cross-Language Evaluation Forum for European Languages', Springer, pp. 291–303.

Grossman, M., Cormack, G. and Roegiest, A. [2016], 'TREC 2016 Total Recall Track Overview'.

Grossman, M. R. and Cormack, G. V. [2014], 'Comments on "The Implications of Rule 26 (g) on the Use of Technology-Assisted Review."', *Federal Courts Law Review* **1**.

Grossman, M. R. and Cormack, G. V. [2017], MRG_UWaterloo and WaterlooCormack Participation in the TREC 2017 Common Core track, *in* 'TREC'.

Grossman, M. R., Cormack, G. V., Hedin, B. and Oard, D. W. [2011], Overview of the TREC 2011 Legal Track, *in* 'TREC', Vol. 11.

Grossman, M. R., Cormack, G. V. and Roegiest, A. [2017], Automatic and semi-automatic document selection for technology-assisted review, *in* 'Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval', ACM, pp. 905–908.

Hannaford-Agor, P. [2013], 'Measuring the cost of civil litigation: Findings from a survey of trial lawyers', *Voir Dire* pp. 22–28.

Harman, D. [2011], *Information Retrieval Evaluation*, 1st edn, Morgan & Claypool Publishers.

Hearst, M. A. [1994], Multi-paragraph segmentation of expository text, *in* 'Proceedings of the 32nd annual meeting on Association for Computational Linguistics', Association for Computational Linguistics, pp. 9–16.

Hearst, M. A. and Plaunt, C. [1993], Subtopic structuring for full-length document access, *in* 'Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval', ACM, pp. 59–68.

Hedin, B., Tomlinson, S., Baron, J. R. and Oard, D. W. [2009], Overview of the TREC 2009 legal track, *in* 'TREC'.

Hersh, W. and Over, P. D. [2002], TREC-2001 interactive track report, *in* 'TREC'.

Hersh, W. R. [2002], TREC 2002 interactive track report, *in* 'TREC'.

Hofmann, T. [1999], Probabilistic latent semantic analysis, *in* 'Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence', Morgan Kaufmann Publishers Inc., pp. 289–296.

Hogan, C., Reinhart, J., Brassil, D., Gerber, M., Rugani, S. M. and Jade, T. [2008], H5 at TREC 2008 legal interactive: user modeling, assessment & measurement, *in* 'TREC'.

Järvelin, K. and Kekäläinen, J. [2002], 'Cumulated gain-based evaluation of IR techniques', *ACM Transactions on Information Systems (TOIS)* **20**(4), 422–446.

Johnson, N., Kemp, A. and Kotz, S. [2006], *Univariate Discrete Distributions, 3rd Edition*, Wiley.

Jones, K. S., Walker, S. and Robertson, S. E. [2000], 'A probabilistic model of information retrieval: development and comparative experiments: Part 2', *Information processing & management* **36**(6), 809–840.

Kanoulas, E., Li, D., Azzopardi, L. and Spijker, R. [2017], 'CLEF 2017 technologically assisted reviews in empirical medicine overview', *Working Notes of CLEF* pp. 11–14.

Kaszkiel, M. and Zobel, J. [2001], 'Effective ranking with arbitrary passages', *Journal of the Association for Information Science and Technology* **52**(4), 344–364.

Kendall, M. G. [1938], 'A new measure of rank correlation', *Biometrika* **30**(1/2), 81–93.

Kim, J. and Kim, M. H. [2004], 'An evaluation of passage-based text categorization', *Journal of Intelligent Information Systems* **23**(1), 47–65.

Krikon, E. and Kurland, O. [2011], 'A study of the integration of passage-, document-, and cluster-based information for re-ranking search results', *Information retrieval* **14**(6), 593.

Kullback, S. and Leibler, R. A. [1951], 'On information and sufficiency', *The annals of mathematical statistics* **22**(1), 79–86.

Lafferty, J. and Zhai, C. [2001], Document language models, query models, and risk minimization for information retrieval, *in* 'Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval', ACM, pp. 111–119.

Lewis, D. D. [1995], 'The TREC-4 filtering track', pp. 165–180.

Lewis, D. D. and Gale, W. A. [1994], A sequential algorithm for training text classifiers, *in* 'Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval', Springer-Verlag New York, Inc., pp. 3–12.

Liu, X. and Croft, W. B. [2002], Passage retrieval based on language models, *in* 'Proceedings of the eleventh international conference on Information and knowledge management', ACM, pp. 375–382.

Losada, D. E., Parapar, J. and Barreiro, Á. [2016], Feeling lucky?: multi-armed bandits for ordering judgements in pooling-based evaluation, *in* 'proceedings of the 31st annual ACM symposium on applied computing', ACM, pp. 1027–1034.

Macaulay, A. [2014], 'The billable hourhere to stay?', https://www.cba.org/Publications-Resources/CBA-Practice-Link/solo/2014/The-Billable-Hour%E2%80%94Here-to-Stay. Accessed: 2014-03-12.

Maddalena, E., Basaldella, M., De Nart, D., Degl'Innocenti, D., Mizzaro, S. and Demartini, G. [2016], Crowdsourcing relevance assessments: The unexpected benefits of limiting the time to judge, *in* 'Fourth AAAI Conference on Human Computation and Crowdsourcing', pp. 129–138.

Mani, I., Klein, G., House, D., Hirschman, L., Firmin, T. and Sundheim, B. [2002], 'SUMMAC: a text summarization evaluation', *Natural Language Engineering* **8**(1), 43–68.

Mazanec, K. [2014], 'Capping e-discovery costs: a hybrid solution to e-discovery abuse', *Wm. & Mary L. Rev.* **56**, 631.

Metzler, D. and Croft, W. B. [2005], A markov random field model for term dependencies, *in* 'Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval', ACM, pp. 472–479.

Milne, D. [2014], 'Wikipediaminer', github. https://github.com/dnmilne/wikipediaminer.

Montague, M. and Aslam, J. A. [2002], Condorcet fusion for improved retrieval, *in* 'Proceedings of the eleventh international conference on Information and knowledge management', ACM, pp. 538–548.

Munoz, M. and Nagarajan, R. [2001], 'Sentence splitter'. Cognitive Computation Group, Dept. CS, UIUC, http://cogcomp.org/page/tools_view/2.

Oard, D. W., Baron, J. R., Hedin, B., Lewis, D. D. and Tomlinson, S. [2010], 'Evaluation of information retrieval for E-discovery', *Artificial Intelligence and Law* **18**(4), 347–386.

Oard, D. W., Hedin, B., Tomlinson, S. and Baron, J. R. [2008], Overview of the TREC 2008 legal track, *in* 'TREC'.

Oard, D. W., Webber, W. et al. [2013], 'Information retrieval for e-discovery', *Foundations and Trends® in Information Retrieval* **7**(2–3), 99–237.

Pavlu, V. [2008], *Large scale IR evaluation*, Northeastern University.

Pavlu, V. and Aslam, J. [2007], 'A practical sampling strategy for efficient retrieval evaluation', *College of Computer and Information Science, Northeastern University* .

Peacock, M. [2009], 'The true cost of ediscovery', https://www.cmswire.com/cms/enterprise-cms/the-true-cost-of-ediscovery-006060.php. Accessed: 2009-11-17.

Prince, M. [2004], 'Does active learning work? A review of the research', *Journal of engineering education* **93**(3), 223–231.

R Core Team [2014], *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria.

Rahbariasl, Shahin [2018], The effects of time constraints and document excerpts on relevance assessing behavior, Master's thesis.

Robertson, S. [2002], 'Introduction to the special issue: Overview of the TREC routing and filtering tasks'.

Robertson, S. E. and Soboroff, I. [2002], The TREC 2002 Filtering Track Report, *in* 'TREC', Vol. 2002, p. 5.

Roegiest, A. [2017], 'On Design and Evaluation of High-Recall Retrieval Systems for Electronic-Discovery'.

Roegiest, A., Cormack, G., Grossman, M. and Clarke, C. [2015], TREC 2015 Total Recall track overview, *in* 'TREC'.

Roegiest, A., Cormack, G. V., Clarke, C. L. and Grossman, M. R. [2015], Impact of surrogate assessments on high-recall retrieval, *in* 'Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval', ACM, pp. 555–564.

Roitblat, H. L., Kershaw, A. and Oot, P. [2010], 'Document categorization in legal electronic discovery: computer classification vs. manual review', *Journal of the Association for Information Science and Technology* **61**(1), 70–80.

Salton, G., Allan, J. and Buckley, C. [1993], Approaches to passage retrieval in full text information systems, *in* 'Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval', ACM, pp. 49–58.

Sanderson, M. [1998], Accurate user directed summarization from existing tools, *in* 'Proceedings of the seventh international conference on Information and knowledge management', ACM, pp. 45–51.

Sanderson, M. and Joho, H. [2004], Forming test collections with no system pooling, *in* 'Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval', ACM, pp. 33–40.

Sandhaus, E. [2008], 'The New York Times Annotated Corpus'. LDC Catalog No.: LDC2008T19, https://catalog.ldc.upenn.edu/ldc2008t19.

Schieneman, K. et al. [2013], 'The implications of Rule 26 (g) on the use of technology-assisted review', *Fed. Cts. L. Rev.* **2013**, 239–239.

Sculley, D. and Cormack, G. V. [2009], 'Going mini: Extreme lightweight spam filters'.

Settles, B. [2009], Active learning literature survey, Computer Sciences Technical Report 1648, University of Wisconsin–Madison.

Settles, B. [2012], 'Active learning', *Synthesis Lectures on Artificial Intelligence and Machine Learning* **6**(1), 1–114.

Smucker, M. D. and Jethani, C. P. [2010], Human performance and retrieval precision revisited, *in* 'Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval', ACM, pp. 595–602.

Smucker, M. D. and Jethani, C. P. [2011*a*], The crowd vs. the lab: A comparison of crowd-sourced and university laboratory participant behavior, *in* 'Proceedings of the SIGIR 2011 Workshop on crowdsourcing for information retrieval'.

Smucker, M. D. and Jethani, C. P. [2011*b*], Measuring assessor accuracy: a comparison of nist assessors and user study participants, *in* 'Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval', ACM, pp. 1231–1232.

Smucker, M. D., Kazai, G. and Lease, M. [2012], Overview of the TREC 2012 crowdsourcing track, *in* 'TREC'.

Soboroff, I. and Robertson, S. [2003], Building a filtering test collection for TREC 2002, *in* 'SIGIR', ACM, pp. 243–250.

Strohman, T., Metzler, D., Turtle, H. and Croft, W. B. [2005], Indri: A language-model based search engine for complex queries (extended version), Technical Report IR-407, CIIR, CS Dept., U. of Mass. Amherst.

Tillé, Y. [2006], *Sampling Algorithms*, Springer Series in Statistics, Springer.

Tombros, A. and Sanderson, M. [1998], Advantages of query biased summaries in information retrieval, *in* 'Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval', ACM, pp. 2–10.

Tomlinson, S., Oard, D. W., Baron, J. R. and Thompson, P. [2007], Overview of the TREC 2007 Legal Track, *in* 'TREC'.

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

Urbano, J. and Marrero, M. [2017], The treatment of ties in AP correlation, *in* 'ACM International Conference on the Theory of Information Retrieval', pp. 321–324.

Vermorel, J. and Mohri, M. [2005], Multi-armed bandit algorithms and empirical evaluation, *in* 'European conference on machine learning', Springer, pp. 437–448.

Voorhees, E. M. [2000], 'Variations in relevance judgments and the measurement of retrieval effectiveness', *Information processing & management* **36**(5), 697–716.

Voorhees, E. M. [2001*a*], Evaluation by highly relevant documents, *in* 'Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval', ACM, pp. 74–82.

Voorhees, E. M. [2001*b*], The philosophy of information retrieval evaluation, *in* 'Workshop of the Cross-Language Evaluation Forum for European Languages', Springer, pp. 355–370.

Voorhees, E. M. [2018], On building fair and reusable test collections using bandit techniques, *in* 'Proceedings of the 27th ACM International Conference on Information and Knowledge Management', ACM, pp. 407–416.

Voorhees, E. M. and Harman, D. [2000], Overview of the Eighth Text REtrieval Conference (TREC-8), *in* 'TREC', pp. 1–24.

Voorhees, E. M. and Harman, D. K. [2005], 'The text retrieval conference', *TREC: Experiment and evaluation in information retrieval* pp. 3–19.

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

Wackerly, D., Mendenhall, W. and Scheaffer, R. [2007], *Mathematical statistics with applications*, Nelson Education.

Wang, J. [2011], Accuracy, agreement, speed, and perceived difficulty of users relevance judgments for e-discovery, *in* 'Proceedings of SIGIR Information Retrieval for E-Discovery Workshop', Vol. 1.

Wang, J. and Soergel, D. [2010], A user study of relevance judgments for e-discovery, *in* 'Proceedings of the 73rd ASIS&T Annual Meeting on Navigating Streams in an Information Ecosystem-Volume 47', American Society for Information Science, p. 74.

Wang, M. and Si, L. [2008], Discriminative probabilistic models for passage based retrieval, *in* 'Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval', ACM, pp. 419–426.

Webber, W., Bagdouri, M., Lewis, D. D. and Oard, D. W. [2013], Sequential testing in classifier evaluation yields biased estimates of effectiveness, *in* 'Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval', ACM, pp. 933–936.

Webber, W. and Pickens, J. [2013], Assessor disagreement and text classifier accuracy, *in* 'Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval', ACM, pp. 929–932.

Wood, L. [2017], 'Global ediscovery market analysis and forecasts 2016-2023: Market was valued at 6 billion in 2016 and expected to reach 13 billion by 2023'. Accessed: 2017-12-13.
URL: *https://prn.to/2UKkcv6*

Yang, H. and Frank, J. A. [2016], TREC 2015 Dynamic Domain Track Overview, *in* 'TREC'.

Yang, W., Zhang, H. and Lin, J. [2019], 'Simple applications of bert for ad hoc document retrieval', *CoRR* **abs/1903.10972**.

Yilmaz, E. and Aslam, J. A. [2006], Estimating average precision with incomplete and imperfect judgments, *in* 'Proceedings of the 15th ACM international conference on Information and knowledge management', ACM, pp. 102–111.

Yilmaz, E., Aslam, J. A. and Robertson, S. [2008], A new rank correlation coefficient for information retrieval, *in* 'Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval', SIGIR '08, ACM, New York, NY, USA, pp. 587–594.

Yilmaz, E., Kanoulas, E. and Aslam, J. A. [2008], A simple and efficient sampling method for estimating AP and NDCG, *in* 'Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval', ACM, pp. 603–610.

Yulianti, E., Chen, R.-C., Scholer, F., Croft, W. B. and Sanderson, M. [2018], Ranking documents by answer-passage quality, *in* 'Proceedings of the 41th international ACM SIGIR conference on Research and development in information retrieval', ACM.

Zhai, C. and Lafferty, J. [2001], Model-based feedback in the language modeling approach to information retrieval, *in* 'Proceedings of the tenth international conference on Information and knowledge management', ACM, pp. 403–410.

Zhang, H., Abualsaud, M., Ghelani, N., Ghosh, A., Smucker, M. D., Cormack, G. V. and Grossman, M. R. [2017], UWaterlooMDS at the TREC 2017 Common Core track, *in* 'TREC'.

Zhang, H., Abualsaud, M., Ghelani, N., Smucker, M. D., Cormack, G. V. and Grossman, M. R. [2018], Effective user interaction for high-recall retrieval: Less is more, *in* 'Proceedings of the 27th ACM International Conference on Information and Knowledge Management', CIKM '18, ACM, New York, NY, USA, pp. 187–196.

Zhang, H., Abualsaud, M. and Smucker, M. D. [2018], A study of immediate requery behavior in search, *in* 'Proceedings of the 2018 Conference on Human Information Interaction & Retrieval', CHIIR '18, ACM, New York, NY, USA, pp. 181–190.

Zhang, H., Cormack, G. V., Grossman, M. R. and Smucker, M. D. [2018], Evaluating sentence-level relevance feedback for high-recall information retrieval.

Zhang, H., Lin, J., Cormack, G. V. and Smucker, M. D. [2016], Sampling strategies and active learning for volume estimation, *in* 'Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval', SIGIR '16, ACM, New York, NY, USA, pp. 981–984.

Zhang, H., Lin, W., Wang, Y., Clarke, C. L. and Smucker, M. D. [2015], WaterlooClarke: TREC 2015 Total Recall Track, *in* 'TREC'.

Zhang, H., Rao, J., Lin, J. J. and Smucker, M. D. [2017], Automatically extracting high-quality negative examples for answer selection in question answering, *in* 'Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Shinjuku, Tokyo, Japan, August 7-11, 2017', pp. 797–800.

Zhang, L. and Zhang, Y. [2010], Interactive retrieval based on faceted feedback, *in* 'Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval', ACM, pp. 363–370.

Zhang, L., Zhang, Y., de Arma, J. and Yu, K. [2009], UCSC at relevance feedback track, Technical report, CALIFORNIA UNIV SANTA CRUZ SCHOOL OF ENGINEERING.

Zobel, J. [1998], How reliable are the results of large-scale information retrieval experiments?, *in* 'Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval', ACM, pp. 307–314.