Spring 2011

Mining for evidence in enterprise corpora

Brian Alan Almquist
University of Iowa

MINING FOR EVIDENCE IN ENTERPRISE CORPORA

by

Brian Alan Almquist

An Abstract

Of a thesis submitted in partial fulfillment
of the requirements for the Doctor of
Philosophy degree in Business Administration
in the Graduate College of
The University of Iowa

May 2011

Thesis Supervisor: Professor Padmini Srinivasan

ABSTRACT

The primary research aim of this dissertation is to identify the strategies that best meet the information retrieval needs as expressed in the "e-discovery" scenario. This task calls for a high-recall system that, in response to a request for all available relevant documents to a legal complaint, effectively prioritizes documents from an enterprise document collection in order of likelihood of relevance. High recall information retrieval strategies, such as those employed for e-discovery and patent or medical literature searches, reflect high costs when relevant documents are missed, but they also carry high document review costs.

Our approaches parallel the evaluation opportunities afforded by the TREC Legal Track. Within the ad hoc framework, we propose an approach that includes query field selection, techniques for mitigating OCR error, term weighting strategies, query language reduction, pseudo-relevance feedback using document metadata and terms extracted from documents, merging result sets, and biasing results to favor documents responsive to lawyer-negotiated queries. We conduct several experiments to identify effective parameters for each of these strategies.

Within the relevance feedback framework, we use an active learning approach informed by signals from collected prior relevance judgments and ranking data. We train a classifier to prioritize the unjudged documents retrieved using different ad hoc information retrieval techniques applied to the same topic. We demonstrate significant improvements over heuristic rank aggregation strategies when choosing from a relatively small pool of documents. With a larger pool of documents, we validate the effectiveness of the merging strategy as a means to increase recall, but that sparseness of judgment data prevents effective ranking by the classifier-based ranker.

We conclude our research by optimizing the classifier-based ranker and applying it to other high recall datasets. Our concluding experiments consider the potential

benefits to be derived by modifying the merged runs using methods derived from social choice models. We find that this technique, Local Kemenization, is hampered by the large number of documents and the minimal number of contributing result sets to the ranked list. This two-stage approach to high-recall information retrieval tasks continues to offer a rich set of research questions for future research.

Abstract Approved: _____
Thesis Supervisor

_____
Title and Department

_____
Date

MINING FOR EVIDENCE IN ENTERPRISE CORPORA

by

Brian Alan Almquist

A thesis submitted in partial fulfillment
of the requirements for the Doctor of
Philosophy degree in Business Administration
in the Graduate College of
The University of Iowa

May 2011

Thesis Supervisor: Professor Padmini Srinivasan

Graduate College
The University of Iowa
Iowa City, Iowa

CERTIFICATE OF APPROVAL

_____

PH.D. THESIS

_____

This is to certify that the Ph.D. thesis of

Brian Alan Almquist

has been approved by the Examining Committee
for the thesis requirement for the Doctor of Philosophy
degree in Business Administration at the May 2011 graduation.

Thesis Committee: _____
Padmini Srinivasan, Thesis Supervisor

_____
Nick Street

_____
Warren Boe

_____
David Eichmann

_____
Kate Cowles

Dedicated to Sharon, whose determined encouragement and support made this possible.

ACKNOWLEDGMENTS

By most measures, this project has taken a long time to reach its current state. Such an effort would not be possible without the contributions and support of many individuals. I shudder at the thought of attempting to identify all the assistance from people without whom this dissertation would still be just a blank sheaf of pages. My chair, Professor Srinivasan, provided invaluable mentoring as she drew me into information retrieval research. Her insights, expertise, and curiosity have informed every step of this dissertation, from initial efforts at large scale indexing, to experiments carefully designed to refine promising results, to persistent editorial recommendations. When I struggled with this project, she pushed me through with her generously offered encouragement.

I am deeply indebted to Professor Street for grounding me in data mining strategies. I would also like to thank my other committee members, Professor Boe, Professor Cowles and Professor Eichmann for their continued flow of recommendations and advice. Professor Filippo Menczer, Professor Shannon Bradshaw, Marc Light and Faiz Currim have all provided impetus for all of my doctoral studies, and their teachings influence my interest in this thesis topic. Finally, I am grateful to Professor Sean Gouglas, Professor David Mills, and Professor Ken Mouré for arranging access to the resources of the Humanities Computing Program at the University of Alberta

I must express my heartfelt appreciation for the insightful comments and edits that I received from my friends in the University of Iowa graduate student community. I want to acknowledge the close support from Xin Ying Qiu, Thaddeus Sim, Kaan Ataman, Yelena Mejova, Viet Ha-Thuc, Aditya Sehgal, Chris Harris, and Robert Arens. The members of the ISOR lab, Data Mining Interest Group, and the Text Retrieval and Text Mining reading group have all provided advice and venues for sounding out early presentations of my research.

In the end, none of this would have happened without the love, strength, and patience that flowed from my father, mother, brother and sister. Sharon Romeo and I began this adventure together as undergraduates. Her interest in legal history sparked my curiosity in legal documents, evidence, and retrieval methods. Her advice, continued support, and tolerance have helped sustain me as I grappled with the many struggles of my graduate studies.

ABSTRACT

The primary research aim of this dissertation is to identify the strategies that best meet the information retrieval needs as expressed in the "e-discovery" scenario. This task calls for a high-recall system that, in response to a request for all available relevant documents to a legal complaint, effectively prioritizes documents from an enterprise document collection in order of likelihood of relevance. High recall information retrieval strategies, such as those employed for e-discovery and patent or medical literature searches, reflect high costs when relevant documents are missed, but they also carry high document review costs.

Our approaches parallel the evaluation opportunities afforded by the TREC Legal Track. Within the ad hoc framework, we propose an approach that includes query field selection, techniques for mitigating OCR error, term weighting strategies, query language reduction, pseudo-relevance feedback using document metadata and terms extracted from documents, merging result sets, and biasing results to favor documents responsive to lawyer-negotiated queries. We conduct several experiments to identify effective parameters for each of these strategies.

Within the relevance feedback framework, we use an active learning approach informed by signals from collected prior relevance judgments and ranking data. We train a classifier to prioritize the unjudged documents retrieved using different ad hoc information retrieval techniques applied to the same topic. We demonstrate significant improvements over heuristic rank aggregation strategies when choosing from a relatively small pool of documents. With a larger pool of documents, we validate the effectiveness of the merging strategy as a means to increase recall, but that sparseness of judgment data prevents effective ranking by the classifier-based ranker.

We conclude our research by optimizing the classifier-based ranker and applying it to other high recall datasets. Our concluding experiments consider the potential

benefits to be derived by modifying the merged runs using methods derived from social choice models. We find that this technique, Local Kemenization, is hampered by the large number of documents and the minimal number of contributing result sets to the ranked list. This two-stage approach to high-recall information retrieval tasks continues to offer a rich set of research questions for future research.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I
INTRODUCTION: CHALLENGES IN LEGAL INFORMATION RETRIEVAL

1.1 E-Discovery

Large organizations, including corporations, are finding themselves increasingly required to store and maintain digital versions of all business documents. While this has its own advantages for facilitating the flow of information throughout an organization, the requirement is often driven by legal needs. In recent lawsuits, defendants have been sanctioned for failing to maintain an electronically searchable repository of business records [8]. In standard litigation processes, a plaintiff is entitled to demand from a defendant any evidence—including evidence in electronic format, such as e-mails, memos, reports and spreadsheets—which may be relevant to the complaint. This process, known as "discovery," is subject to negotiations between the parties over the meaning of relevance, determining which evidence is "responsive" to the complaint, and which can be withheld as "privileged." The current practice for finding relevant documents in a corporate document collection entails the use of keyword searches, with additional potential negotiations between the parties over the parameters of this search, including any Boolean constraints. Documents returned from the search are then reviewed for responsiveness and privilege before being handed over to the complainant.

This "e-discovery" context for search on collections of electronically stored information provides new venues for research in Information Retrieval. Litigation requires a search process that retrieves all relevant documents to a query, as opposed to the more common search request satisfied with a sample of relevant documents.

The ability to electronically search an organization's documents for evidence is a relatively recent development in legal procedure. Civil procedure rules now take account of this ability; failing to produce electronically searchable document collections has

resulted in expensive losing verdicts, an increasing number of sanctions against both litigants and counsel, and even jail sentences [95, 98]. These rules provide for and guide negotiations between the litigating parties over the technical aspects of the search technology and the document collections. Procedures for conducting the searches are negotiated between the parties, as well as the actual search formulations. These rules for civil procedure also allow for changes due to "developments in computer technology," creating an environment where current research can significantly influence the operations of the legal system.

The current practice in e-discovery ranges from complete retrieval, where all documents in a collection are manually evaluated, to a system where all documents resulting from a key-word query modified by Boolean constraints are reviewed. Complicating the e-discovery search problem is the cost of manually reviewing the documents for suitability as evidence. In the context of the legal system, only lawyers are considered to be qualified evaluators of evidence. Compensating lawyers for performing such reviews can be very costly. Worker placement agencies now provide lawyers as temporary labor for large document review projects [38].

Furthermore, the discovery of evidence in collections of electronically stored information is made more difficult by the heterogeneity of organizational documents, which may include e-mail, reports, memos, and multimedia components. Moreover, digitization is required to make older documents, which may only exist as hard copy, searchable.

1.2 TREC Legal Track

The Text REtrieval Conference (TREC) is an annual series of workshops initiated in 1992 by its co-sponsors, the National Institute of Standards and Technology and the U.S. Department of Defense, for the purpose of, "encouraging research in information retrieval on large test collections." [84, 96] The TREC conferences is organized around a

set of workshops, or "tracks," addressing an area of focus. Examples of focus areas include tracks on mining sentiment analysis from blogs, locating experts using organizational document collections, and developing strategies for handling increasingly larger sets of documents. Within this framework, the Legal Track was introduced in 2006. Designed to provide a venue for exploring questions raised by e-discovery, participants were asked to find documents relevant to a series of fictional complaints. The document collection, amalgamated from six tobacco companies and their affiliated think tank, consists largely of digitized versions of paper documents, including correspondence, printed e-mails, ad proofs, and scientific articles. The complaints were designed to resemble those in lawsuits that might be filed against tobacco companies.

The Legal Track was added to the conference program in 2006 with the aim of addressing the particular needs of lawyers conducting or preparing for potential e-discovery processes. The primary challenge of the track is to create an information retrieval system that retrieves more relevant documents than the specified baseline procedure—the results of a Boolean keyword search manually constructed by legal experts. The secondary goal is to improve precision, so that items at the beginning of a retrieved set of documents are most likely to be relevant. In 2007 a relevance feedback task was added, which extends the basic retrieval task by allowing participants to use information about documents identified as relevant in 2006 for training their retrieval systems on the same problems.

1.3 Research Questions

Our research goals align with those of the TREC Legal Track. We consider the e-discovery problem, as formulated by the track organizers, as an example of a high-recall task, with a ranking component to improve document review efficiency. We break this into two subtasks: ad hoc retrieval using information about a request for production of evidence to formulate an initial query, and relevance feedback where we execute

modified queries or refine our results using information gleaned from queries executed previously. We discuss these two tasks in greater detail in the next two chapters.

Within the information retrieval community, the ad hoc retrieval task is seen as the basic problem [58]. An information need is a topic about which a user desires additional information. The information need is translated into a query, which the information retrieval system uses as the basis for identifying relevant documents from within a specified collection. In the e-discovery context for this task, the collection is a set of documents maintained at the enterprise level. The information need is a demand for all documents that are relevant to a specific complaint. Documents are considered to be relevant if they refer to the topic.

In addressing the problem of ad hoc search for e-discovery, we wish to design a suite of information retrieval strategies that are most successful at identifying relevant documents. Our questions concern the nature of the strategies and the nature of the enterprise document collection. We identify characteristics that are specific to the e-discovery ad hoc task. These characteristics reflect both the type of information sought, and the documents where the information is to be found. As an example of the former, we consider that the information request is expressed as a request for production of documents, along with substantial supplementary material that reflects the litigation context. We characterize the documents as existing in a collection maintained by an organization, deriving from diverse sources. Can standard information retrieval strategies be adapted to the e-discovery task, and if so, how? In Chapter 4, we propose a series of methods that we have tuned to optimize ad hoc retrieval. We also identify traits of the problem that offer continued channels for researching additional improvements.

In information retrieval, relevance feedback is understood to reflect a situation where, after an initial query, the user evaluates a sample of the retrieved documents. Information about the documents determined to be relevant is then typically used to modify the initial query. In broader terms, we consider relevance feedback to reflect any

scenario where prior information about document relevance is used to inform retrieval for the same topic.

We identify the review of retrieved documents as a major cost for e-discovery and other high recall-oriented tasks. In Chapter 5, we link this problem to relevance feedback. We propose a supervised classifier that uses information about documents that are known to be relevant and non-relevant to prioritize other documents for review. Our research questions concern rank aggregation methods and evaluating the performance of such systems. What are appropriate comparison strategies? What are appropriate comparison measures? Is this strategy scalable and applicable to the high-recall retrieval task?

In Chapter 6, we identify further research questions surrounding the use of the classifier as a ranking method. Can we apply the ranker to datasets that do not reflect the e-discovery task? Can we combine the output of the classifier with the output of other strategies to generate improved results? Can we find additional measures to assess the effectiveness of these strategies at prioritizing documents for review? Finally, we ask whether the ideas of other fields, such as social choice theory, can be used to improve the results of different ranking strategies.

In each of these investigations, while we find interesting answers, we also find numerous directions that remain unexplored in this thesis. We conclude by identifying several potential future research opportunities.

CHAPTER II

TEXT RETRIEVAL IN THE LEGAL DOMAIN

2.1 Text Retrieval: An Overview

Text retrieval as a computational problem has a history that extends to nearly to the Second World War [87]. In 1945, Vannevar Bush identified the increasingly daunting "mountain of research" as a problem that would make finding scientific information a monumental task [17]. It did not take long for researchers to understand the potential for using general purpose computers for document retrieval [6].

Many of these scientists were primarily concerned with access to scientific literature. The problem of searching and retrieving information, identified by Maron and Kuhns as the "library problem," was only partially related to the exponential growth rate of documentation accumulating in many scientific libraries. Retrieval systems would need to do more than simply match terms in a manually generated index. "Meaning" in text was a concern, as it would be important for systems to estimate the relevance of a document for a given request, or identify which of two documents may be "closer" to a third document [60].

Maron and Kuhns proposed "Probabilistic Indexing,"where the information retrieval system, given a user's query, would predict, for each document, the probability that the user would actually want the document. These calculations are dependent on manual estimates of the likelihood that a user finding the document to be relevant had used a particular index term in their query. These probabilities would then be used to rank the documents, so that the user would be presented with a selection of documents that they would be most likely to find relevant. Maron and Kuhns propose using co-occurrence of indexing terms as the basis for query expansion, a form of relevance feedback. Maron and Kuhns paper also describes an empirical evaluation for their retrieval system. Since that point, there has been an ongoing effort to develop testbeds

for large-scale experimental evaluation of information retrieval systems, from the Cranfield Project to the ongoing intervention by the U.S. federal government through the Text REtrieval Conferences [22, 23, 39, 51].

Maron and Kuhn's proposal for feedback-informed query expansion is explicitly developed in greater detail and described as relevance feedback by Rocchio [72]. Using explicit feedback, queries are modified by adding terms from documents that are known to be relevant. Negative feedback can also be applied—by removing terms from the query that are found in documents known to be non-relevant—but this has not proven to be as effective. In the absence of knowledge of actual document relevance, it is possible to conduct similar "blind" feedback steps by assuming that top results from an initial query are relevant [20, 61].

The vector space model, as developed by Salton and his students [74], has developed into a powerful tool for information retrieval. In this model, documents in a collection are represented by *n*-dimensional vectors, where *n* is the number of distinct terms in the collection. In other words, a document D_i in the collection is modeled as $D_i = (d_{i1}, d_{i2}, \ldots, d_{in})$. Here, d_{ij} represents the weight assigned to the *j*th term in the collection. The query is similarly modeled. Weights are assigned to each term using many different methods [77], but the common approach involves the application of term frequency-inverted document frequency (TF-IDF) [80]. The weight for each term *t* in a document *d* is calculated as:

$$w_d^t = f_d^t \cdot \log \frac{|D|}{d_t} \tag{1}$$

where f_d^t is the frequency of the term within the document, $|D|$ is the number of documents in the collection, and d_t is the number of documents in the collection containing the term. A term in a particular document with a high TF-IDF score appears frequently in the document, but infrequently throughout the corpus. A document's

similarity to other documents—or a query, if the query is also modeled as a term vector—can then be calculated using a variety of methods, including cosine similarity.

As system capacity increased, full text document retrieval developed as a complement to human-curated bibliographic document retrieval systems. Previously, document retrieval systems relied on metadata supplied by human indexers. Commercially available document retrieval systems offering full text searching were presented to the legal community by Lexis-Nexis and West Publishing (Westlaw). These systems primarily offered users a Boolean query search interfaces, but eventually, both services also provided ranked retrieval to their customers. The indexed documents for these systems were the materials that had been accumulating in law libraries—laws and appeals court decisions from both the federal and state level. Nevertheless, the continued popularity of Boolean query searching within the legal community has had a clear impact on initial approaches for conducting search on electronically stored information.

2.1.1 Definitions

For the purposes of this dissertation, we will need to define several components of an information retrieval system as they will apply to our work. As this dissertation relies heavily on techniques from both information retrieval and data mining, we will attempt to place each term within the appropriate framework.

2.1.1.1 Document

A document is the unit that is to be indexed, and, more importantly, the unit that is returned to the user in response to a request for information. It is possible for a document to represent a segment from a larger text; the indexed document may only consist of a single chapter, page, paragraph or sentence. Within the e-discovery context, documents represent single pieces of evidence. In the index compiled for this project, documents range in length from the advertising copy from Marlboro ads to hundred-page reports. For the information retrieval portions of this thesis, we represent a document

(and a query) as an *n*-dimensional vector of weights, each representing the significance of the term in describing the document. As long as the collection of documents is static, the vector representing a given document will also be static. Since the query may also be represented as an *n*-dimensional vector, there is no need to modify the document vectors to reflect the context of a particular information need.

In data mining, the unit item to be classified or used for system training is referred to as an instance. In our system, a document is a single instance, represented as a vector of features used to describe the document. These features are context dependent—a particular document may be represented differently for different information needs.

2.1.1.2 Collection

The collection of documents, often referred to as the "corpus", consists of the set of documents that have been indexed by our system. Every document in the collection is evaluated by the information retrieval system, and then ranked in order of an estimated likelihood that the user will find the document to be relevant. In most implementations, only a set amount of the top-ranking documents will be presented to the user. For our data mining system, we construct a different collection of documents for each information need; to qualify for these smaller collections, a document will need to have been included in the set of retrieved documents by another information retrieval system.

2.1.1.3 Relevance

For the purposes of this dissertation, upon assessment, a document may be found to be "Relevant" or "Non-Relevant." For evaluation purposes, the relevant document is a positive match for the information need. In the context of the e-discovery task, assessing a document as relevant meant that the document contained a reference to the topic of the information request, and that the document met all practical requirements of the request (e.g., a request may be for "Internal Memoranda" only) [9].

The concept of relevance is subject to some discussion. Hodge and Milstead suggest that there are two important aspects to our idea of relevance: "Aboutness is the inherent subject of the document and is distinguished from the 'meaning' of the document, i.e., the reason for which the user is seeking the document or the purpose which it may serve for the user." [43] Taken together, we might consider "relevance" to mean that the user "wants" the document within the context of the topic [59]. The decision criteria for the e-discovery task are concerned with "aboutness," while "meaning" would be an important criterion for deciding whether to use the document as evidence.

2.1.2 Measures

In order to evaluate the success of different systems and strategies for the e-discovery problem, the organizers of the TREC Legal Track employ a set of both longstanding and newly devised information retrieval measures. As new measures are added to this set over the years, they more closely reflect the goals of the track, in particular the desire to maximize recall while avoiding a reduction in precision that would render assessment of retrieval results less affordable. What follows is a brief description of the measures.

2.1.2.1 Precision

Briefly, precision is the fraction of all documents retrieved for a query that are relevant [58]:

$$\begin{aligned}\text{Precision} &= \frac{|\{\text{relevant documents}\} \cap \{\text{retrieved documents}\}|}{|\{\text{retrieved documents}\}|} \\ &= P(relevant|retrieved)\end{aligned} \tag{2}$$

Precision is the equivalent of the positive predictive value of a classifier. Given the basic task of the e-discovery problem—find all relevant documents in the collection—the large volume of documents retrieved suggests that precision values will be very low. Precision by itself is not a very useful measure for us for assessing an entire retrieved set of documents. But we are interested in assessing the effectiveness of a system at placing relevant documents at the beginning of the ranked list. We can do this with precision measures at arbitrary depths.

2.1.2.2 Precision at *n*

R-Prec, or Precision at *R*, is precision calculated at the *R*th position of a ranked retrieval set. Here *R* is set to be the number of known relevant documents for the query. As with any straightforward precision measure, the placement of relevant documents within the top *R* documents in the ranked list is not reflected in the calculation. The calculation described above is thus modified:

$$\text{Prec@}R = \frac{|\{\text{relevant documents}\} \cap \{\text{top } R \text{ retrieved documents}\}|}{R} \qquad (3)$$

We can use *R* as a way to adjust for the number of relevant documents for a given topic—if *R* is less than *N* (the number of retrieved documents) the best possible precision value is $^{R}/_{N}$, while the best possible R-Prec value is 1.Though not strictly a recall-based measure, using the number of relevant documents and the flat weighting of the precision calculation made this a reasonable proxy for recall in the TREC 2006 Legal Track.

We use smaller values for *n* (i.e., Prec @ 5) when we wish to assess how effective a ranked retrieval system is at bringing relevant documents to the head of the ranking list.

2.1.2.3 Mean Average Precison (MAP)

Given a ranked retrieval set, we can calculate precision at each relevant document in the set. Average Precision is calculated as the mean of the collected precision calculations. If the set of relevant documents for a given topic is $\{d_1, \ldots, d_m\}$, and R_k is the result set of a ranked retrieval from the top ranking until you get to document d_k, then:

$$AP = \frac{1}{m} \sum_{k=1}^{m} \text{Precision}(R_k) \tag{4}$$

This approach provides a benefit to sets that successfully order relevant documents at the highest ranks. However, Precision(R_k) = 0 if d_k is not included in the set of retrieved documents. Thus, a system cannot "game" average precision by limiting retrieval to documents with high confidence of relevance. Mean Average Precision (MAP), then, is the mean of this score calculated over all topics as part of a TREC run. This measure was the primary precision measurement for the TREC Legal Track in 2006.

2.1.2.4 Recall

Recall is the equivalent of the sensitivity of a classifier. Recall is the proportion of all relevant documents in a document collection that are included in the specified result set:

$$\begin{aligned} \text{Recall} &= \frac{|\{\text{relevant documents}\} \cap \{\text{retrieved documents}\}|}{|\{\text{relevant documents}\}|} \\ &= P(retrieved|relevant) \end{aligned} \tag{5}$$

In e-discovery, we are concerned with finding all evidence that may be pertinent to a particular case. In this aspect, we consider one aspect of e-discovery to be a high-recall task; the more successful of two systems, each returning the same number of documents, will have found more relevant documents without regard to ranking.

The trivial way to maximize recall is to return all documents in a collection, so for recall to be a meaningful measure, recall is usually measured at a particular depth in the ranked retrieval, or the retrieved result set is capped at some size smaller than the document collection. The TREC Legal Track ultimately allowed result sets containing up to 100,000 documents (1.4% of the collection). With such a large result set, it is possible for a threshold precision measure (i.e., R-Prec) to have a low value paired with a high recall score if retrieved but relevant documents are ranked below the threshold.

One of the 2006 TREC Legal Track participants conducted a "depth probe," successfully establishing that relevant documents could be found at depths in a ranked retrieval far beyond that of the 5,000 documents requested by the track organizers [90]. For 2007, a new measure, Estimated Recall at *B* (Est-RB), was created to improve assessment for larger amounts of retrieved documents. Using a stratified sampling technique, the track organizers estimated the number of relevant documents for a topic in a collection by interpolating from an evaluated document's probability of being selected for evaluation. This estimation method was also used to estimate how many relevant documents a run will have retrieved at a particular depth. In this case, the set amount, *B*, is the number of documents retrieved by the Boolean "reference run." Thus, *B* is not an estimated value, but Est-RB is calculated using two estimates—an estimate of the number of relevant documents in the collection, and an estimated number of documents retrieved before the cutoff of *B*.

One side effect of calculating this measure is that it allowed the track organizers to conclude that the Boolean queries were not retrieving all of the relevant documents.

This verified that the goal of finding improvements over Boolean keyword search was a worthwhile project.

2.1.2.5 Balanced Measures

As pointed out above, a trivial operation to generate a high recall score is to simply return all available documents. But this same operation will also generate a very low precision score. The e-discovery task, as modeled by the TREC Legal Track organizers, reflects two different retrieval questions: high recall for the collection of all evidence relevant for a case, and high precision for easing the cost of actually evaluating the results.

The measures for the TREC 2008 Legal Track reflect the desire to calculate a measure that explicitly balances recall and precision. The *F*-measure allows for a balance by calculating the weighted harmonic mean of recall and precision:

$$F_\beta = \frac{(\beta^2 + 1)\text{precision} \times \text{recall}}{\beta^2 \times \text{precision} + \text{recall}} \quad (6)$$

β serves as a parameter used to weight either precision or recall. F_1, or the "balanced *F*-score," is thus calculated as:

$$F_1 = \frac{2\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \quad (7)$$

This score is estimated using the sampling probabilities in the same manner as for Est-RB. The estimated balanced *F*-score is calculated at at the estimated value for *R* (the smallest integer greater than or equal to the estimated number of relevant documents for the topic).

2.2 Information Retrieval and the Legal Community

Information retrieval is a task that the legal profession is familiar with. The system of legal precedence requires that lawyers must reinforce their arguments by properly referring to pre-existing decisions. By 1980, the Lexis database included the full text of all U.S. state and federal published case opinion, and it currently includes all U.S. statutes and laws. When reviewing a specific opinion, legal research systems feature "citation services," such as Westlaw's KeyCite system, that will inform the user of the current document's authoritative status by detailing the frequency and manner in which it has been cited, and whether a decision has been overruled or not. These databases often derive their power from topical indexes in the form of highly detailed taxonomies.

These precedent search systems use an extended Boolean query system for information retrieval. The influence of these retrieval systems is reflected in the Boolean queries currently used for e-discovery, particularly with the prevalence of proximity operators—constraints specifying that particular terms in a document must be placed within a certain number of words or pages of each other. Precedence search has a different functional objective than e-discovery. In the case of the former, the user is seeking relatively few documents and relevance considerations include an explicit temporal factor to determine the current binding precedent. These differences can be explained by the different roles that evidence and arguments play in legal proceedings.

2.3 The TREC Legal Track: Improving the Status Quo

The typical track at TREC is designed to address a specific problem. For instance, the Terabyte Track concerns itself with searches against large, web-based collections. The Confusion Track addressed information retrieval against document collections with large quantities of noise within the text. Tracks are also created to explore information retrieval problems associated with a particular domain. The

Chemical IR Track looks at chemical information retrieval methods, and the Microblog Track examines information seeking behaviors in microblogging (i.e., Twitter) environments. Track organizers arrange for the distribution of an existing document collection, or they frequently compile their own. Participants acquire the document collection and construct their own indexing and retrieval systems. The TREC organizers will also frequently provide a sampling of topics along with associated relevance judgments, thus allowing participants to train their systems in advance of the release of the actual topics. In order to provide sufficient diversity in queries, the organizers will create, or identify from existing search logs, several dozen to thousands of topics. Participant submissions are presented as ranked lists of documents representing the results of their system when applying the topic to the collection. Teams may often make multiple submissions, representing different configurations of the same system, or even different systems. The track organizers pool the submissions, and present a sampling from the pool to assessors to gather relevance judgments. The submissions are then evaluated using criteria established by the track organizers. In many cases, the purpose of a given track is to explore different evaluation methods in addition to identifying state-of-the-art information retrieval techniques.

The TREC Legal Track is a domain-focused track. The document collection was selected because if its similarity to enterprise document collections that might be subject to legal discovery; it is large, contains documents of many types with inconsistently applied metadata, and is subject to noise introduced in the process of converting paper documents into machine searchable formats. The stated goal of the track is to demonstrate that ranked retrieval can identify more relevant documents than a Boolean query designed by legal professionals. The document review benefit from ranked retrieval is a secondary purpose for the track. TREC Legal Track topics are manually designed and the product of negotiations conducted in roleplay. They provide extensive supplementary information that is atypical of topics in other tracks.

The TREC process employs legal experts to generate a set of relevance judgments on retrieved documents for each topic. No less than 500 documents were evaluated for each topic in TREC Legal. The TREC evaluators judged the top five ranked documents for each topic from each submitted run, and then picked the remaining documents for evaluation based on an estimated probability for selection. The Legal Track organizers estimate that they were the beneficiaries of more than 880 hours of volunteer time spent evaluating documents for the ad hoc task in 2007, and that such an effort conducted by "summer associate" lawyers would have cost about $132,000 [89]. This leaves thousands of retrieved documents unjudged for each query, and barely covers the number of documents retrieved by some of the Boolean constrained keyword queries. For most submissions to the TREC Legal Track, a lengthy traversal down the list of ranked documents is usually not required before one encounters documents that were not selected for evaluation. One of the stated reasons for conducting the Interactive and Relevance Feedback tasks for TREC Legal 2007 was to enrich the pool of judged documents for the 2006 topics. If an incremental amount of money were to become available for completing more judgments, it may be desirable to target documents most likely to be relevant for evaluation. Likewise, in an actual legal discovery situation, it may be desirable to target documents that are believed to have a high probability of being relevant.

Any system that potentially reduces the amount of time spent on review by professional-class labor is of value to the sponsors of the TREC Legal Track. The Boolean-constrained keyword searches offer such reduction, relative to reviewing the entire document collection, but estimates indicate that they may be only retrieving less than a quarter of all relevant documents in a collection [10]. Additionally, the documents retrieved by these searches are not ranked, thus extending the amount of time spent reviewing the results. The purpose of the TREC Legal Track is to explore whether some other algorithm can more effectively retrieve and rank the appropriate relevant

documents while simultaneously limiting the number of non-relevant documents retrieved [9].

In the first two years of the TREC Legal Track no team was able to create a system that was more effective at this task than the standard retrieval obtained by the baseline Boolean query. Still, there were a number of relevant documents found by these strategies that were missed by the baseline Boolean query. This provided encouragement to continue attempting to improve these processes. The TREC results have allowed the track organizers to draw conclusions about the nature of the dataset and the effectiveness of the state-of-the-art techniques, while also providing useful data that can serve as starting points for further research. As noted before, the presence of judgments from 2006 for documents in the dataset allows document retrieval informed by relevance feedback in 2007. New judgments from these later runs can be added to earlier judgments to improve the usefulness of the TREC Legal dataset as a standardized retrieval test bed.

2.4 Designing State-of-the-Art Algorithms:

E-Discovery vs. General Information Retrieval

The primary research aim of my dissertation is to design and test the strategies that will best meet the information retrieval needs as expressed in the "e-discovery" scenario. In other words, we wish to develop a high-recall system that effectively prioritizes documents with a higher likelihood of relevance in the context of legal discovery applied to an enterprise document collection. Our approaches parallel the two-task structure afforded by the TREC Legal Track, where *ad hoc* retrieval provides an initial set of resulting documents in response to a query, and the second where knowledge derived from *relevance feedback* is used to refine the results. Within each of these approaches, we will explore the relative merits of several strategies. We believe that

successful research will improve retrieval performance for comprehensive document searches, while increasing efficiency in the document review process.

Ad hoc retrieval and relevance feedback are common information retrieval tasks. In the context of legal research, the objectives of these tasks have been modified to reflect the search for pertinent evidence. For instance, a common ad hoc type query might reflect the need for answer to a question about a specific subject (i.e., *How many soccer teams play in the World Cup?*). The objective of such a search is to find a document containing the information required to answer the question. In the e-discovery context, the query concerns the documents themselves—any document could be pertinent if it speaks to the subject at hand. If the subject concerns marketing to children, the legal questions are multiple: *Did Company X discuss marketing tobacco products to minors? What was discussed? Who was involved in the discussions, and when did they occur?* Comprehensive answers to these questions are unlikely to be found in a single document. In e-discovery, the ad hoc retrieval strategy needs to reflect the need to find all relevant documents so that investigators can provide the answers to the broad array of questions that are likely to arise in litigation.

Information retrieval for e-discovery has multiple applications that could utilize relevance feedback strategies. In order to more accurately assess the features of a relevant document, an expert on the topic may be asked to review a sample of documents. The expert feedback may include detailed descriptions of the features that are likely to be present in a relevant document, or a system may attempt to automatically infer these features if the expert only has time to label the documents in the sample as relevant or non-relevant. Another application for e-discovery relates to the review of documents for relevance. Documents identified as potentially relevant require review for actual pertinence and privilege. The ad hoc search has already improved this process by filtering out documents that are estimated to be not-relevant, but a high-recall search is likely to return a large number of documents for review. Relevance feedback data gained

from a preliminary set of judgments by a reviewer may allow for more effective reprioritization of documents for review.

Due to the particular requirements of the e-discovery task, general techniques for improving ad hoc and relevance feedback retrieval results may not be appropriate. For domain specific tasks such as this, it is a common strategy to start with state-of-the-art information retrieval techniques. Researchers then address the specific requirements of the domain by developing or identifying additional methods that add specialized dimensions to the generalized search techniques. For legal discovery, strategies for improving recall while limiting loss of precision complement existing techniques that rank documents on predictions of relevance. Other examples include different scientific domains, where information retrieval systems need to handle the ambiguous names of genes or the complex names of chemical compounds. In these and other cases, domain-specific techniques such as the use of domain thesauri offer improvements over general information retrieval techniques [48].

If we were to consider the e-discovery task from a cost perspective, the cost of failing to retrieve a relevant document is greater in the legal discovery context, whereas the cost for retrieving a document that is not relevant, while not insignificant, is relatively less than it is for general information retrieval under more casual settings. Using these cost penalties, we can identify similar information retrieval goals where failing to identify all relevant documents is costly. In case law or patent searches, failure to locate precedent court decisions or similar patents can prove very expensive. We use the e-discovery task as a model that is potentially applicable to other high-recall tasks.

Within the ad hoc framework, we propose an approach that includes query field selection, mitigation of OCR error, ranking strategies, query language reduction, pseudo-relevance feedback, and merging result sets. Many of these strategies are potentially applicable to general information retrieval tasks, while additional techniques, such as exploiting the retrieval results from negotiated Boolean queries, are particular to the e-

discovery task. For relevance feedback, we propose a model for constructing new result sets from the pools of unjudged documents retrieved and ranked by a selection of previously implemented systems. We use data mining techniques to seek out signals from the collected relevance judgments that allow for effective ranking of the final merged result sets. Finally, we explore the results of our relevance feedback experiments to assess the impact of the number of unjudged documents on the effectiveness of our system, gauge the relative value of our approach to traditional relevance feedback techniques, and to explore further optimizations to these ranking methods. In summary, our goals are to develop models for effective automated "batch" retrieval in the e-discovery environment. We concern ourselves with the characteristics that distinguish e-discovery from other information retrieval problems, including the unique form of the discovery information request and the likely traits of a targeted enterprise document collection potentially containing the desired information. Our retrieval strategies will incorporate a variety of traditional information practices with data mining techniques.

2.4.1 Ad Hoc Strategies

Our strategies for improving retrieval results for the ad hoc task represent a mixture of tactics designed to directly address the challenges inherent in the TREC Legal document set while effectively leveraging all available additional information. We tackle these distinguishing features of the set both directly and by modifying established information retrieval techniques. We explore these strategies in greater detail in the following chapter.

Using the vector space model as our mechanism for retrieval, we look at two different mechanisms for expanding our queries. As a strategy, query expansion—adding new terms to the query vector—will generally improve recall if the added terms match new documents. In particular, we first look at terms in the sample Boolean queries that employ wildcard operators. We find that the extensive presence of noise in the form of

optical character recognition (OCR) generated errors presents performance difficulties if we attempt to include all eligible candidate terms from these operators. In response, we conduct a series of experiments to determine an appropriate amount of wildcard eligible terms to include in our query.

We employ blind relevance feedback as a strategy for finding new but related terms that we can add to our queries. We conduct an initial query with a ranked retrieval, identify a set of terms with high TF-IDF weights amongst a set of the top ranked documents, and add these terms to a second query.

We also find that not all terms contained in the TREC Legal Track topics are useful terms for our queries. We develop and test both automatic and manual procedures for excising a portion of the queries that could be described as standard legal language that do not have any bearing on the actual topic.

Ranking is an important element of the ad hoc task. We replace our baseline TF-IDF ranking method with the Okapi-BM25 term weighting method [71]. This approach has the added benefit of taking into account the length of the document when determining term weights for a document. We also alter our rankings by adjusting a document's ranking score with a "boost" factor if the document meets the criteria established by the sample Boolean queries. In Chapter 4, we begin our exploration of rank aggregation. By using a weighted Borda Count scheme to combine the results from multiple distinct runs, we attempt to bring the best results from each run into a single merged run.

We also report two strategies that did not generate improved results. An attempt to expand our queries to include document metadata beyond the title and text contents did not increase performance in precision or recall-based measures. We also could not identify a link between the amount of OCR error in a document and the probability that the document would be found to be relevant for any topic.

2.4.2 Relevance Feedback Strategies

A simple approach to relevance feedback is to take the pseudo-relevance feedback approach to ad hoc retrieval and retrain the system for handling documents that are actually relevant. Beyond implementing that approach, our intention is to utilize both the judgments and the ranking of the relevant documents in previously submitted runs to merge the unjudged documents into a new run. We use ranking data offered by several information retrieval systems to train a classifier for each topic to estimate the probability of relevance of unjudged documents, which we then use to generate a new relevance feedback-informed ranked set of unjudged documents.

Chapter 5 will be used to explain the details of this strategy, along with the effects of the many parameter and configuration choices that this presents. This method generates significant improvements over popular heuristic rank aggregation methods in several usage scenarios, all of which we evaluate for precision and recall values at low cutoff numbers. We are not able to duplicate this success when the scope is expanded to the scale that would be expected of a full e-discovery ranked retrieval. When we apply the classifier-based ranker against both an heuristic baseline and traditional relevance feedback, we find that rank aggregation improves recall for the entire retrieval, but underperforms when using balanced measures applied at lower thresholds.

In Chapter 6, we expand on these rank aggregation results. The classifier-based ranker is tested against the heuristic baselines using larger inputs, and on additional datasets. We also consider the effectiveness of these methods at ranking. We add rank-order sensitive measures to our analysis, and revisit the results from the full-scale results in order to examine the success of the traditional relevance feedback method over the rank aggregation methods at overall ranking.

Our last experiments reflect an interest in applying social choice theory to rank aggregation methods to optimize ranking results. We find that the success of using a voting model is sensitive to the underlying method that generated the initial merged

ranking. With a limited set of electors (different ranked retrievals for the same topic) but an extensive number of candidates (documents) results in large amounts of documents sharing tied votes. Design decisions related to the handling of these ties have a significant impact when assessing the effectiveness of ranking for the entire merged run.

CHAPTER III

TREC LEGAL TRACK ANALYSIS

3.1 The E-Discovery Information Need

As mentioned earlier, the TREC Legal Track organizers and sponsors are concerned with the specific requirement in most discovery requests that "all" relevant documents to a particular complaint be delivered to the requesting party. This requirement stands in contrast to the more common information need which could be described as a request for a sample of relevant documents. The comprehensive data need for legal discovery reflects both the possibility that pertinent information to a complaint may exist, but also that the pervasiveness of the pertinent information may have an impact in the resolution of the case.

At the point of the creation of the TREC Legal Track, the state-of-the-art information retrieval technique for the task of e-discovery is described as "free text Boolean searching." This would be akin to a standard web search engine, where a single term or phrase is entered into the user interface and documents containing the terms are returned. All documents returned by the system will have to be evaluated by lawyers for "responsiveness" (relevance) to the complaint and "privilege"—correspondence with legal counsel that is protected from disclosure. As new search technologies are adapted for e-discovery, it is expected that the scope of negotiation over searches will move from bargaining over search terms to discussions over proper refining and expanding clauses in extended, more complex Boolean queries, as well as the appropriate use of ranked retrieval and other Information Retrieval approaches.

Research in the 1980s established that, despite lawyers' belief in their search skills, they were not finding all relevant documents within large collections. Of note is a much cited study by Blair and Maron, where an estimated 20 percent of relevant documents for a particular case had been found, despite the confidence of the attorneys

that they had found closer to 75 percent of the documents [10]. The concern that discovery searches could be missing important evidence accounts for the TREC Legal Track's orientation towards recall. Search techniques that produce more relevant documents than the current standard of Boolean keyword searches while expending the same amount of resources for evaluation could be adapted by the legal community as new standard procedure.

3.2 The TREC Legal Ad Hoc Task

A primary problem addressed by the field of information retrieval is the "ad hoc search". The common conception of the problem involves a static collection of documents, and an information need, frequently expressed as a query, where the goal of the system is to locate the documents that best address the need. Most users are conducting an ad hoc search when they use a search engine to find useful web pages or to query a library catalog. An ad hoc retrieval task was present for the first four iterations of the TREC Legal track.

The two key elements of ad hoc search are the query and the indexed collection of documents, sometimes referred to as "the database." Given a query, a system will use a variety of strategies to find documents that are relevant to the user's information need. For ranked retrieval, the documents are presented to the user in order of an estimated probability of relevance. In other words, the first document in the presented list is predicted as most likely to be relevant, the second document is next most likely to be relevant, and so on.

Each participant in the TREC Legal Track is given access to the same collection of documents, which they can organize, store, and index as they see fit. Each team is also given the same set of 40-50 "topics," which contain information that describe the information need. For TREC Legal, the information need is modeled after the formal language used in typical discovery requests, including detailed information about the

nature of the legal complaint that motivates the request. A partial example of this information is presented in Figure 1. For space reasons, we have omitted fields with information not unique to the topic.

A participating team may submit up to eight runs, with each of the runs representing a different information retrieval strategy or system as applied to each of the topics. In order to be evaluated, a run must, for each topic, submit a minimum of one document, and less than a specified maximum number of documents. It will be assumed, for the purposes of evaluation, that the submitted documents for a topic, the result set, are ranked in order of estimated likelihood of relevance. Teams may use outside information sources to supplement the information provided with the topics in order to construct their queries. Finally, teams are asked to specify which runs are "automatic", or implemented without any human intervention or knowledge beyond the specifics of the training data. In other words, a retrieval strategy may involve some manual intervention, but such strategies are identified when results are published.

For the TREC Legal track, the existence of a technology that represents a reasonable state-of-the-art technique allows the organizers to set, if not exactly a baseline, a target that they hope the participants would exceed. The negotiated extended Boolean queries included in the information provided for each topic can be used to return a set of documents, though ranking such a set is not required. Track organizers hoped that some form of ranked retrieval would return more relevant documents in a set of comparable size than that generated using the Boolean query. An example of such a query is the "FinalQuery" XML element in Figure 1. This query is designed to find documents which include references to improved crop yields rising from the use of phosphate-based fertilizers.

In the first two years of the TREC Legal Track, in 2006 and 2007, none of the participants succeeded in surpassing the recall effectiveness of the negotiated Boolean queries. This benchmark was effectively cleared in 2008 by submissions from several

```
<ProductionRequest>
      <RequestNumber>52</RequestNumber>
      <RequestText>Please produce any and all documents that discuss the
      use or introduction of high-phosphate fertilizers (HPF) for the
      specific purpose of boosting crop yield in commercial
      agriculture.</RequestText>
      <BooleanQuery>
            <FinalQuery>(("high-phosphat! fertiliz!" OR hpf) OR
            ((phosphat! OR phosphorus) w/15 (fertiliz! OR soil))) AND
            (boost! OR increas! OR rais! OR augment! OR affect! OR
            effect! OR multipl! OR doubl! OR tripl! OR high! OR greater)
            AND (yield! OR output OR produc! OR crop OR
            crops)</FinalQuery>
            <NegotiationHistory>
                  <ProposalByDefendant>"high-phosphate fertilizer!" AND
                  (boost! w/5 "crop yield") AND (commercial w/5
                  agricultur!)</ProposalByDefendant>
                  <RejoinderByPlaintiff>(phosphat! OR hpf OR phosphorus
                  OR fertiliz!) AND (yield! OR output OR produc! OR crop
                  OR crops)</RejoinderByPlaintiff>
            </NegotiationHistory>
      </BooleanQuery>
      <FinalB>3078</FinalB>
      <RequestSource>2007-A-1</RequestSource>
      <Instruction>
            <P>1.  These requests require the production of all
            responsive documents within the sole or joint possession,
            custody or control of the Defendant, including their agents,
            departments, attorneys, directors, officers, employees,
            consultants, investigators, insurance companies, or other
            persons subject to Defendant's custody or control.</P>
            <P>2.  All documents that respond, in whole or in part, to
            any portion of these Requests must be produced in their
            entirety, including all attachments and enclosures.</P>
            <P>3.  For purposes of these requests, the words used are
            considered to have, or should be understood to have their
            ordinary, everyday meanings.  Plaintiffs refer Defendant to
            any dictionary in the event that Defendant asserts that the
            wording of a request is vague, ambiguous, unintelligible, or
            confusing.</P>
      </Instruction>
…
</ProductionRequest>
```

Figure 1. A sample of the information in a TREC Legal Ad Hoc Topic

teams, including our own. Furthermore, the existence of relevant documents retrieved by participants but not by the Boolean queries support the premise that the baseline Boolean queries are unsuccessful in locating many of the known relevant documents in the dataset. Estimates derived from sampling the result document pools indicate that the average TREC 2007 topic had 16,904 relevant documents, while the average size of the Boolean query retrieval was 5,004 relevant documents, which implies that the Boolean queries are failing to retrieve as many as 70% of relevant document. The presence of relevant documents found in a run of documents randomly retrieved from the collection in the 2007 evaluations for the TREC Legal Track further suggest that the number of relevant documents for most topics exceeds those found by the collective efforts of the participants [89]. These results indicate that there is still further room for improvement. As a participant team, we have contributed runs to the 2007 and 2008 iterations of the *ad hoc* task.

Ad hoc information retrieval can be described as the scenario where the system is aware of the set of available documents, but does not know about the specific resource information needs of the users. This approach is called *ad hoc* due to "the arbitrary subject of the search and its short duration." [96] Early approaches to this problem began with the attempt to create a probabilistic model for indexing, where properties of a document, in the form of explicit metadata, are used to estimate the document's probability of relevance to a given query. Maron and Kuhn discuss the approach of assigning a score for the purpose of ranking documents in order of probability of value to the user, using manually assigned tags as the basis for indexing documents [60]. Robertson discusses a "formalization" of this "probability ranking principle," or PRP, and its implications in [68]. In 1982, Robertson, Maron and Cooper formalize the two predominant probabilistic models and propose a unified variant [69].

Salton compares manual indexing of keywords to retrieval based on automatic content analysis in an overview from 1970, much of which is based on the SMART

retrieval system [75, 76]. Salton further extends his automatic document processing system by developing the vector space model for weighting terms in an index in order to create distinguishing space between documents in a collection [74]. Similarity is usually calculated through a variety of measures, such as cosine similarity, or TF-IDF term weighting, usefully catalogued by Zobel and Moffat [104], themselves extending work by Salton and Buckley [77]. One such formulation, Okapi-BM25, adjusts for collections with variant document length, and has been popularized through succeeding TREC Conferences [71].

The impact of OCR and the associated errors resulting in "confused text" has not been ignored by the information retrieval research community. Taghva, et al., demonstrate that the effect of OCR-generated error on Boolean logic-style information retrieval on their collection of scientific documents is "insignificant." [82] When they conducted similar experiments using the vector space model and relevance feedback, they found that precision and recall were unaffected by OCR error, but that document ranking was effected, and that feedback would not assist in retrieval for "badly degraded documents." [83] OCR error was the focus of a TREC "Confusion" Track, where organizers concluded that probabilistic reconstruction of error-ridden text performed better than ignoring the error, and that high recall applications "like litigation and security" would suffer as confused text would impede sufficient recall [46].

A query, as initially formulated by a user, is often, by itself, not sufficient for finding desired documents. Queries are often augmented with terms from supplementary sources, including thesauri, as well as words and phrases extracted from the document collection using statistical methods and co-occurrence data [75, 81]. Using highly-ranked documents from an initial retrieval as a source for additional query terms, a technique referred to as *blind* or *pseudo-relevance feedback*, is a natural extension to this strategy. The usual design is to use all terms from an initial query and add some number of query expansion terms, where the selection of feedback documents and the criteria for term

selection are both variable and often tuned for a specific information retrieval task [33]. Several formulations for ranking terms for expansion have been developed, with the purpose of estimating the "value" of a term for retrieval [67]. Automatic query expansion encompasses a wide variety of strategies applicable in different environments, and, despite its lengthy presence in the field, is considered appropriate for continued research, with iterative retrieval, random-walk term selection from feedback documents, and contextual modeling amongst many variants for developing the technique [24, 53, 61, 70, 72, 86]. In 2008, TREC created a relevance feedback track to develop a test bed for continued explorations of this technique.

On the flipside of query expansion, the rich language of the TREC Legal Track document requests introduces dimensionality problems which promote the retrieval of non-relevant documents. A systematic means for identifying and reducing unnecessary terms would prove useful for focusing retrieval in the context of enterprise document collections [45].

3.3 The TREC Legal Relevance Feedback Task

In the context of legal discovery, it is reasonable to assume that a follow-up search may become a reasonable technique, especially if information learned from an evaluation of a subset of the initial set of returned documents can be used to refine the search. This approach, known as relevance feedback, is long established as a means for improving the performance of information retrieval systems. Here, an advanced query may be constructed using knowledge derived from the characteristics of documents known to be relevant. An intuitive application of this is to use terms from relevant documents in a new query with the intention of finding similar documents, with the expectation that such documents are more likely to be relevant.

The Relevance Feedback task was introduced in 2007, the second year of the TREC Legal Track. The collection of documents is the same as that used for the ad hoc

task, and the topics consist of a subset of *ad hoc* topics from previous iterations of the track. Additionally, participants have access to all of the relevance judgments from earlier years. Though not frequently used for this type of task, all previous years submissions to TREC are also available to current participants. Teams are allowed to use any of this information in putting together their runs. The Relevance Feedback task uses the same measures for evaluation as the ad hoc task. Measures are calculated only considering "residual relevant" documents. In other words, only those documents that have been found to be relevant solely through judgments generated for the current iteration of the relevance feedback task are considered.

The concept of "relevance feedback" exists in the ad hoc context, where a query may be modified based on information derived from documents collected in an initial retrieval. This strategy is also known as *pseudo-relevance feedback* because user feedback is typically not available, and documents selected for "feedback" are based on heuristics and assumptions. Pseudo-relevance feedback, as an ad hoc strategy, is covered in §4.6. Many of the described strategies are also applicable when the user provides feedback information to the system. In our research, we are interested in feedback in the form of existing relevance judgments, with knowledge of search results generated by using different retrieval systems. Our strategy involves combining unevaluated documents from each of the different result sets into a single ranked list. Bookstein argues for using a sequential history (i.e., known judgments from a ranked list), to make probabilistic decision-theoretical judgments on documents to present to a user for further evaluation [11]. These new evaluations are added to the collection of known judgments, and the new information can be used to modify the probabilistic judgments on remaining documents in the history [21, 49]. We propose a similar "active", or supervised, learning approach, where we select documents from a collection of such "histories". This allows information about a document to be gathered from different perspectives—different strategies for evaluating and ranking documents. A supervised learning approach allows

our system to combine the information from these strategies to predict the relevance, and thus the ranking score, of the remaining documents. As the relevance for a document is established by a human evaluator, this knowledge can be fed into the pool of training instances, which in turn improves the predictions for remaining documents.

This is a data fusion problem, where we can assume that the contributing lists are not disjoint, otherwise known as rank aggregation. Simple approaches include round robin selection, a process that can be improved by learning appropriate weights for assigning contribution proportions [97]. An extensive coverage of different combining methods is included in [29]. A distinction made in rank aggregation research concerns systems which solely utilize the order of the documents in the separate ranking lists as input to the ranking function, while others also make use of any scores applied to the documents in the ranked lists. Some examples of the former include the techniques derived from summing of ranks [78], while the latter could be represented by attempts at modeling the distributions given to different relevance scores used to rank the different runs [57]. A second useful distinction derives from the use of labeled data to train systems with improved accuracy in rank aggregation. Voting systems, such as the combined sums or other models, are typically developed with an unsupervised framework [4]. Recent work has used supervised learning, with labeled data in the form of relevance judgments, to train Markov Chain-based rank aggregators [55].

In order to reflect increasing interest in the ranking of e-discovery search results, the TREC 2010 Legal Track has phased out the ad hoc retrieval task and modified the relevance feedback to emphasize both ranking and relevance probability estimates. Submissions consist of a full ranking of each document in a collection for each topic. Along with a ranking, participants must provide, for each document, an estimate of the probability that it is relevant for each topic. Participants are given hundreds of relevance judgments for each topic that can be used for training. Although some participants employed strategies that are the equivalent of ad hoc strategies and traditional relevance

feedback [91], we note the presence of systems that model the document collection through latent semantic indexing [37]. Finally, one system employs a Naïve Bayes classifier, trained with document features derived directly from the contents of the document to organize the ranking of each system [30].

Similar to this last system, the supervised learning approach employed by our system reflects important research questions beyond the selection of the classifier. In particular, our system does not describe documents using features that reflect the actual document contents. Further, our system applies the classifier to a rank aggregation strategy. To contrast our system with [30], document features for the Naïve Bayes classifier reflect the information that could be retrieved from the index of a collection. The classifier, trained on relevance feedback data, fulfills the ranking function that a similarity measure provides when ordering a set of retrieved documents. Our system ignores indexing data to calculate document features, relying instead on relevance feedback information and rank data from previously developed ad hoc retrieval systems. By framing the relevance feedback task as a rank aggregation problem, we propose that exploiting and combining the results from existing rankings is an effective alternative to building a ranked result set from traditional ad hoc retrieval similarity statistics.

As the e-discovery field of research matures, the associated information retrieval tasks must change to model the increasingly sophisticated workflows developed by organizations to manage the growing expenses and resources demanded by litigation. A multi-tiered process for handling enterprise document collections has developed for application either in response to or anticipation of litigation [26]. The concept of relevance has moved beyond the question of whether an item contains terms that reference the topic of a Request for Production of documents (§2.1.1.3). Users develop a "theory of relevance" that reflect questions such as "who communicated what to whom, when, and, to the extent possible, why." [3]

These developments provide extensive opportunities for research on the applicability of supervised learning. The manual review of a preliminary sample of documents can be used to train an information retrieval system to generate a smaller, but high recall, result set. This process, described as human-assisted computer-assessment, typically "embed[s] human interaction within an iterative IR loop." [44] These systems reflect an approach to e-discovery that is expressed by the TREC Legal Track Interactive Task. This task—which aims to model the conditions faced by organizations as they meet document production obligations—allows the participants to consult directly with an expert for guidance on the characteristics of relevant documents [63]. Our relevance feedback system, which does not use content to characterize documents, would not benefit the input from a topic authority except to use sample relevance assessments as feedback.

Multiple systems have been developed to organize documents into categories, or clusters if no training data is available for this purpose. This organization is used to present users with documents for review. After a minimal amount of documents have been assessed, this data is used to retrain the system—documents are re-categorized and reordered to bring documents most likely to be relevant forward to the user for review, possibly in an iterative procedure [7, 65, 73]. As we note above concerning current TREC Legal Track strategies, these classifiers are used as a mechanism for generating a single ranking using document content, rather than as a tool for implementing rank aggregation. Iterative (or active) learning is a powerful tool that we hope to exploit for future experiments. We also see constructing document features from content and metadata as a potential supplement to the current set of features employed by our system.

3.4 TREC Legal Track Tobacco Master Settlement Dataset

3.4.1 Documents

3.4.1.1 Document Format

The document collection for the TREC Legal Track is a subset of the evidence accumulated as part of the Tobacco Master Settlement Agreement. Each of the documents was determined to be relevant for at least one of the lawsuits filed against the tobacco companies. As a whole, the documents were collected from six tobacco companies and one non-profit industry-sponsored research center/think tank. The documents range from memos and business letters, spreadsheets, scientific articles, and even advertising proof sheets. In order to render the documents available to the public, the documents were warehoused and processed, a process that involved scanning them and the manual entry of document metadata. Optical character recognition was used to convert images of the digitally scanned documents into a text-based format more suitable for automated indexing and retrieval. The scanned images of these documents are now available on the Internet at the Legacy Tobacco Documents Library, hosted by the University of California-San Francisco [52].

In preparation for archival, the scanned documents were processed using optical character recognition (OCR) software. All text generated from the digitizing procedure for a document is stored in a single XML field. The age of the documents dates back several decades to as recent as the filing of the tobacco lawsuits. Furthermore, different origination formats, such as printed emails, magazine clippings, and spreadsheets, also present document quality variance. The condition of the documents prior to digitization varies. It is not clear that consistent standards for scanning procedures were established, and it is possible that different OCR software packages were used to generate the plain text.

Table 1. A sampling of metadata elements in archive file iitcdip.o.x.xml from the Legacy Tobacco Document Collection

Element	Descriptor	Sample Contents
<au>	High Quality – Authors (people)	"CTR"; "JEFCOATE CR, UNIV WI MEDICAL SCHOOL"; "RIEHL, TF", "Guilford Laboratories Inc", "Newton-C B&W"
<ca>	High Quality – Authors (organizations)	"PHILIP MORRIS INC", "INBIFO, INSTITUT FUR BIOLGISCHE FORSCHUNG", "PMMC, PHILIP MORRIS MANAGEMENT CORP", "MUNGER TOLLES", "PMUSA, PHILIP MORRIS USA"
<d>	High Quality – Short document description	"LISTS PAYMENTS AND BALANCES", "AMMENDED PROJECT SUMMARY", "SURVEY ON SMOKING", "PERSONAL, EDUCATIONAL AND PROFESSIONAL EXPERIENCE", "EDUCATION, HONORS AND GRANTS, PUBLICATIONS, AND ABSTRACTS"
<no>	High Quality – Organizational Names Mentioned in Document	"TIME", "BUONICONTI FUND; EDELMAN SPORTS; TOYOTA", "COVINGTON BURLING", "LIG, LIGGETT; MILBERG WEISS; PHILIP MORRIS; RFA; SAG", "F+K; TAB"
<tp>	High Quality – Controlled Vocabulary	"SMOKING BY-PRODUCTS", "BIOLOGICAL ACTIVITY OF CIGARETTES; IN-HOUSE RESEARCH ON SMOKING & HEALTH; SMOKING BY-PRODUCTS", "NICOTINE AND ADDICTION; NICOTINE CONTROL; SMOKING BY-PRODUCTS", "SMOKING BY-PRODUCTS; COMPANY WEBSITES", "SAFER CIGARETTE; PREMIER; SMOKING BY-PRODUCTS"
<DS>	Proximity – Document Source (organization code)	"c", "l", "b", "a", "r" (seven different codes are used in this field)
<dd>	Proximity – Document date	"19930624", "19901113", "19961200/DE", "19791005", "00000000 (was 00000002)"
<ci>	Proximity – Original Case ID	"10004026", "20000699", "20000178"
<cn>	Proximity – Original Case Name	"Oklahoma AG", "Minnesota AG", "MISSOURI AG", "Florida AG", "NORTHWEST LABORERS"
<fn>	Proximity – File Name	"315 – CASTING RESEARCH – 790000", "91284410/91284596/5546-2978; 91284543/91284595/5546-2978;", "2022901554/2022901804/PM 1322 HOUGHTON ET AL 880722 ARTICLE INCORPORATING PAPER-ALUMINUM TUBES, CARBON HEAT SOURCE, AND TOBACCO PELLETS (DE LTA);", "Insertion Orders 1984 Brown and Williamson International Folder #2 Kool Lucky Strike", "2028875208/2028875688/INBIFO MONTHLY REPORTS BGE DISPOSAL SUSPENSION;"

The TREC Legal Track document collection consists of 6.86 million documents stored in 58 Gbytes of files, with each document representing 8.4 Kbytes of text and metadata. The index includes more than 200 million unique terms distributed over the entire set of documents, a result of the OCR error present in the document text. The size of the collection and the noise caused by the optical character recognition are challenges that can be expected to be present in other corporate document collections that include material predating the widespread use of digital documentation.

3.4.1.2 Document Metadata

The text of the documents, as generated by optical character recognition, is made available to TREC Legal Participants in XML format. The XML schema for this dataset allows for the definition and collection of extensive metadata that provides additional context for each document. Along with the document identifier, title, and the raw OCR-generated text, the metadata also contains information about the authorship, distribution, the source organization, and the storage of the document after the case was settled. Table 1 illustrates some of the metadata fields and example contents. Most of the metadata is manually entered, and it is not consistently formatted or even consistently present across the collection.

As Table 1 indicates, the metadata is categorized as “High Quality” and “Proximity” information. The high quality records are entered manually, and are considered to be potentially useful for text retrieval. In the table, the “Document Source” refers to the specific tobacco company or related organization where the document was collected. The “Document Date” is inconsistently entered, but should indicate when the document was created. Each of the documents in the collection were submitted as evidence in one of the cases covered by the Tobacco Master Settlement Agreement, the original litigation providing the basis for the “Original Case ID/Name” fields. The “File

Name" is highly variable, and contains identification or accession numbers, names, and descriptive information.

3.4.2 Topics

3.4.2.1 Complaints and Requests for Documents

Each topic represents a specific request for documents related to a specific complaint. The complaints themselves may be of a large enough scale that several requests are required to cover the scope of the lawsuit. A complaint concerning the marketing of cigarettes to children may have distinct requests for documents pertaining to television marketing, movie marketing, marketing at live performances, and the use of cartoons, to name a few. For the purposes of TREC, information concerning the complaint gives context, and is considered part of the topic. Given the one-to-many relationship between complaints and requests, complaint information is duplicated for as many as a dozen topics. The complaint information itself is written as a detailed description, in the format of court filings, of a fictionalized complaint. This fictionalization aspect refers to non-existent parties, locations (i.e., the planet Vulcan or the State of New Searchland), and organizations (i.e., Smokin' Cigarettes).

The complaint itself often describes the lawsuit parties, the nature of the legal claim, the jurisdiction and venue of the case, the substantive allegations, additional allegations, claims for relief, extensive background instructions for production of documents (discovery), and the Boolean queries to be used in searching for said documents. The complaint itself can consist of several thousand words.

3.4.2.2 Presentation

TREC provides the topics to participants in XML files. The topic file, a sample of which is presented in Figure 1, duplicates complaint information for each of the related topics. Information that is unique for each topic includes the specific narrative request

for production of documents, along with a set of Boolean queries that represent a negotiation history between the involved parties. One of the Boolean queries is designated as the "Final" result of the negotiation. The track organizers execute this query against their own index, and provide an unranked list of the documents that are retrieved to the participants in a separate file, which can be used as seen fit. The count of documents found by this baseline, also called the "Boolean reference run," is also provided as part of the topic information.

3.4.2.3 The Negotiated Boolean Queries

The negotiated Boolean queries are constructed through a series of modifications. The "plaintiff" proposes a query designed to cast a broad search, and then the "defendant" offers a rejoinder that narrows the search. For 2008, this process may iterate two or three more times. The "Final" query represents a negotiated "middle ground," though there is no reason that it may not more closely resemble the position of either party. For the purposes of TREC, the final negotiated query is biased to retrieve a number of documents that fall within a specified range, i.e. it must retrieve more than 100 documents, and less than 100,000. An example of the query negotiation history is included as part of the BooleanQuery element in the sample topic presented in Figure 1. We can trace the negotiation in the example. Looking at one aspect of the topic, the plaintiff, motivated to retrieve as many documents that may be possibly be relevant to their case, requests all documents containing the prefix "phosphat!", "fertiliz!", the word "phosphorus", or the acronym "HPF." The defendant, presumably a tobacco company, prefers to limit the search solely to documents containing the phrase "high phosphate fertilizer", which would, for instance, remove documents from the pool of potential evidence that only use the acronym to refer to the substance in question. The negotiated solution narrows the plaintiff's request by linking the "phosphat!" prefix with the "high" and the second prefix in the phrase "fertiliz!", allowing the acronym as an alternative, and

also allows inclusion of documents concerning “phosphorus” if they are mentioned in proximity in the document with fertilization or soil.

The Legal Track organizers execute the Final query against the dataset. The results from this query serve as a baseline that is considered comparable to current practices. The number of documents retrieved by the Final query is included in the topic file, and the document identifiers are made available to participants in a separate file. TREC Participants are encouraged to use information from the Final query to improve the performance of their own systems.

3.4.2.4 Training Data

The track organizers provided an initial set of topics and relevance judgments for the first year of the track. For subsequent years, the topics and judgments from previous years are made available for training. Our participation in the TREC Legal Track began with the 2007 iterations, so our system has always benefited from at least a year's participation data for tuning our strategies.

3.4.3 Baseline Results

The “Final” negotiated Boolean query represents the equivalent the approximate state-of-the-art, a retrieval strategy that resembles the current practice for electronic legal discovery. The Boolean query can be considered a filter, in which documents either fall within the constraints defined by the query and are included in the result set, or they break one or more of the constraints and are thus rejected. Without modification, this framework provides no basis for estimating a retrieved documents probability of relevance. Thus, the set of retrieved documents are not ranked. In order to improve on the performance of the baseline, a participating system would have to retrieve more relevant documents than the baseline system while retrieving the same number or even fewer documents from the collection.

Comparing participant runs against the baseline allowed the track organizers to verify that the baseline was failing to retrieve relevant documents. In 2008, several participant runs succeeded in exceeding the recall results of the baseline run, though the organizers took the opportunity to note that the instructions to topic creators were modified to allow a broader range of relevant documents.

Both the count and identifiers of the documents retrieved by the baseline are provided to participants in the TREC Legal Track. Organizers encourage the participants to use this information to enhance their own systems.

CHAPTER IV

TREC LEGAL – AD HOC RETRIEVAL

In this chapter, we discuss the details of our solutions for the TREC Legal ad hoc retrieval task. We present the results of our experiments for addressing different dimensions of the problem. These include strategies for handling queries with "wildcards," modifying and truncating the natural language queries, leveraging the results generated from the negotiated Boolean queries, exploiting document metadata, and applying different models for pseudo-relevance feedback. We explore the possible connection between the amount of OCR error found in individual documents and the likelihood that a document will be found to be relevant. Finally, we consider the benefits of combining the results generated from different selections of strategies.

4.1 The inputs

As described in the Chapter 2, the given inputs for the TREC Legal ad hoc retrieval task consist of a set of topics and the dataset of documents. Track participants are allowed to organize the dataset as they see fit, and, likewise, they may supplement the topic information through external sources. The dataset is provided in 650 XML files, each roughly 75-100 Mb in size.

We use the Java-based Lucene indexing libraries to create our index [56]. The document identifier, title, and OCR-processed text are extracted to create a Lucene document. The text is processed using the Lucene StandardAnalyzer, which leaves intact most terms containing punctuation and uses a minimal "stop list" to filter common terms from the index.

Our standard query approach is to construct a query term vector from the information provided in the topics. Terms are selected in a pre-processing step for inclusion in a query. The query itself is constructed as a series of Boolean OR clauses,

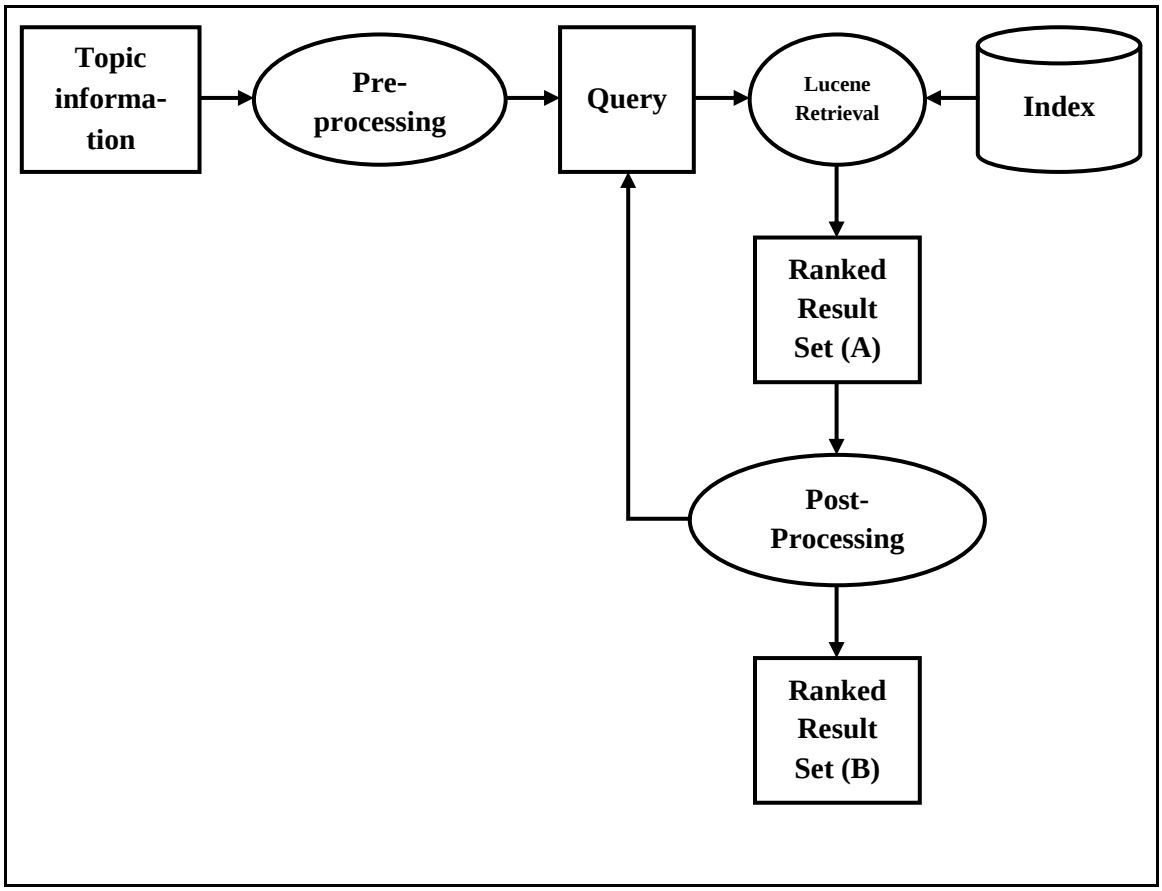

Figure 2. A high-level system diagram for our retrieval system.

with each included term represented as a single clause. Lucene returns a set of documents containing at least one of the terms from the query vector, ranked according to their similarity to the query. A post-processing step is used to rerank the result set using a more sophisticated similarity function, Okapi-BM25, which, among other aspects, accounts for the varied lengths of the documents in the collection. A high-level overview of our system is described in Figure 2.

4.2 Indexing the TREC Legal Document Collection

A collection containing approximately 7 million documents does not, by itself, present an interesting challenge for research. But, the TREC Legal dataset contains

documents in a wide variety of information formats and the text of the documents is comprised of a "vocabulary" that includes all of the term varieties generated by OCR error. Documents range from memos or advertisement proofs with few words, to procedural manuals with hundreds of pages of text. The OCR error additionally gives each "actual" word in the text several potential misspellings. This latter effect also presents challenges when handling query terms where only some of the characters are specified (i.e., "wildcards"), and both challenges create word count distortions that interfere with the statistical language models that are frequently the basis for information retrieval.

4.3 Wildcard Expansion

For the TREC Legal Track ad hoc task, each topic includes a "FinalQuery" constructed in an extended Boolean query format. This query is the result of a negotiation between opposing lawyers (see Figure 1). 43 out the 50 "FinalQueries" in the 2007 Legal Track included wildcard operators. Designed to find words with similar roots as the terms in the query vector a term will match a term with a wildcard operator if it has the same letters in all locations as the wildcard term, except for zero or more characters at the location of the operator. For instance, the FinalQuery for topic 53 includes the term "rodent!" Any document containing any term beginning with "rodent"—such as "rodents", "rodentlike", "rodentworld"—will qualify as a match so long as it doesn't violate any of the other Boolean constraints. The appeal of wildcards in a manual search is that it allows the user to avoid adding all possible variants of an important term to the query. This approach is also relevant to automated search, and in the TREC collection it allows the system to find instances of a term where, due to OCR error, words are mistakenly compounded or corrupted. For instance, the wildcard term "mistake!" would match both "mistakenlycompounded" and "mistakeh".

Table 2. R-Prec and MAP scores, along with sample candidate replacements for different numbers (*n*) of replacements.

Wildcard Replacements (*n*)	**0**	**1**	**2**	**3**
R-Prec	0.0785	0.0893	0.0968	0.0916
% improvement from *n* = 0		13.76	23.31	16.69
MAP	0.0430	0.0462	0.0554	0.0522
% improvement from *n* = 0		7.44	28.84	21.40
Wildcard Term (examples)	**Candidate Terms**			
calif!	California	Calif	califano	californians
ad!	Ad	additional	advertising	addition
cigar!	Cigar	cigarette	cigarettes	cigar

The practical effect of the wildcard operator on the query is to add every qualifying term in the document collection to the query term vector. When testing our index using the TREC Legal Track training topics, we found that one of the supplied Boolean queries included a wildcard term which would necessitate the addition of every term in the corpus beginning with the characters "ad" to our query term vector. The large number of terms in the English language that meet this criteria would be enough to push the memory limits of our system, a problem compounded by the number of variants of these words to be found in the collection introduced by OCR Error.

Our first strategy was to cap the number of accepted expansions for any wildcard in the topic Boolean queries. Using MAP and R-Prec as evaluation measures, we assessed our retrieval system using a decreasing number of eligible wildcard expansion terms. Our system, which constructs each query without any prior knowledge of the document or the index, does not assign prior weights to any of the terms in the query term vector. Therefore, when a wildcard adds a large number of terms to the query, it reduces

the influence of the other terms in the query when scoring the matching documents. Our solution, then, requires selection of an arbitrary number of eligible expansion terms with the greatest document frequencies, i.e., they appear in the greatest number of documents throughout the corpus.

In Table 2, we have indicated the performance in MAP and R-Prec for different *n* quantities of accepted wildcard candidates. In this case, the improved performance when $n = 2$ confirms results from a smaller training dataset, where scores for both measures improved as we reduced *n* from 10 to 2 and then declined as we eliminated the two remaining wildcard terms.

Given these results, unless otherwise specified, all subsequent runs of the system replace each term in the Boolean queries containing a wildcard operator with two evenly-weighted matching replacements. Given the level of noise in the documents in the TREC Legal dataset, terms that appear infrequently are most likely due to OCR error. We then assume that terms with the greatest document frequency are most likely to have been desired by the human composer of the Boolean query.

4.4 Ranking the Result Sets

When retrieving documents, Lucene uses a TF-IDF similarity measure to rank matching documents by comparing them to the query term vector. We found that we would garner improvements in MAP and R-Prec by identifying the top $1.5m$ documents using Lucene, where *m* is the target size for our result set, and then reranking the documents using an Okapi-BM25 scoring formula. The advantage of the Okapi-BM25 similarity measures is that it additionally considers the length of the average document in the collection when normalizing for document length.

Our Okapi-BM25 score is determined using the following formula for calculating the Score for Document *D* given term vector query *Q*:

$$Score(D,Q) = \sum_{i=1}^{N} \left(\frac{f(q_i, D)(k_1 + 1)}{f(q_i, D) + k_1 \left(1 - b + \frac{b|D|}{\text{avg}|D|}\right)} \right) \left(\log\left(\frac{C - d(q_i) + 0.5}{d(q_i) + 0.5} \right) \right) \quad (8)$$

where N is the number of terms in the query, q_i is the ith term in the query vector, $|D|$ is the number of terms in the document, and avg$|D|$ is the mean of $|D|$ over the entire document collection. The function $f(q_i, D)$ returns the frequency of term q_i in document D. In a typical TF-IDF formula, raw term frequency relative to a document's length may distort the influence of a term that appears frequently in a small document, especially if it does not appear in many other documents. The first component of the Okapi-BM25 score flattens, but does not eliminate, the impact of these high-frequency terms. Instead, it reduces the impact of low-frequency terms as the length of the document increases beyond the average document length in the collection.

In the second component of the score, C is the number of documents in the collection, and the function $d(q_i)$ is the document frequency of term q_i, or the number of documents in the collection containing the specified term. This second component gives a greater weight to terms in the query that appear in fewer documents in the collection, much like the IDF component of the TF-IDF score.

Formula 1 is a simplified variant of the full Okapi-BM25 measure, which also takes into account the presence of relevance feedback, and where the weight is reduced for repeated terms in the query vector [71]. Our implementation uses parameters $k_1 = 1.2$, and $b = 0.75$. These values are typically identified as "default" settings in various non-Lucene implementations of BM25 formula [54, 85]. Each repeated instance of a term in our query vector was treated as a unique term, which is consistent with setting the k_3 parameter to ∞, thus allowing us to remove it from our formulation.

4.5 Metadata experiments

Metadata is information used to describe a unit of information--in this case, a document. Each document in the TREC Legal Track dataset has several fields of associated metadata, as previously discussed in §3.4.1.2. We note that the OCR-generated text itself is included as part of the metadata of the document, as it is largely an (automated) attempt to describe the words embedded within an image of the actual document.

Direct queries using the terms from the supplied Boolean queries generated poor results when applied to the document-level metadata included in the collection, with the exception of the extracted OCR text. The most promising result was found in the "title" field of the document. The final index used in our experiments includes the text from the "title" and the "OCR-text" fields combined into one single field representing the contents of the document. It is against this field that all queries are executed.

4.6 Pseudo-Relevance Feedback

Blind feedback, or pseudo-relevance feedback, had been established as an effective technique for retrieval and ranking tasks. As a technique, it has been investigated for decades, with the "Rocchio algorithm" used as a standard approach for incorporating feedback information. In the vector-space model of information retrieval, documents and queries are modeled as vectors of terms. The Rocchio algorithm provides a mechanism for modifying an original query vector with information from known relevant and non-relevant documents. Terms from both types of documents are added to the term vector, with terms from non-relevant documents given negative weights [58].

The implementation of our system for the TREC Legal ad-hoc task involves a preliminary run to retrieve an initial set of top-ranked documents. From this batch, terms are extracted for query expansion from a few of the top-ranked documents. The queries are re-executed, using the exact configuration as for the preliminary run, except that

candidate query expansion terms have been added to the query term vector. Rocchio's algorithm is general enough that it encompasses our system [72]. Documents selected from the top-ranks of the initial query result are assumed to be relevant. Terms are selected from these documents using a series of filters and formulae and added to the original query vector. The end result is a new query term vector based on the original but modified with terms from "relevant" documents.

Our approach to blind feedback currently involves a single iteration of the expansion strategy described above. The number of top-ranked documents analyzed (*N*), and the number of terms extracted (*T*) are both parameters that can adjusted. To qualify for inclusion in the query term vector, a term has to consist entirely of letter characters, it cannot consist entirely of consonants, it cannot already exist in the query vector, and it has to meet a minimum document frequency threshold. This threshold, set to 1 + *N*, forces the system to add terms that occur in documents besides the set of top-ranked documents. A term was also passed over for inclusion if WordNet failed to recognize it as a word, or if it was included in the list of commonly used "stop words" employed by the SMART retrieval system [76], a much more extensive list than the one included with Lucene.

After identifying the set of terms that meet the filtering criteria described above, we select terms for inclusion in the expanded query vector by taking the top *T* ranked terms as determined by the following scoring formula:

$$score(t, \mathbf{D}) = \frac{d(t, \mathbf{D}) \log f\ (t, \mathbf{D})}{n \log d\ (t, |\mathbf{C}|)} \tag{9}$$

where $d(t, \mathrm{D})$ is the number of documents within the set of documents D that contain term *t*, and $f(t, \mathrm{D})$ is the number of times that the term appears within the set of documents D. Here, D is the set of *N* top-ranked documents from the initial query, and

|C| is set of documents representing the entire document collection. Once a set of qualifying expansion terms have been selected, they are added to the original term vector, and the query is executed once again.

The effect is to score terms according to their presence in the top-ranked documents relative to their distribution across the complete dataset. A high term-frequency in the top documents is beneficial if the term is also present in more of the documents. The practice of limiting terms from relevant documents to those with a high "local" term frequency has been explored previously with differing conclusions [16, 40, 58].

Further explorations with blind feedback can be conducted by using iterative expansions, making adjustments to the query term vector each time until the set of returned documents stabilizes. We can alter our query term vector to use a more sophisticated term weighting framework. One further aspect for future research would be to leverage the documents retrieved by the negotiated Boolean queries by privileging them when ranking our initial retrieval results. Thus, the human intelligence represented by these queries would influence our blind feedback expansions by giving them preference as a source of new query terms.

As part of our official submissions to the TREC 2007 Legal Track ad hoc task, we created experimental runs designed to test our blind feedback query expansion strategy. We present the results from this test in Table 3. We created one run ("Base") that implemented the strategy of incorporating terms from the Request Text and the available Boolean queries, with wildcards expanded to two candidate terms. Our test run included blind feedback query expansion ("QE"), with five terms from the top three documents of a preliminary query. For this experiment, the "Base" run provided the preliminary query and results. In addition to R-Prec (Precision at the number of relevant documents) and Mean Average Precision, we include the recall measure Est_RB. This value represents the proportion of the estimated number of relevant documents in the collection that are

Table 3. Results for runs submitted to the 2007 TREC Legal Track ad hoc task to test blind feedback query expansion.

RUN	Base	QE
Est_RB	0.1639	0.1669†
R-prec	0.1806	0.1733
MAP	0.1304	0.1325

† indicates a significant improvement, $\alpha \leq 0.05$

estimated to be ranked above the threshold indicated by the number of documents returned by the baseline Boolean query.

The results from these experiments indicate that the query expansion generated a significant improvement in estimated recall @ *B*, where *B* is the number of documents returned by the "Final" negotiated Boolean query. Remember that the terms added to the query as part of blind feedback expansion are from documents that are retrieved by the "Base" query. But because these terms are not actually in the "Base" query, it has the effect of creating a broader search that includes terms that are presumably related. This broader search increases the chances of finding relevant documents in the collection. But such a change would not necessarily have an effect on precision, as reflected by the lack of significant movement in the precision-based measures between the two runs.

4.7 Query Term Reduction

The Request Text is a field that is constructed for each topic. It is presented as a narrative request for documents, a natural-language expression of the information need using established vocabulary appropriate for the setting of a civil proceeding, often in natural language sentences. In the sample topic presented in §3.2, the Request Text

Table 4. A sample RequestText field and the modified versions used in our training

Full RequestText	All documents discussing, referencing, or relating to the doctrine of "market share liability" which also relate to one or more events taking place in the State of California.
Manually edited RequestText	doctrine of market share liability which also relate to one more events taking place in the State of California.
Automatically reduced RequestText	doctrine market share liability which also relate one more events taking place state California

reads: “Please produce any and all documents that discuss the use or introduction of high-phosphate fertilizers (HPF) for the specific purpose of boosting crop yield in commercial agriculture.” These requests include terms that refer to the search itself instead of the specific information need. Incorporating terms such as “please”, “produce”, “documents”, or “discuss” in the query term vector can increase the potential for non-relevant documents to be included in the ranked result set.

It appears that there is “boilerplate” language in the Request Text that is repeated for all of the topics under the umbrella of a particular complaint. In order to explore the effects that this language has on the query, we compare the query with a version where this language is eliminated. We use two different techniques to remove the language from the field. Our automatic system (AutoReduce) deletes any terms that appear in the Request Text for an excess of topics over the entire set of topics in a run. We also conduct a manual edit (ManualReduce) that involves removing the terms that appear to be boiler plate from the request text until the first topic-specific term is encountered (Table 4). Note that both of these methods may still leave terms in the query term vector that may not apply directly to the topic (i.e., “specific” or “introduction”). The presence of these terms suggests that this rough strategy can be further refined to reduce the impact

Table 5. Results for runs submitted to the 2007 TREC Legal Track ad hoc task to test manual and automatic query term reduction.

RUN	QE	AutoReduce	AutoReduce2	ManualReduce2
Est_RB	0.1669	0.1613	0.1531	0.1419†
R-prec	0.1733	0.1842‡	0.1761	0.1731
MAP	0.1325	0.1394	0.1288	0.1200††

† = significant decline, $\alpha \leq 0.05$

†† = significant decline, $\alpha \leq 0.0005$

‡ = significant improvement, $\alpha \leq 0.05$

of terms that describe the nature of the request for production rather than the information need.

Experimental runs designed to test the query term reduction strategies were part of our official submissions to TREC in 2007 (Table 5). In Table 5, we first compare the result of applying the AutoReduce procedure to a baseline procedure, and we then compare the results from automatic and manual reduction of the terms in the RequestText. "QE" is the run with the same label in Table 3—the query includes terms from the pseudo-relevance feedback procedures. "AutoReduce" modifies "QE" by automatically reducing the terms from the RequestText field that are added to the query term vector. "AutoReduce2" uses only the FinalQuery from the set of Boolean queries available to the topic (see §3.4.2.3) when building the query term vector, so it is a separate baseline from which we compare the results of manually reducing the terms from the RequestText field ("ManualReduce").

The results indicate that this automatic reduction improves the precision of our system, but has the potential to hurt system recall. Interestingly, by manually editing the

request text to remove presumed irrelevant terms, we generated significantly worse results in both estimated recall and average precision when compared to the automated truncation strategy. One should note that the improvements in precision induced by reducing the request text appear to be reversed in the most recent results from 2008. This may be partially explained by the design of the topics in 2008 to reflect larger run sizes, larger return sets from the Boolean FinalQuery, and a larger pool of relevant documents.

4.8 Boolean Query Boosting

As presented in Figure 1, the TREC Legal Track topics include a series of Boolean queries that represent the stages of a negotiation by the opposing lawyers. The end result of this negotiation is the "Final" Boolean query for the topic. In both the 2006 and 2007 TREC Legal Tracks, the baseline runs generated by implementing the "Final" Boolean queries had proven to be at least as effective at finding relevant documents from the dataset as any of the ranked retrieval runs. These baseline runs are conducted prior to release of the topics to the track participants. Thus, the identifiers for the documents found by the baseline system were provided to the participants with the topics. It is reasonable to assume that a document in a participant result set is more likely to be relevant if it is also included in the results retrieved by the baseline run. This suggests the strategy of identifying the intersection of documents that are contained in the results for both the baseline and our ranked retrieval strategies. Once those documents are identified, an adjustment is made to the ranking score, allowing us to privilege the documents from the baseline results in our own rankings.

We implemented this strategy by multiplying a "boost-factor" (x) to the Okapi-BM25 score prior to ranking. If a document in our return set is also included in the documents retrieved by the "Final" negotiated query, we multiplied the original Okapi-BM25 score by x. We tested values for x ranging from 0.75 to 2.0, using a three-fold cross-validation on the TREC 2007 data. Note that for $x < 1$, this strategy actually

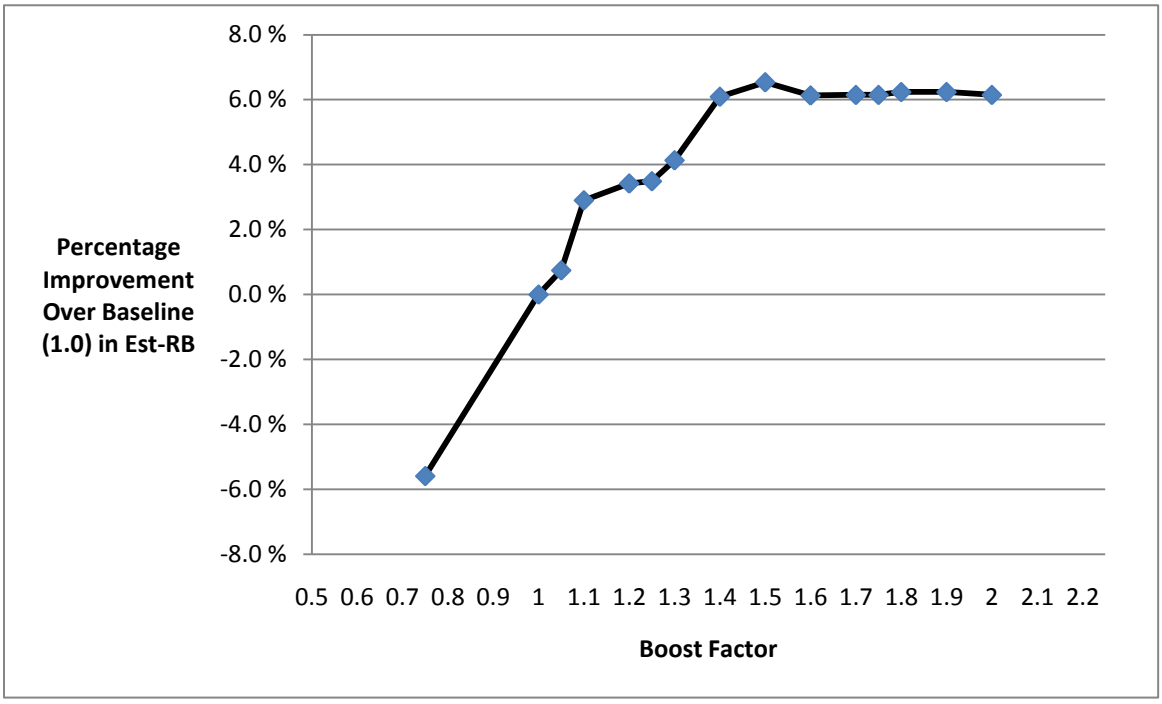

Figure 3. Effects, measured in Estimated Recall at *B*, of applying the boost factor to the ranked results from a run using both query expansion and RequestText Reduction strategies.

penalizes documents if they were retrieved by the final Boolean query; we wish to test the opposite hypothesis, that matching the Boolean query is more likely to predict non-relevance. We applied the boosting strategy to a result set that was initially retrieved using query expansion and automated truncation of the RequestText. Documents in the result set are then re-ranked after applying the "boost factor." The distribution of Okapi-BM25 scores will affect the distance in the rankings reflected in the shift of a document that had received the boost.

We tested this strategy using a three-fold cross-validation. We found this to be an effective technique when added to our query expansion strategies with performance measured by the Estimated Recall @ B measure, as presented in Figure 3. We also found that a different boost could be more effective for different implementations of the

Table 6. Results for runs submitted to the 2008 TREC Legal Track ad hoc task to test the Boost Factor strategy.

RUN	QE	QE+Boost	QE+AR	QE+AR+Boost
Est_F1R	0.2081	0.2210	0.2187	0.2275
Est_RB	0.2921	0.2893	0.2938	0.2937
MAP	0.1426	0.1500	0.1379	0.1495†

† = significant improvement, $\alpha \leq 0.05$

strategies, and when training for different measures. For the example in Figure 3, note that applying a "penalty" factor of 0.75 resulted in a significant ($\alpha \leq 0.005$) reduction in performance.

As a further test of this strategy, we submitted two runs to the TREC Legal ad hoc task in 2008 using query expansion baselines and two runs using the Boolean query boost factor, as detailed in Table 6. The first baseline, labeled "QE" consisted of a run constructed using the query expansion strategies described in §4.6. We then applied a "boost factor" of 1.8—the value with the highest average score in Estimated Recall @ B in our training on 2007 TREC Legal dataset. Our second baseline, labeled "QE+AR" included both the query expansion strategy, and the AutoReduce request text reduction described in the previous section. The 1.5 "boost factor" applied was also found through initial training on the 2007 dataset. In particular, we note the significant 8.41% improvement in Mean Average Precision when the boost is applied to runs using both the query expansion and the automatic query reduction strategies.

4.9 OCR Error and Retrieval

OCR error in the raw text is a potential source of complication for text retrieval strategies. For an information retrieval task where it is only necessary to identify a sample of relevant documents, the potential for reduced performance due to a relevant document being missed because of OCR error is mitigated by the strong likelihood that another relevant document in the corpus will still contain the necessary matching terms. Further, any key term in a relevant document should appear frequently enough that the document itself will still be retrieved for the query [82]. But the impact of the OCR error in the collection cannot be ignored. It explodes the number of unique terms in the collection and distorts the statistics that are used for ranking the results and, in our system, for selecting terms for use in query expansion as well.

With this observation in mind, we consider the effect of OCR error on retrieval. Our initial analysis of this problem was prompted by the difficulties with full wildcard expansions (see §4.3). Our intent was to build a document profile by considering the document frequencies of its component terms. We found that earlier heuristics, such as those suggesting that terms appearing in less than 1% of the documents are likely to be the result of OCR error, were not accurate for the TREC Legal Collection, especially when these terms were checked against WordNet for validity [35]. For the TREC Legal collection, we observe that a term must appear in less than 0.1% of the documents before we label it as an invalid term derived from OCR error. We then considered the notion of "distinctiveness." We sought to determine whether the amount of OCR error in a given document would have a link to the likelihood that the document would be relevant for any query.

We use "distinctiveness" to describe a range of document frequencies. Specifically, we use document frequency to characterize terms into five levels of distinctiveness. With these categorizations, we can build a profile for each document. In Table 7, we enumerate the term distinctiveness levels, L1 to L5, by the document

Table 7. Term Distinctiveness Levels

	L1	L2	L3	L4	L5
Minimum Document Frequency	6,864	687	70	8	1
Maximum Document Frequency	68,629	6,863	686	69	7
Percentage of valid terms	71.50	36.58	10.33	1.70	0.10
Percentage of terms in judged docs	19.20	8.13	4.74	3.48	5.28

frequency range for each level. Distinctiveness Levels are defined as follows: L1 includes terms that are present in 0.01 – 0.1% of the document collection, L2 includes terms that are present in 0.001 – 0.01 % of the document collection, and so on. At L5, we reach the smallest scale for processing these terms, 0.000001 – 0.00001 % of the document collection. L5 terms are the most distinctive. Distinctiveness Level 0 includes all terms that appear in more than 0.1% of the documents in the corpus. In this table, a check against WordNet is performed to determine "validity," so this analysis is subject to the caveat that many proper nouns are possibly labeled incorrectly. We limited this analysis to documents that were judged for relevance as part of the TREC 2007 Legal Track evaluation. We calculated the percentage of terms at each distinctiveness level for each of these documents. We then averaged these percentages across all of the judged documents.

The results indicated in Table 7 are not surprising. The percentage of terms represented by a given level of distinctiveness decreases as distinctiveness increases. This picture is complemented by the percentage of "valid" terms. i.e., terms that are found in the WordNet database. As distinctiveness increases, the likelihood that a term is valid declines dramatically; reaching 0.1% probability that the term is valid when it

Table 8. The percentage of terms for each distinctiveness level, for relevant and non relevant documents

Document type	**L0**	**L1**	**L2**	**L3**	**L4**	**L5**
Relevant	56.64	20.86	8.67	5.10	3.74	4.99
Non-relevant	59.94	18.79	7.97	4.62	3.39	5.29

appears in seven or fewer documents in the collection. We do note the increase in the presence of L5 terms in these documents. The "invalid" profile of those terms, suggests that OCR error is present in the documents that are selected by the TREC Legal participant strategies. In particular, the fact that 40.8% of the terms in judged documents appear in 0.1% of the documents of the document collection indicates that the OCR error is contributing to a large vocabulary of distinct, but not necessarily meaningful, terms in the collection. The profile of documents that are not retrieved by any of the systems is not studied; it is possible that the distinctiveness profiles on non-retrieved documents may be considerably different than those that are selected.

One potentially powerful use for this data was to determine whether relevant or non-relevant document have different term distinctiveness profiles. However, using 2007 TREC Legal relevance judgments, we found that the profile of the average relevant document was barely distinguishable from the profile of a non-relevant document. In Table 8, we show that the average proportion of terms at each distinctiveness level is similar—for example, 21% as opposed to 19% of L1 terms in relevant and non-relevant documents, respectively—whether the document has been judged as relevant or non-relevant. Thus, term distinctiveness has not proven to be a useful predictor of document relevance.

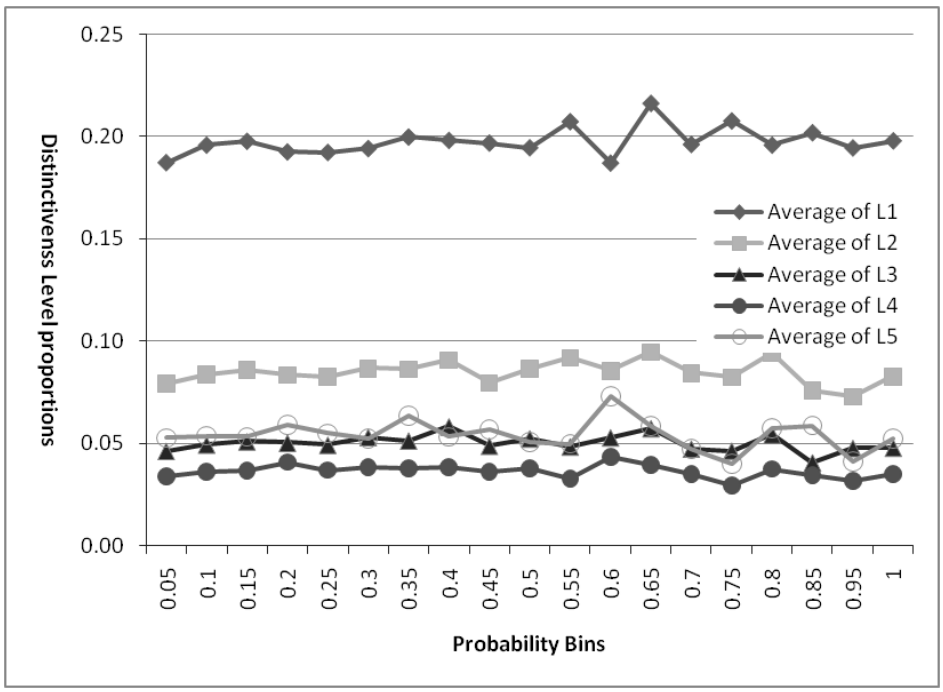

Figure 4. Proportion of term distinctiveness levels
by probability of selection for evaluation

It became clear that the term distinctiveness profile of a document also had little bearing on its likelihood of being selected for evaluation. TREC 2007 used a variable sampling technique to determine a document's probability of selection for evaluation for a given topic. This probability is tied to the document's highest ranking in a TREC submission for the topic. Probability of selection is therefore a measure of the depth of a document in the run assigning it the best probability of being relevant. Figure 4 plots the average proportion of terms in each distinctiveness level across all documents against the probability of selection for evaluation. The documents are grouped into "bins" representing a range of 0.05 in probability. The bin label indicates the ceiling of probability of selection for documents included in the bin. The charted proportion for a distinctiveness level is averaged across all documents in the bin. We should note that none of the judged documents were assigned a probability of selection between 0.85 and

0.90. It is apparent that the distinctiveness levels would be poor predictors of the best ranking a document would have achieved across the runs.

One can use the distinctiveness levels described above to draw conclusions about a document collection. In the TREC Legal document collection, we have observed that judged documents generally have more terms that appear in fewer than seven documents than they have terms that appear in more than eight and less than 686 documents. Since a term in the most distinct category in this collection will have a 1 in 1000 chance of being "valid," we can conclude that many of these terms are generated as the result of OCR error. Documents in this collection that aren't selected by any term-matching information retrieval system may have a greater amount of error; they would potentially have even a greater proportion of L5 terms. Documents in a collection that are manually entered—typed instead of scanned—may have less error, and thus fewer highly distinctive terms in their documents.

We also re-ranked one of our TREC submissions using several distinctiveness-based criteria, with the goal of creating a new result set with more relevant documents shifted to the head of the ranking. Re-ranking the run according to proportion of terms at a particular distinctiveness level generates poor results. Sorting the documents in their bins according to their Okapi scores and then within the bin by term distinctiveness also failed to improve the system scores. Finally, multiplying the Okapi score by the proportion of terms from a distinctiveness level also failed to generate significant improvements over all topics, though individual techniques did work for particular topics. In particular, Table 9 notes the results of runs generated by calculating a ranking score by the inverse proportion of terms in a given distinctiveness level. In that table, we use L*N* to indicate the percentage of terms in the document with distinctiveness level *N*. When compared against a baseline ranking using the Okapi-BM25 score, this strategy failed to generate an improvement in any of the measures, except for an improvement in Recall at

Table 9. TREC measures for runs ranked using scores including a distinctiveness-based factor.

Ranking Formula	Est-RB	MAP	R-Prec
BM25	0.1145	0.1614	0.2139
BM25 × (1-L1)	0.1390	0.1399	0.1930
BM25 × (1-L2)	0.1138	0.1596	0.2063
BM25 × (1-L3)	0.1143	0.1573	0.1980

B when using a ranking score calculated by multiplying the original Okapi-BM25 score by (1 - L1). We concluded that this improvement was not significant as it was limited to extreme improvements for only a few topics. Thus, at this point, we have not been successful at exploiting the proportion of OCR errors in order to improve retrieval performance. However, this strategy may be useful for 'topic' tailored retrieval.

4.10 Combining Ranked Results

TREC has a long history of combining the ranked documents from multiple sets of contributing runs [78]. One simple approach for this is to rank the documents by a score that is the sum of the inverted ranks across runs. Within this approach, it is also possible to weight the contributions of different runs to the summed ranking score [4]. We submitted two result sets, each employing the weighted variant of the Borda Count to the TREC Legal 2008 ad hoc task.

The first of these two "merged" runs incorporated results from two contributing runs, weighting their contributions to the new ranking score by the runs' respective performance against topics from the previous year. The contributing runs implement our Boolean query boost (§4.8) and query expansion strategies (§4.6), with the difference

between the runs present in the use of the automatic query reduction strategy (§4.7). The two runs are merged by calculating a new ranking score for each document by counting the number of documents with a lower rank across all runs. This merge strategy is known as Borda Count [12, 66, 94], and the ranking score for each document *i* is calculated according to the following formula:

$$score(i) = \sum_{k=1}^{l} \#\{j | i >_{\tau_k} j\} \quad (10)$$

where $\tau_1 \cdots \tau_l$ is the set contributing rankings, and $i >_{\tau_k} j$ indicates that document *i* has a higher ranking than document *j* in the ranked list τ_k [55].

The second merged run considers rankings from three sets of results. The new, additional contributing run used all the strategies mentioned above, but it also includes terms extracted from all of the Boolean queries representing an extended negotiation. Because these queries are only available in 2008 TREC Legal topics, it was not possible to train this merge against topics from earlier years. In the absence of training, weights were assigned arbitrarily.

Results for these merged runs are included in Table 10. Contributing runs are given identifying numbers, {4, 5, 8}, for tracking the contents of the merged runs. "QE" and "AR" signify the use of Query Expansion and AutoReduce strategies, respectively (see Table 6). "EB" signifies that the query term vector includes terms from a topic's entire Boolean Query negotiation history, which was expanded for some new topics in 2008. For most measures, none of these merged runs outperformed its best contributing run. Of note, though, is the two-run merge, which collected the highest MAP assessment for all submissions from all participants in the TREC 2008 Legal Track ad hoc task.

The concept of merging different result sets is explored in greater depth in the next chapter as part of our efforts to incorporate existing relevance judgments as data for

Table 10. Results for runs submitted to the 2008 TREC Legal Track ad hoc task to test the result set merging strategy.

Run	EstF1-R	Est_RB	MAP
4:QE	0.2210	0.2893	0.1500
5:QE+AR	0.2275	0.2937	0.1495
Merge-4-5	0.2219	0.2931	0.1519
8:QE+EB	0.2208	0.2902	0.1490
Merge-4-5-8	0.2181	0.2876	0.1494

supervised training of classifier-based rankers. The approach described here, for the ad hoc task, is derived more from heuristics for merging, and no supervision is applied to improve the rankings.

4.11 Additional Complaint Information

TREC Legal topics are extraordinarily rich with information which largely remains unexploited when constructing queries. The topics are presented as complaints to be addressed in the course of a larger lawsuit. For instance, complaints about marketing to children, marketing through television, and marketing through sporting events may be covered under the umbrella of a larger lawsuit. The complication with information contained within the "complaint fields" is its presence amongst lengthy statements designed to meet legal requirements. Furthermore, the contents of these fields are typically identical across the set of topics that represent a single lawsuit. The scenarios used to construct the topics are designed to resemble actual complaints but with identifying information altered to protect the process. This introduces language to the

topic that may include misleading terms (i.e., references to “Dromedary brand cigarettes”).

Our current system does not utilize any of the complaint information, but there are many ways to determine which, if any, of these topic fields are useful for retrieval. But future research would reflect the addition of the complaint information fields to the query term vector in a manner consistent with the RequestText, including reducing the weight or eliminating terms that are not specific to the topic. We could also construct queries using only the complaint fields, and use the retrieved documents as a set of documents that we would boost when they are retrieved as a response to topic-based queries. This approach would be similar to how we utilize the results from the reference Boolean runs. Finally, future research would benefit from analysis of documents found to be relevant for multiple complaints of a lawsuit. It may be useful to analyze these documents to find terms or other features that can be used to enhance our use of all of the information about a topic that is made available as part of a production request.

4.12 TREC Legal Ad Hoc Task Submissions and Results

We submitted the following runs to the TREC Legal Ad Hoc Task in 2007 and 2008. The runs are described below. Their performance is summarized in Table 11, with our best performing runs in each year highlighted in bold for each measure.

- *IowaSL07REF*: The query term vector is built from the contents of the topic RequestText field after it is processed by the Lucene StandardAnalyzer, which removes stopwords and punctuation. (2007)
- *IowaSL0702*: The same as IowaSL07REF, except that terms from all three Boolean queries are added to the term vector. Wildcards remain unexpanded. (2007)

- *IowaSL0703*: The same as IowaSL0702, except that wildcards from the Boolean queries are replaced with two eligible candidate terms, as described in §4.3. (2007)
- *IowaSL0704*: The same as IowaSL0703, except that we have now added pseudo-relevance feedback query expansion terms to the term vector as described in §4.6. (2007)
- *IowaSL0705*: The same as IowaSL0704, except that, as a preprocessing step, we have applied AutoReduce to the RequestText field, as described in §4.7. (2007)
- *IowaSL0706*: The same as IowaSL0705, except that all Boolean queries, save the one representing a final query negotiated between both sides of the legal dispute, have been dropped from the query term vector source. (2007)
- *IowaSL0707*: The same as IowaSL0706, except that, as a preprocessing step, we have applied ManualReduce to the RequestText field, as described in §4.7. (2007)
- *IowaSL08Ref*: The query term vector is built from the contents of the topic RequestText field after it is processed by the Lucene StandardAnalyzer, the same as IowaSL07REF. (2008)
- *IowaSL0804*: The same strategy is used as IowaSL0704. (2008)
- *IowaSL0804b*: The documents returned are the same as IowaSL0804, but the Okapi-BM25 scores for documents also found by the Boolean Reference run are multiplied by a "boost" factor of 1.8, and the set is then reranked, as described in §4.8. (2008)
- *IowaSL0805*: The same strategy is used as IowaSL0705. (2008)
- *IowaSL0805b*: The documents returned are the same as IowaSL0805, but the Okapi-BM25 scores for documents also found by the Boolean Reference run

are multiplied by a "boost" factor of 1.5, and the set is then reranked, as described in §4.8. (2008)

- *IowaSL0808b*: The same as IowaSL0804b, except that the query term vector includes terms drawn from the extended set of Boolean queries that represent a negotiation history that were made available for TREC 2008. (2008)
- *IowaSL08m2*: This run is generated by combining the results from IowaSL0804b and IowaSL0805b. The results are fused using a weighted CombSum (combined sum) where the weight assigned to each run is based on the estimated $F_1@R$ measure for the contributing run's strategy when applied to the 2007 topics. (2008)
- *IowaSL08m3*: This run is generated by combining the results from IowaSL0804b, IowaSL0805b, and IowaSL0808b. The merging strategy is a weighted CombSum (combined sum), applying arbitrary weights of 2, 1, and 3 respectively, such that the queries with the most information are given the most weight. (2008)

Our results from the TREC Legal evaluations are consistent with several of our expectations. Adding the query expansion strategies to our system resulted in a significant ($\alpha \leq 0.05$) improvement in estimated recall. The mean average precision of our system also benefitted from an improvement, at the expense of recall, when the automatic reduction of query terms was applied. We should note, though, that this result was reversed with the 2008 evaluation, possibly due to the greater size of the retrieved result size. The 2008 results also demonstrate the effectiveness of using the results of the negotiated Boolean query as a lever for reordering the results from the ranked retrieval. The improvements in precision with the slight decline in recall suggest the usefulness of privileging those documents that respond to the Boolean query, but that it fails to help us find relevant documents amongst those with Okapi scores that place them outside of the run size threshold.

Table 11. Official submissions to the 2007 and 2008 TREC Legal Track ad hoc tasks

Run	Est_F1@R*	Est_R@B	MAP
IowaSL07REF	0.1188	0.1078	0.0884
IowaSL0702	0.1327	0.1341	0.1286
IowaSL0703	0.1503	0.1639	0.1304
IowaSL0704	0.1577†	0.1669‡	0.1325
IowaSL0705	0.1257	0.1613	0.1394†
IowaSL0706	0.1287	0.1531	0.1288
IowaSL0707	0.1000	0.1419	0.1200
IowaSL08Ref	0.1523	0.1685	0.0401
IowaSL0804	0.2081	0.2921	0.1426
IowaSL0804b	0.2210	0.2893	0.1500
IowaSL0805	0.2187	0.2938†	0.1379
IowaSL0805b	0.2275†	0.2937	0.1495
IowaSL0808b	0.2140	0.2877	0.1474
IowaSL08m2	0.2219	0.2931	0.1519†
IowaSL08m3	0.2181	0.2876	0.1494

*Est_F1@R results were not officially calculated in 2007; results were generated using the 2008 evaluation software provided by the TREC organizers.

In summary, we have identified several strategies that, when combined, have demonstrated proficiency in the high recall information retrieval task modeled after the legal discovery of electronically stored information. These tasks are typically conducted against collections documents created from diverse sources, and will often include material that has been digitized and converted to searchable text using optical character recognition. In order to accommodate the noise introduced into the corpus through OCR, we need to limit query expansion from wildcard operators, but assessing the amount of noise in an individual document will not improve retrieval performance. We note improvement from employing a term weighting schema, Okapi-BM2, that accounts for the wide range of document length present in such a collection. The complaint information used to construct the queries used for e-discovery retrieval includes excess language that can be removed, automatically, to improve precision. On the other hand, adding terms not included in the complaint but identified through pseudo relevance feedback can improve performance. Finally, we can improve precision by giving ranking boosts to documents that respond to Boolean queries constructed by legal professionals.

We note that we could not identify a method of effectively querying the wealth of document-associated non-content metadata that is typically provided with enterprise document collections. However, our queries were limited to information directly extracted from complaint information. We see future research possibilities in the potential for applying relevance feedback to searching non-content metadata fields.

CHAPTER V
THE CLASSIFIER-BASED RANKER AND E-DISCOVERY

5.1 Introduction

The information retrieval community has explored several strategies for improving results beyond relying solely on the information provided in an initial, ad hoc query. Relevance feedback, one such strategy, is generically described as using information gained from executing a prior query to inform subsequent retrieval for the same information need. Often, this information is extracted from a subset of previously retrieved documents and then used to create a new query with the intention of improving on the results of the initial query. We use an implicit variant of this approach. In contrast to standard query modification strategies, our approach to relevance feedback reorders documents in the retrieved set that have not yet been evaluated. That is, we do not conduct a second stage retrieval using terms extracted from the original document set. We use relevance feedback to rerank documents in an existing pool of retrieved but unjudged documents. Documents are reranked with the aim of maximizing precision within the pool.

Another differentiating characteristic of our system is that the pool of unjudged documents is defined by merging several ranked results. This differs from traditional relevance feedback, which operates on the results from a single run. Here the results may be the outcomes of different search systems or different parameter configurations of the same system, all operating on the same topic and collection of documents. Combining, or merging, a set of ranked lists generated using different methods, to create an improved list that reflects the best of these strategies, is known as *rank aggregation*. Rank aggregation is well studied, with established techniques range from simple heuristics to

sophisticated applications of probabilistic strategies. Our system employs relevance judgments to perform the merging of several ranked lists into a single ranked list.

5.2 Relevance Feedback and the TREC Legal Track

In 2008, the TREC Legal track added a relevance feedback task to complement the ad hoc retrieval task. The relevance feedback task, designed to assess "the utility of a two-pass search process," recycled topics from previous years [63]. Participants were allowed to use relevance judgments from previous years when processing the topics for the current task. We explore the use of these relevance judgments with our supervised rank aggregation methods.

The same performance measures are calculated for evaluating both ad hoc and relevance feedback task submissions. The difference in evaluation procedures for the relevance feedback task is that only new judgments are used in the calculations, and documents judged in previous years are discarded from the submissions.

5.3 Standard Relevance Feedback

Although our main emphasis is on supervised merging of ranked retrieval sets, we also explore the traditional approach to relevance feedback. In particular, we extract terms from documents that are known to be relevant, and use those terms to create a new query. In particular, we modified our approach to pseudo-relevance feedback, described in §4.6, into true relevance feedback by incorporating actual relevance judgments. Because we have a partial list of relevant documents from the collection, our first modification is that we look at the top-ranked *n* documents that are known to be relevant, instead of the top-ranked *n* documents regardless of relevance status. System training in this traditional approach is supervised now that we have a set of relevance judgments. Using the same term selection formula that we use for the pseudo-relevance feedback term selection for ad hoc retrieval, we found, through training on the TREC 2006 and

2007 Legal Track topics that the best results were generated using 10 terms extracted from 12 known relevant documents.

5.4 Supervised Rank Aggregation and the Relevance Feedback Task

TREC participants have access to submissions contributed by all participating groups from previous years. A submission consists of a list of retrieved documents, ranked by estimated probability of relevance, for each of several topics designed to represent diverse information needs within the task. Each participating team independently constructs their own document indexing and retrieval systems and each submission represents a distinct strategy as implemented by a particular team. The union of the collected submissions for any TREC Legal topic, which we refer to as the "pool", consists of all documents retrieved and submitted that may or may not have been judged for the topic.

We use information about the judged documents, such as their rank and the characteristics of their neighborhood of documents within each of the submitted rankings. We aggregate this information across the submissions. This information is then applied to the task of estimating the relevance potential of each unjudged document in the pool. We train a document classifier on this information and then use probabilities derived from the output of the classifier to rank the unjudged documents in decreasing likelihood of relevance.

Merged rankings can also be created without relevance feedback information. Unsupervised merging techniques, such us the summed rank (Borda Count) approach that we implemented for our 2008 TREC Legal Track ad hoc task submissions, can be used effectively for this purpose. Our system seeks to improve on these unsupervised data merging strategies. We show that our classifier-based system accomplishes this goal when considering smaller pools of unjudged documents. This improvement is retained,

Run1	N 82e	N 55f	R 75d	N 18c	R 37d	— 66c	R 80a	N 72f	N 19d	— 39e	…
Run2	R 75b	R 75d	N 00e	N 11e	N 81c	N 82e	R 32e	N 19d	R 72e	R 88c	…
Run3	N 75a	N 32d	N 15c	N 15a	N 32a	— 15b	— 43e	— 87d	— 42c	— 15d	…
Run4	N 10e	N 62f	R 39c	R 39a	N 35c	R 89e	N 54a	— 03d	— 13d	— 13a	…
Run5	N 00e	N 11e	R 41d	N 03c	N 61c	R 24d	R 36c	R 33f	R 71d	— 70d	…

Figure 5. The top 10 ranked documents from five TREC Legal Track ad hoc submissions for topic 80, with duplicate documents highlighted

even if the heuristic aggregation strategies employ weights indicating the performance of the contributing runs. We note that this weighting is itself a subtle form of relevance feedback, and so this baseline is also supervised.

5.5 System Inputs

For each topic that has been assessed, the TREC organizers provide all of the relevance judgments made. Thus it is possible to represent each submitted ranked list as a series of documents that are either Relevant (R), Non-Relevant (N), or unjudged (−).* Figure 5 shows the top ten documents from five different submissions. As the figure demonstrates, it is possible for a document to appear in multiple ranked lists. We also see that documents (distinguished from each other by using a shortened document id such as "55f") may be characterized by the surrounding neighborhood of Relevant, Non-Relevant, and unjudged documents.

*For the TREC Legal Track, a portion of evaluated documents are deemed to be "unjudgeable" for any of a variety of reasons—document length, document legibility, evaluator uncertainty. For the purposes of this research, we consider these documents to be unjudged. For the 2008 TREC Legal Track, the additional judgment category "Highly Relevant" is added. For the purposes of this research, we consider these documents to be "Relevant."

Given a pool of documents created from a group of submissions, each unique document is treated as an instance for the purpose of training our classifier. Features are then calculated for each document instance. We explore several types of positional features. Our features are designed to summarize a document's neighborhood characteristics. It may be that information about surrounding documents indicates the likely relevance of an unjudged document. Our features take into account the possible presence of a document in multiple runs. Since the same document can appear at different ranks in different runs, attempts to utilize information about surrounding documents need to be able to aggregate this information in a meaningful way.

5.5.1 Contributing Runs and Document Pool Definitions

Our system takes as input the ranked retrieval results from any number of distinct runs, representing submissions from different systems, strategies, and/or sets of parameters used to configure the systems. As mentioned earlier, all runs submitted to TREC are made available to participants. The first question we address concerns which of the available runs for a given topic we should try to merge. One broad solution is to merge all available runs. As an alternative, we could identify and merge subsets of the collection of ranked lists, where these subsets are chosen to reflect different perspectives. Here, we explore combinations of runs consisting of:

- The set of all available runs. This model provides the maximum amount of information to the classifier about a given document. However, some of the contributing runs may have performed poorly, and may provided distorted information to the classifier.
- Half of the runs, selected by the best mean precision at 100 documents. Here we attempt to limit the information available to the classifier to sources that are perceived to be of higher quality. It is possible, however, that information from the remaining runs may provide the learner with a greater ability to

discriminate between documents that are likely to be relevant and those likely to be non-relevant.

- A small set of the runs with the highest and lowest scores for mean precision. Here we take the opposite approach to the above strategy, in that we try to provide the classifier with highly variable quality of information.
- A family of subsets where each represents the contributions from a single participating organization. We note that outside of the experimental TREC environment, the application of the first strategy will be constrained by the user's ability to replicate the variety of retrieval systems employed by the organizations that participate in the TREC Legal Track. These organization specific subsets are designed to constrain the system so that runs from one organization are available.

Each run combination yields a distinct pool of documents to be merged. We have assessed our supervised method with the different document pools in comparison with heuristic or other simplified merging techniques.

5.5.2 Depth

Additional criteria have an impact on the construction of the document pools. Besides the set of ranked lists that are included in the pool, we also need to consider how many documents are contributed to the pool from each run for a given topic. We use the depth parameter *d* to quantify this number. A ranked list from a submission needs to have at least *d* elements in order to be included in the document pool. In particular, the experiments that follow reflect relatively shallow document pools ($d = 100$). In this chapter, we briefly consider the consequences when applying our supervised system towards creating a large-scale ranking of 100,000 documents, the 2008 TREC Legal Track maximum submission size. For these submissions to TREC, we accept the entire

contents of runs from previous years (d = 5,000 or d = 25,000, depending on the TREC specification that year).

In this model, our document pools can be described by the set of contributing ranked lists and d. In the next chapter, we increase the value of d and we examine the results from the large scale aggregations.

5.6 Training and Testing Datasets

An important step of our classifier-based procedure is the identification of testing and training datasets. Prior to the calculation of the features, some of the judged documents are selected for training our classifier with the rest used to test performance. Further, depending on its application, these two datasets are used in different manners. In our first study, which we consider a pilot study, it was not feasible to acquire additional relevance judgments on previously unjudged documents. Thus, evaluation is conducted using a "leave-one-out" strategy. Given a pool of judged documents, we train the system on all but one of the documents and test it on the single document, using the calculated probability estimate as the ranking score for that document. After iterating through this pool, we are able to use the ranking scores to construct a new ordered list of the documents. We evaluate our system by comparing this list to other ranked lists generated using baseline techniques.

The experiments conducted with larger values for d were submitted to the TREC 2008 Legal Track relevance feedback track for evaluation. This provided a natural way to design the experiment. System training was conducted using all documents that had been judged during prior iterations of the TREC Legal Track ad hoc task. The system is tested by applying the models derived from the judged documents to all of the unjudged documents in the specified document pool. The classifier-assigned score is then used to rank the documents for submission to TREC.

5.7 Document Features

As mentioned earlier, each document in the pool is considered to be an "instance" for the purpose of training or testing the classifier-based ranker. The sets of features that our system calculates for each document instance are derived from the position of the document relative to known relevant and non-relevant documents, as well as information retrieval statistics calculated at various depths. Table 12 describes the calculated features that we have implemented. All of the features are calculated by first locating the document in each run that may contain it. Features such as "RetrieveTopic" or "BestRankTopic" are determined by counting the number of runs contributing the document or by tracking the best rank. Windowed features are used to describe the "neighborhood" of the document, and each of these features concerns either Relevant or Non-relevant documents. N determines the size of the neighborhood, and in our experiments, $N \in \{1, 2, 3, 4, 5, 10\}$. "Plus" or "Minus" in the feature name indicates whether the neighborhood consists of elements ranked higher or lower in the contributing ranking list. The table also identifies features that we will explore in future research.

Neighborhood features are calculated in four different ways. The most straightforward of these methods involves determining the proportion of Relevant or Non-relevant documents in the neighborhood across all runs. However, if a document is at the top of a ranked list, it is possible that its contribution to this calculation may be underweighted. A windowed feature with balanced weighting calculates the neighborhood proportion for each contributing ranking with the document, and then calculates the mean of these proportions. We also calculate windowed features where the proportions are calculated for each contributing run with the document, but the precision for the run is used to weight the calculation of the mean. Finally, we also calculate another window feature where the weights are based on the contributing run's Recall of relevant documents from the document pool.

Table 12. Calculated features that describe the document

Class of Feature	Description	Implemented
RetrieveTopic	Number of times document is retrieved	X
BestRankTopic	Highest rank for document across runs	X
WorstRankTopic	Lowest rank for document across runs	X
AvgRankTopic	Average rank for document across all runs containing the document	X
WinPlus*N*Rel	Proportion of *N* documents ranked immediately higher than this one that are relevant. Accumulated across all runs containing document.	X
WinPlus*N*Non	Proportion of *N* documents ranked immediately higher than this one that are not relevant. Accumulated across all runs containing document.	X
WinMinus*N*Rel	Proportion of *N* documents ranked immediately lower than this one that are relevant. Accumulated across all runs containing document.	X
WinMinus*N*Non	Proportion of *N* documents ranked immediately lower than this one that are not relevant. Accumulated across all runs containing document.	X
BalWin…	Window measures except proportions are calculated locally and averaged across all runs containing the document.	X
WeightedWin…	Window measures where proportions are calculated locally, and combined to create a weighted average across all runs containing the document. Weights are determined by run performance in precision or recall.	X
TF-IDF	Document similarity to query based on TF-IDF score	
Okapi-BM25	Document similarity to query based on Okapi-BM25 score	
TF	Accumulated number of terms in document matching terms in query vector	
DocLength	Number of tokens in document	
OCR-L{0,1,2,3,4,5}	Term distinctiveness proportions (see §4.9)	

For each of the four windowed feature calculations, we calculate each possible combination of the feature parameters described above, for both relevant and non-relevant documents, for a total of 96 neighborhood features. For instance, the "BalWinPlus2Non" calculates the proportion of Non-relevant documents in the two documents that precede the document, and then averages this proportion against all runs containing the document. Including the three features based on the ranking of the document across all contributing runs, as well as the "RetrieveTopic" feature we described above, we calculate 100 features to describe each document instance in the pool.

The current set of features is a portion of those possible, and in future work we will explore additional document characteristics. In particular, our system calculates document features that are specific to the document in the context of the run. These features are based on the document position, or the frequency and quality of relevance judgments of documents both before and after the document. We observe that documents in the tail end of ranked retrieval sets frequently have few neighbors with judgments, rendering many of our run-context features useless as distinguishing characteristics for estimating probability of relevance. As we expand our studies, we will add features with more expansive definitions of a "neighborhood" to better handle the sparse data available to us.

One way to view the features of our system is to look at the different levels of context where we can find the document. The features that our system currently uses are an attempt to describe a document solely by placing it in the context of its position relative to other documents in the runs in which it was found. Other features could describe the document relative to the topic at hand (e.g., BM25 score calculated for the document and the topic query), while others could simply describe the document within the context of the corpus (i.e., normalized document length, or OCR-error).

5.8 Baseline Techniques

Our system is a supervised learner designed to fuse multiple ranked lists. The phrase "rank aggregation" describes the task; given a collection of ranked lists with possible overlap in the included elements, it attempts to generate an optimal ranking using the available elements. It is supervised through the use of relevance judgments while training, which provide guidance for an idealized ranking. Our system also has access to performance information for the strategies that generated each of the ranked lists.

During our pilot study, we set $d = 100$, and we compared our system against a data fusion strategy known as Borda Count [4, 78]. The premise of this strategy is simple. A ranking score is assigned to a document by adding its ranks across all lists in which it appears, including a default (large) score for each run that it is not found in. Documents are then ranked in ascending order according to this ranking score. This essentially privileges documents that are highly ranked across multiple runs. Our system is compared against this baseline Borda Count. We also compare it against a weighted Borda Count, which uses run performance in Precision at a depth of 100 documents as a weight for the run's contribution to the ranking score. Our results demonstrate that, with this limited amount of knowledge about run performance information, the weighted Borda Count offers considerable improvements over the basic Borda count.

5.9 Classifier Selection

We use the sequential minimal optimization (SMO) algorithm implemented as part of the Waikato Environment for Knowledge Analysis (WEKA) 3.5.7 to train a support vector machine [28, 64, 99]. This classifier was selected after testing the effectiveness of different classification algorithms at the task of identifying relevant and non-relevant documents from the TREC Legal Track ad hoc data set. We use the default configuration for the WEKA SMO classifier—training data is normalized, the complexity

parameter is set to 1.0, and the machine employs a polynomial kernel with an exponent of 1.0. These experiments were conducted using the WEKA Experimenter—a framework designed for comparing classifiers in a limited environment—which allowed us to perform cross-validation on static datasets..

The selection of the SMO classifier was completed before we developed a system that recalculates document features for different folds of cross-validation, along with the means for evaluating effectiveness at ranking the merged datasets. This recalculation is important, because removing a document in a ranked list for the purpose of cross-validation will affect the positional data concerning the other ranking of other documents. Removing a single document from the pool will also affect the proximity features of any document in its "neighborhood" in any of the runs. Revisiting the classifier selection would be appropriate for further explorations.

Recall that, in order to rank the documents in our test set, we need to map the classifier output to a non-binary probability distribution. WEKA's implementation of the SMO classifier provides for this by fitting a logistic regression, built on the training data, to the model's output. But SVMs, as "optimizing" classifiers, have a potentially useful built-in ranking heuristic derived from the margin maximizing behavior [101]. Other possible ranking systems to implement for comparison are Brefeld's support vector machine optimized to maximize the AUC measure [5, 14], a ranking classifier based on an ensemble method along the lines of Freund's "RankBoost" [36], or using a linear programming formulation that maximizes an approximation of the Wilcoxon-Mann-Whitney statistic that is calculated for the output ranking [5].

Ensemble classifiers are classifiers that use the predictions of several classifiers to create an overall prediction. The metaphor is often a committee where everyone has a vote, often of differing weights. On a simple level, we can build ensemble classifiers where each classifier represents a different set of judged documents. Additionally, we

can construct ensembles that resemble set of ensemble strategies such as Boosting, Bagging and appropriate committee-style collections of voting classifiers.

5.10 Pilot Experiment and Results

Our first set of experiments is a pilot study. We explore the following pools of runs with depth d = 100. Features are constructed as described in §5.7. We employ a leave-one-out study design; we train an SMO-based classifier on all but one of the judged documents in the pool, apply it to the remaining document to get the ranking score, and then repeat for each of the judged documents. We used the following fifteen subsets of contributing runs to build document pools. These are described in greater detail in §5.5.1.

- All available runs (All60)
- The top half of the runs ranked by their performance in precision (Top30Prec)
- The top seven runs ranked by their performance in precision (Top7Prec)
- The top seven runs ranked by their performance in recall (Top7Rec)
- The top four and bottom four ranked by their performance in precision (MixedRuns)
- Ten runs each defined solely by a different organization's contributions to TREC (CMU [103], Fudan*, IowaE [34], IowaS [1], OpenText [88], Sabir [15], UMass [93], UMKC [102], Ursinus [50], Waterloo [18]).

The merged sets of ranked documents generated by our classifier-based ranker are evaluated by using precision and recall measures commonly used in information retrieval research. These are compared to rankings generated by other merging techniques. Precision and recall are calculated at depths of 5, 10, 15, 20 and 25. The shallow depths of these evaluations are because of the small amount of judged documents in the

* Fudan University did not submit a paper to the TREC proceedings in support of their submissions to the TREC Legal Track.

document pools for some topics, especially when using smaller subsets of contributing runs. Note that recall here considers only the documents present in the pool and ignores all others.

At each evaluation depth, similar measures are calculated for ranked lists generated using the two Borda Count strategies described above. To determine weights for the weighted Borda Count, we calculated Mean Precision at 100 and Mean Recall across all topics for each of the contributing ranked lists. Two tests, the Wilcoxon Signed Rank test and Fisher's Randomization test, are executed to determine the significance of the improvement by the classification ranker over each of the two baselines for each of the two measures at each of the five depths [79]. With fifteen subsets of contributing runs, five depths, two measures, comparisons against two baselines, and two different significance tests, we conducted 600 significance tests over the course of the pilot study. For each of the fifteen subsets, we report the number of significant improvements (two-sided, $\alpha \leq 0.10$), and declines by the classifier compared to the baseline methods over the forty possible tests.

Results from these experiments are presented in Table 13. The table indicates how many runs are included in each document pool. For each subset, we report how many tests, out of forty, that the classifier-based ranker significantly outperformed or underperformed either of the baselines. The improvements are tracked at two different significance levels ($\alpha \leq 0.10$ and $\alpha \leq 0.05$), and reported as "Wins." Significant ($\alpha \leq 0.10$) declines are reported as "Losses," and we also indicate how many tests failed to indicate a significant result in either direction. The table is ordered by the total number of "Wins" assigned to the classifier-based ranker. For example, with the subset identified as "MixedRuns," we used eight runs to build the document pool. Precision at 100 documents was 0.1023 for all topics when averaged across all eight runs. Likewise, the eight runs retrieved, on average, 16.70% of all available relevant documents, when averaged across all topics under consideration. The classifier-based ranker generated

significant improvements over either the Borda Count or weighted Borda Count baseline in 34 of the 40 (85%) significance tests that we conducted for this subset. We detected no significant declines by the classifier-based ranker against either baseline. The remaining 6 comparisons yielded no significant differences.

We found that the classifier-based ranker was most successful, relative to the baseline methods, when it had a large number of contributing pools (All60, Top30Prec). This advantage is nearly eliminated if we reduce the number of runs in the subset while selecting runs for performance (Top7Prec, Top7Rec). On the other hand, this advantage is retained for a smaller subset if the runs are selected to maximize the variance in retrieval performance (MixedRuns). We found that the classifier-based ranker outperformed the heuristic methods when limiting the pool of documents to runs from some, but not all, individual organizations (IowaS, CMU, UMass). This indicates that pool definition is critical to the success of the classifier-based ranker.

In Table 13, we have summarized performance over all five measurement depths. In Table 14, we present the results from each of the ranking methods for precision and recall calculated at a depth of five documents. We have indicated, for each baseline score, the significance level of any improvements or declines by the classifier-based ranker ("CBR") as established using Fisher's randomization test. The baselines, Borda Count and Weighted Borda Count, are indicated by "BC" and "WBC" respectively. Tables for the other four depths have been placed in Appendix A.

These initial experiments suggest that use of the classifier-based ranker is a valid technique applied against small pools of documents when all obtainable knowledge is made available to the system. In Table 14, it is important to note that the precision and recall results between different subsets for precision and recall measures are not directly comparable as the pool of documents for each subset will contain a different distribution of relevant and non-relevant documents. In many cases, the contents of the subsets will have an impact on these measures. For instance, the "MixedRuns" subset, where the

Table 13. Subsets of contributors to document pools and the accumulated results of comparing the classifier-based ranker against two baseline methods

Subset	Runs	Mean Precision @ 100	Mean Recall	Wins @.05	Wins @.10	Win%	Nonsig	Loss @.10
All60	60	0.1240	0.1950	37	3	1.000	0	0
MixedRuns	8	0.1023	0.1670	31	3	0.850	6	0
Top30Prec	30	0.1599	0.2473	27	6	0.825	7	0
IowaS	7	0.1624	0.2433	26	5	0.775	9	0
CMU	8	0.1529	0.2364	21	10	0.775	9	0
UMass	6	0.1429	0.2207	26	5	0.775	9	0
Waterloo	6	0.1282	0.2112	6	5	0.275	29	0
Fudan	5	0.1113	0.1816	10	1	0.275	29	0
UMKC	4	0.1340	0.2076	9	2	0.275	29	0
Ursinus	8	0.0763	0.1111	7	1	0.200	29	3
IowaE	4	0.0499	0.0893	0	2	0.050	35	3
Top7Prec	7	0.1760	0.2753	0	1	0.025	39	0
Top7Recall	7	0.1685	0.2843	0	0	0	40	0
OpenText	6	0.1661	0.2753	0	0	0	39	1
Sabir	4	0.1055	0.1547	0	0	0	40	0

classifier-based ranker records the highest precision-at-5 measurement, has the third-smallest proportion of relevant documents per topic of all the sets. In recall-at-5, the weighted Borda count records the highest measure for a subset which contained on average for each topic, 17.78 relevant in each pool, in contrast to the 34.12 average per pool across all subsets.

The classifier-based ranker achieved significant improvements over both heuristic merging techniques for several subsets at measurement depth of 5, with only a single significant decline against the weighted Borda Count. 12 of the improvements at

Table 14. Recall and precision measures assessed at depth 5 for each subset of contributing runs

Contributing Runs	*Precision*			*Recall*		
	CBR	**WBC**	**BC**	**CBR**	**WBC**	**BC**
All60	0.6465	0.5349‡	0.5302‡	0.0687	0.0528‡	0.0554†
MixedRuns	0.6857	0.5524‡	0.5429‡	0.1075	0.0943	0.0868‡
Top30Prec	0.6233	0.5535	0.5209	0.0898	0.0725	0.0603
IowaS	0.5810	0.4143‡	0.4524‡	0.1430	0.0985‡	0.1140‡
CMU	0.5814	0.4977†	0.4884†	0.1308	0.0909‡	0.0956‡
UMass	0.6558	0.5767†	0.5163‡	0.1722	0.1869	0.1169‡
Waterloo	0.5302	0.4698	0.4605	0.1122	0.0994	0.0790
Fudan	0.5810	0.5619	0.5381	0.2102	0.2111	0.1970
UMKC	0.6095	0.5714	0.5143‡	0.1868	0.2052	0.1643
Ursinus	0.6000	0.5366	0.5268	0.1661	0.1816	0.1920
IowaE	0.5800	0.6050	0.5500	0.2460	0.2989*	0.2559
Top7Prec	0.5571	0.5286	0.5571	0.0847	0.0698	0.0874
Top7Recall	0.5667	0.5476	0.5857	0.0881	0.0807	0.0958
OpenText	0.5333	0.5286	0.5286	0.0897	0.0740	0.0847
Sabir	0.5512	0.5659	0.5366	0.2422	0.2536	0.2625

† indicates that the classifier-based ranker outperformed the specified baseline at $\alpha \leq 0.10$ using the randomization test.

‡ indicates that the classifying ranker outperformed the specified baseline at $\alpha \leq 0.05$ using the randomization test.

* indicates that the classifying ranker underperformed the specified baseline at $\alpha \leq 0.10$ using the randomization test.

measurement depth 5, including both Wilcoxon and randomization significance tests, occurred in precision, and 8 in recall. Over the entire set of measurement depths, 63% of the significant improvements generated by our system were for the tests on the precision measurements, as opposed to improvements in the recall of relevant documents existing in the pool. Precision, as a measure, will be more sensitive to ordering of documents. Recall, especially at shallow depths, will also be sensitive to document ordering, except in the cases where there are very few relevant documents. If we are calculating our measures at 25 documents, if there are only three relevant documents in the pool, the recall measure does not distinguish whether the three documents are placed in the top five, or in positions 23-25. The topics with few relevant documents will potentially present as ties in recall even if the precision measures are quite distinct. These ties are eliminated from the number of comparisons considered by the significance tests, thus increasing the difficulty of exceeding the threshold for significance.

We also conclude that the weighted Borda Count method is a more difficult baseline, as only 38.5% of the significant improvements were generated over this strategy. The weighted Borda Count is the only baseline to significantly outperform the classifier-based ranker in any of the tests.

5.11 TREC 2008 – Experiments and Results

The pilot experiment described in the previous section was conducted on small document sets, taking the top 100 documents from each contributing run, and building our training/testing sets from those pools. The 2008 TREC Legal Relevance Feedback task allows the opportunity to test the classifying ranker on larger scale return sets. By analyzing performance from TREC, it is possible to assess the fitness of the ranking classifier system for larger and more realistic environments. In particular, it allows us to see how the system performs when trained with limited pools of judged documents but tested against large document sets.

For the TREC 2008 relevance feedback task, we submitted results from the ranking classifier using three different pools of documents compiled from three different sets of contributing runs. For a baseline, we used strategy similar to Borda Count known as rCombMNZ [62, 78], where the initial summed ranking score is multiplied by the number of contributing runs that contain the document in question. We established a second baseline using a more traditional relevance feedback approach by modifying an original adhoc query with information gained from relevance feedback documents. The pools used for TREC submissions include:

- All available runs
- Runs submitted to the 2007 TREC Legal Track ad hoc task from our research group ("IowaS")
- The top four runs and the bottom four runs from each year based on the year's evaluation measure (R-Prec for 2006, Estimated Recall at B for 2007).
- One additional result set was submitted by applying the classifier models trained using all available runs to the test set of documents from the "IowaS" research group.

Results from TREC suggested that the ranking classifier strategy, as implemented, does not perform favorably to the baseline method, rCombMNZ [2]. When ranking documents from each of the three pools, the rCombMNZ ranking outperformed the classifier-based ranker, for both the balanced F1 at R and estimated Recall at B measures. In particular, we note that, for the "diversity" pool, in F1 at R, the classifier-based ranker significantly underperformed against the baseline (28.12%, $\alpha \leq 0.05$). Estimates indicate that the aggregated runs are successful at recall when considering the entire submission, confirming the effectiveness of aggregation strategies for high-recall information retrieval problems. Follow-up retrieval for evaluation, which needs to be ranked for improved precision, is hampered by the classifier-based ranker strategies when using the full TREC scale result sets. We further explore these results in chapter 6, but additional

corrective measures would include designing document features that more effectively describe characteristics of documents that are ranked at judgment-sparse depths. Because the system demonstrates effectiveness with smaller datasets, another possible adjustment would to create an iterative (or active) learning environment. In this scenario, after a certain amount of additional judgments are added to the pool, we wish to see what, if any, performance gains are to be had by recalculating the features, retraining our models, and reranking the remaining unjudged documents.

In summary, we learn from this chapter that the classifier-based ranker has potential as a rank aggregation method, but with many variables remaining to be explored. We have found that it outperforms effective heuristic baselines when considering data sets with a limited size, though this improvement appears to be subject to the selection of sources to be aggregated. We have found that rank aggregation is effective for large-scale queries that require high recall. At that scale, the classifier-based ranker is, however, not as effective as the heuristic Borda Count or traditional relevance feedback techniques at prioritizing relevant documents.

The classifier-based ranker is apparently hampered by the limited number of relevance judgments available at lower ranks in contributing ranked lists. This suggests that document features that are not dependent on the presence of nearby judged documents in ranked lists might improve the performance of the classifier-based ranker for problems requiring a greater depth. The selection and configuration of the classifier itself is also grounds for further inquiry, as it was selected based on its performance at a simple classification task, and not on its effectiveness at ranking.

CHAPTER VI

EXTENDING THE CLASSIFIER-BASED RANKER

The classifier-based ranker described in the previous chapter has demonstrated its effectiveness when considered against popular and powerful heuristic techniques for rank aggregation. But this effectiveness was limited to a constrained setting. In our pilot study, we limited the pools of documents to the union of only the top 100 (d = 100) ranked documents from each of the selected contributing runs. The relative success of this system was not matched when the size of the contributing runs and the resulting output was increased to a much larger scale consistent with high-recall searches. When building larger pools from the union of all documents from contributing runs, the classifier-based ranker underperformed in comparison to the heuristic baseline, rCombMNZ. The disadvantage in larger pools is that the relevance data used to create our largely position-based features becomes sparse. In this chapter, one of our goals is to explore pool depth greater than 100. In particular, we increase d to 1000 for the experiments in this chapter in order to increase our understanding of the effect of pool depth on performance.

We also assess the robustness of our method for evaluating its effectiveness when applied to two other datasets. We follow this with an examination of higher level combinations of strategies. With the 2008 TREC Legal Track relevance feedback task as our basis, we create pair-wise combinations of the classifier-based ranker, the heuristic ranker, and the traditional relevance feedback retrieval methods. Here we introduce two measures—Average Precision (AP) and Area Under the ROC Curve (AUC)—to our analysis, justification of which is provided later. Our final set of experiments in this chapter concern possible optimizations derived from social preference theory, referred to as "Local Kemenization," that can be used to refine our output rank aggregations.

6.1 Robustness and Pool Depth

6.1.1 Beyond E-Discovery

As demonstrated in the previous chapter, the classifier-based ranker is effective at prioritizing documents from relatively small pools of documents, but our results could be described as limited to the particular set of results from the TREC Legal Tracks from 2006 and 2007. To evaluate the robustness of this system, it would be necessary to test the classifier-based ranker on other appropriate datasets. It has been argued that different information retrieval tasks have different optimal strategies [31, 41]. We hope to demonstrate that our system can be generalized outside of the domain that we have used to develop our system.

For comparing the effectiveness of our system across multiple environments, we need to identify test sets created for purposes distinct from the legal e-discovery task. Any such test set needs to have sufficient number of topics, a set of ranked lists that represent different retrieval strategies for each topic, and a set of judged documents selected through some sort of random sampling criteria such that each rank list contains some number of judged documents. Once appropriate test sets are identified, we can construct document pools that reflect combinations of runs similar to those created from the TREC Legal Track submissions in Chapter 5.

In this experiment, we also increase the depth of the document pools by including the top 1000 documents from each of the contributing runs, thus we filter out any run that didn't submit at least 1000 documents for each evaluated topic. Table 15 presents results for $d = 100$ and $d = 1000$ which enables us to compare the effect of increasing d. Aside from increasing the number of included documents, our pools were constructed using the same method as the "All60" pools that we constructed in Chapter 5.

We replaced the leave-one-out validation that we used for the pilot experiment (Chapter 4) with a five-fold validation method by randomly allocating judged documents

to five sets of equal size. For each set, the classifier was tested by ranking the documents in the set after being trained on the remaining documents. As with the full-scale system, ranking is based on the probability estimate calculated by using the mapping generated by fitting a logistic regression to the built classifier's outputs.

With the TREC Legal Track dataset, we were able to ensure that each topic had at least 125 judged documents that would be included in each of the subset pools, which enabled our evaluation to test for precision and recall at a depth of 25 documents for each fold of the five-fold validation. As we expanded our research to include other datasets, we found that we could not make such a guarantee. Thus, we opted to evaluate our system using measures that assess the quality of an entire ranking. Average precision (AP) and area under the ROC curve (AUC) are capable of measuring the effectiveness of a system at ranking relevant documents higher than non-relevant documents. Average Precision is often used to assess information retrieval systems, but the distribution of AP scores is subject to the underlying distribution of relevant documents. For our experiments, the AUC measure represents the probability that a randomly chosen relevant document will be assigned a higher ranking by the rank aggregation method than that assigned to a randomly chosen nonrelevant document [5, 13]. We then average the values for the measures across all five folds, and then that value is averaged over all topics in the data set.

We tested our classifier against Borda Count, a weighted Borda Count biased by run performance, and a reciprocal rank fusion technique. The Borda Count is described in §4.10, and the weighted Borda Count in §5.8. In both cases, the ranking score is calculated by counting the number of documents across all runs ranked after the document that is being scored. In Borda Fuse, however, weights—determined by the performance of the contributing runs—are assigned to the counts on a run-by-run basis. Reciprocal rank fusion uses ranking scores calculated by summing the reciprocal ranks of a document across all runs containing the document, where the reciprocal rank is

calculated as $1/(61+y)$ where the constant 61, suggested in Cormack, *et al*.'s pilot study, is used to minimize the dominating effect of high-ranking documents [27]. Each of these baselines is used to rank the same set of documents in the test set for each fold. AUC and AP are then averaged across all folds and then across all of the topics in the test set, as described earlier. Significance tests are conducted to compare the classifier-based ranker scores and the results from Borda Fuse, which turns out to be the most effective of the baseline ranking methods.

6.1.2 Data Sets

First, we expanded our TREC Legal data set to include the topics and contributing runs from the 2008 iteration of the Legal Track. We also identified two datasets from the TREC Terabyte and Genome tracks that include sufficient numbers of topics to allow evaluation, a sufficient number of contributing runs, and relevance judgments assigned to documents selected through a random sample.

6.1.2.1 TREC Legal

We evaluate the three TREC Legal Tracks both separately and collectively. By increasing the minimum depth from 100 (the pilot experiment in Chapter 5) to 1000, we reduce the number of eligible contributing runs from the TREC 2006 and TREC 2007 tracks. The increased depth results in an average of 39.9% more judged documents per topic. On average, adding these documents reduced the percentage of judged documents that are relevant for each topic from 30.25% to 27.80% for an average reduction of 9.10%.

With the depth increased, we found that the performance of the classifier-based ranker was essentially unchanged for both Average Precision and AUC (Table 15). We also note that the Reciprocal Rank method, with AUC scores below 0.5 for both values of *d*—indicating that it is ranking non-relevant documents ahead of relevant documents—

Table 15. The effect of increasing the depth on the classifier-based ranker and three heuristic ranking methods

Merge Method	*AP*				*AUC*			
	CBR	**Recip Rank**	**Borda Count**	**WBC**	**CBR**	**Recip Rank**	**Borda Count**	**WBC**
$d = 100$	0.6010	0.3371	0.5093	0.5952	0.7657	0.4640	0.6561	0.7765
$d = 1000$	0.5906	0.2984	0.5067	0.5611	0.7775	0.4632	0.6989	0.7636
Percent Improvement	-1.73	-11.48‡	-0.51	-5.73†	1.54	-0.17	6.52*	-1.66

† indicates a significant decline in performance ($\alpha \leq 0.005$).

‡ indicates a significant decline in performance ($\alpha \leq 10^{-6}$).

* indicates a significant improvement in performance ($\alpha \leq 5\times10^{-6}$).

has significantly lower average precision. The Borda Count method delivered a significant improvement in AUC as we increased *d* to 1000, though the weighted Borda Count method suffered a decline in AP.

One effect of this expansion can be explained by reviewing the impact on the Borda Score for a document that is ranked #1 and one that is ranked at #100. When considering only the top 100 documents as the contribution to the pool, the #1 document contributes 100 "points" to its Borda score, and the #100 document is given only 1. With a 1000-document cut off, document #1 is given 1000 points, and document #100 is given 901. Increasing the depth minimizes the potentially deleterious effect on a document's Borda Score due to its placement further from the top. As this effect is broadly applied to the entire pool of documents, it appears that relevant documents are shifted up in their rankings relative to non-relevant ones, leading to an increase in the area under the ROC curve. But weighting the Borda Score according to each contributing run's precision appears to introduce distortions that negate this improvement. This, in turn, suggests that

Table 16. The percentage of improvement by the classifier-based ranker over the three heuristic methods, measured for $d = 100$ and $d = 1000$

Merge Method	AP		AUC		% change in improvement by topic	
	$d = 100$	$d = 1000$	$d = 100$	$d = 1000$	AP	AUC
Recip Rank	78.31**	97.96**	65.01**	67.84**	12.71†	5.14
Borda Count	18.02**	16.56*	16.70**	11.25*	-5.98	-27.96‡
WBC	0.98	5.25	-1.40	1.83	849.88†	-191.00†

† indicates a significant improvement in performance ($\alpha \leq 0.05$).

‡ indicates a significant improvement in performance ($\alpha \leq 0.005$).

* indicates a significant improvement in performance($\alpha \leq 5\times10^{-5}$).

** indicates a significant improvement in performance ($\alpha \leq 10^{-6}$).

the qualitative differences between runs that leads to assignment of different weights occur primarily at the top of the ranking. A run with high precision will have more non-relevant documents in its lower ranks, which we will be adding to the pool as we increase d to 1000. But these non-relevant documents will be favored against documents from a run with low precision, even though there may not be much difference in precision at increased depth.

Table 16 tracks the improvement, in percentage terms, of the classifier-based ranker over the three baseline methods. Since this improvement is significant in the case of Reciprocal Rank and the Borda Count, we want to see if the if we achieve better relative results as we increase the value of d from 100 to 1000. For example, the classifier-based ranker performs 78.31% better against Reciprocal Rank method for AP when $d = 100$. The amount of improvement increases to 97.96 % as we set $d = 1000$.

Both of these improvements are significant. We can assess this change in improvement at the topic level as we increase *d* from 100 to 1000; we can see that this increase in the amount of improvement—on average 12.71%—is itself significant. This validates increasing *d* to 1000 when assessing for AP. We also see a significant decline in the improvement from the classifier-based ranker over the Borda Count in AUC—though the retained improvements (11.25%) are themselves still significant. As the classifier-based ranker and the weighted Borda Count have essentially the same performance scores for both measures and both values of *d*, the change in improvement as *d* is increased from 100 to 1000, while significant, is not useful. Overall, it appears that $d = 1000$ is more effective than $d = 100$.

6.1.2.2 Terabyte

The TREC Terabyte Track was introduced in 2004 to address concern that the practices for pooling documents for evaluation purposes were insufficient for assessing information retrieval systems when applied to large datasets so common with modern retrieval tasks [19]. The "Terabyte" dataset consists of the "GOV2" collection—the set of documents (including websites, PDFs, Word documents, and postscript files) gathered in a 2004 crawl of the ".gov" domain. Despite the track name, this collection of 25 million documents was "only" 462 GB. In the 2006 iteration of the track, automatic runs were assessed against 149 topics including 99 that were also used in the prior two years. Retrieval submissions designated as "Manual," indicating that they were generated with human interference, were not included in the contributing runs for our document pools. Our document pools incorporated the top 1000 documents ($d = 1000$) from all available automatic submissions. The TREC organizers generated Relevance judgments for the Terabyte Track by assessing the top 50 documents from each run. We used these judgments for training and testing as we did with the TREC Legal Track dataset.

6.1.2.3 Genomics

The TREC Genomics track ran from 2003-2007. By 2005, the ad hoc task was designed to reflect the information needs of researchers accessing biomedical literature, typically through an aggregator of scientific journal articles. To model this task, the document collection consisted of a subset of the MEDLINE bibliographic database—4.6 million records and 8.9 GB of data [42]. The 2005 ad hoc task consisted of 50 topics generated by asking researchers for actual information needs. The top 60 documents from each run were selected for evaluation at TREC.

6.1.3 Results

We compared the classifier-based ranker to three baselines. For each dataset, we provide comparisons of the classifier-based ranker ("CBR") against that of the three

Table 17. Mean average precision (MAP) of the classifier-based ranker and three baselines (d = 1000)

Dataset	RR	BC	WBC	CBR	Percent Improvement (WBC → CBR)
Legal 06	0.1931[c]	0.2956[b]	0.3739	0.4905	+31.1
Legal 07	0.2740[e]	0.4961[a]	0.5529	0.5709	+3.3
Legal 08	0.3708[e]	0.6159[a]	0.6559	0.6576	+0.3
Legal (All)	0.2984[e]	0.5067[d]	0.5611	0.5906	+5.6
Terabyte 06	0.2740[e]	0.4978[e]	0.5312[e]	0.6284	+18.1
Genomics 05	0.1680[e]	0.3985	0.4565	0.4301	-5.8

[a] indicates a significant improvement in performance ($\alpha \leq 0.05$)

[b] indicates a significant improvement in performance ($\alpha \leq 0.01$)

[c] indicates a significant improvement in performance ($\alpha \leq 5\times10^{-4}$)

[d] indicates a significant improvement in performance ($\alpha \leq 5\times10^{-6}$)

[e] indicates a significant improvement in performance ($\alpha \leq 10^{-6}$)

Table 18. Area under the ROC (AUC) of the classifier-based ranker and three baselines (d = 1000)

Dataset	RR	BC	WBC	CBR	Percent Improvement (WBC → CBR)
Legal 06	0.4454[c]	0.5787[a]	0.7064	0.7476	+5.8
Legal 07	0.4663[e]	0.7326[a]	0.7868	0.7836	−3.3
Legal 08	0.4691[e]	0.7232[c]	0.7701	0.7866	+2.1
Legal (All)	0.4632[e]	0.6989[d]	0.7636	0.7775	+1.8
Terabyte 06	0.5062[e]	0.7408[e]	0.7721[e]	0.8298	+7.4
Genomics 05	0.4285[e]	0.7266[a]	0.7846	0.7686	-2.0

[a] indicates a significant improvement in performance ($\alpha \leq 0.05$)

[c] indicates a significant improvement in performance ($\alpha \leq 5\times10^{-4}$)

[d] indicates a significant improvement in performance ($\alpha \leq 5\times10^{-6}$)

[e] indicates a significant improvement in performance ($\alpha \leq 10^{-6}$)

baseline systems: Reciprocal Rank ("RR"), Borda Count ("BC"), and Weighted Borda Count ("WBC"). The results using the Average Precision measure are included in Table 17 and the results using Area Under the ROC Curve are in Table 18.

The classifier-based ranker significantly outperformed the strongest of the heuristic ranking methods on the Terabyte dataset, and no significant difference was detected on the others. The improvements were significant for Average Precision against all three baselines for the Terabyte dataset, against Borda Count and Reciprocal Rank for all of the Legal Track datasets, and against Reciprocal Rank for the Genomics dataset. When considering AUC, the improvement was also significant against all three baselines for the Terabyte dataset, and against Borda Count and Reciprocal Rank for the remaining

datasets. The great improvement by the classifier-based ranker against Borda Fuse in Average Precision for the 2006 subset of the TREC Legal dataset reflects large increases on individual topics. However the classifier-based ranker outperformed the Borda Fuse for only 9 of the 13 topics, which is not enough to allow us to make a claim of significance for that subset.

Thus performance of the classifier-based ranker against three datasets, with *d* increased to 1000, suggests that this method of rank aggregation may have broader applicability. However detecting issues of scale may prove to be more complicated. It is also apparent that Reciprocal Rank Fusion, at least without training to establish correct parameterization, can be a deleterious technique.

6.2 High Depth Analysis

6.2.1 The Comparable Runs

As discussed in Chapter 5, we submitted several experimental runs to the TREC 2008 Legal Track relevance feedback task. Our goal here is to examine the effect of larger pool sizes on the ranking effectiveness of different rank merging methods. We want to compare the performance, using both precision and recall-based measures, of traditional relevance feedback-based retrieval against rank aggregation methods. We wish to explore the interplay between merging runs that are different rankings of the same set of documents, and merging runs that contain different documents. From earlier experiments, we may expect that the former approach will potentially improve precision by promoting documents that are highly ranked and demoting poorly ranked documents. The latter approach would be expected to improve recall by bringing relevant documents retrieved using diverse but effective methods.

The best performance, using the track's Recall-at-*B* measure, was attained by one of our baseline runs created using rCombMNZ on a mixed pool of eight contributors—four each of the runs with the highest and lowest scores in the evaluated retrieval measure

for their respective submission years. The weighted Borda Count and rCombMNZ are differentiated by their weighting mechanisms—both use the sum of a document's ranks over all contributing runs as the underlying basis for the score, but rCombMNZ weights the final score by the number of runs containing the item, whereas the weighted Borda Count uses weights assigned to each contributor to give runs with higher performance greater influence on the final calculated ranking score [62]. As a shorthand, we will refer to this rCombMNZ run as "MNZMix," to reflect that it is a run derived from a combination of multiple runs. Note that MNZMix serves as the "baseline" strategy for our research from this point on. Here we attempt to compare it with the run built from the same set of contributors, but using the classifier-based ranker to order the documents—hereafter identified as "ClassMix." We also include in our comparison a "Traditional" submission created using relevance feedback information to modify an original query (see §5.3).

We also generated additional merged runs that represent higher level combinations—MNZMix merged with ClassMix ("MNZClass"); MNZMix merged with Traditional ("MNZTrad"); and ClassMix merged with Traditional ("ClassTrad"). Each of these merges was accomplished using an unweighted Borda Count. Each of these lists represents a ranking of the same set of documents. We took each of these three merged runs and compared them with each other, and also with the two runs that individually contributed to the merges. With the higher level merges, it is possible that we will see a further improvement in ranking effectiveness as top-ranked documents in different rankings are brought together.

6.2.2 Results

Using the official TREC relevance judgments for the 2008 Legal Relevance Feedback track, we determined $F_1@R$, Precision@5, and Recall@100000 (all estimated) for each of these runs. Estimated $F_1@R$ calculates the estimated F_1-score at the estimated

number of relevant documents. This should result in a measure that balances the recall and precision needs of the legal discovery task. In Table 19, we compare the performance of these merged runs with their component parts, and any other run sharing their component parts. Thus, MNZClass is compared against MNZMix and ClassMix. We also compare MNZClass to MNZTrad to consider the relative value of merging MNZ with either of these other sources. Likewise, we compare MNZClass against ClassTrad. Since the Traditional run itself is not a component of MNZClass, that comparison is not done.

The MNZMix run achieved the highest score (0.2091) with a significant (39.12%, $\alpha \leq 0.05$) improvement over the ClassMix run. Merging the MNZMix with the ClassMix produced a score significantly better (26.41%, $\alpha \leq 0.05$) than the ClassMix. Merging the Traditional run with either of the "Mixes" failed to generate significant changes, and the MNZTrad and ClassTrad scores are nearly identical.

Table 19. Comparison between combined runs and their component parts (column vs row), Est-F_1@R.

Est_R-F1	**(score)**	**ClassTrad**	**MNZTrad**	**ClassMix**	**MNZMix**	**Traditional**
MNZClass	0.1900	-0.0176 / -9.26%	-0.0175 / -9.21%	-0.0397 / -20.89%†	0.0191 / 10.05%	
ClassTrad	0.1724		0.0001 / 0.06%	-0.0221 / -12.82%		-0.0124 / -7.19%
MNZTrad	0.1725				0.0366 / 21.22%	-0.0125 / -7.25%
ClassMix	0.1503				0.0588 / 39.12%†	0.0097 / 6.45%
MNZMix	0.2091					-0.0491 / -23.48%
Traditional	0.1600					

† indicates significant differences ($\alpha \leq 0.05$, two-sided)

Table 20. Comparison between combined runs and their component parts (column vs row), Est-P5.

Est_P5	**(score)**	**ClassTrad**	**MNZTrad**	**ClassMix**	**MNZMix**	**Traditional**
MNZClass	0.5972	-0.1946 / -32.59%	0.1042 / 17.45%	-0.2139 / -35.82%	0.0528 / 8.84%	
ClassTrad	0.4675		0.1992 / 42.61%†	-0.0842 / -18.01%		0.1825 / 39.04%
MNZTrad	0.6667				-0.0167 / -2.50%	-0.0167 / -2.50%
ClassMix	0.3833				0.2667 / 69.60%†	0.2667 / 69.60%
MNZMix	0.6500					0 / 0
Traditional	0.6500					

† indicates significant differences ($\alpha \leq 0.05$, two-sided)

Estimated Precision @ 5 evaluates a method's effectiveness at placing relevant documents at higher ranks. From Table 20, we note that the MNZMix had a significant improvement (69.60%, $\alpha \leq 0.05$) improvement over the ClassMix, again. The Traditional submission had an identical overall score as the MNZMix, but improvements over the ClassMix were not significant. Where merging the traditional run with each of the "mixed" runs reduced the performance difference between the two runs from 69.60% to 42.61%, the improvement of MNZTrad over ClassTrad was still significant ($\alpha \leq 0.05$).

Estimated Recall @ 100000 evaluates how effective each method is at retrieving relevant documents, but doesn't concern itself with the ranking of relevant documents relative to non-relevant ones. As demonstrated in Table 21, MNZMix, ClassMix and MNZClass each consist of different orderings of the same set of documents, so their scores are identical for each of the topics. The Traditional submission registered a significant improvement (13.19%, $\alpha \leq 0.05$) over the Mixed runs. Merging the traditional run with each of MNZMix and ClassMix significantly improved performance over each

Table 21. Comparison between combined runs and their component parts (column vs row), Est-R100000.

Est_R100000	**(score)**	**Class-Trad**	**MNZ-Trad**	**Class-Mix**	**MNZ-Mix**	**Traditional**
MNZClass	0.5451	0.1172 / 21.50%*	0.1207 / 22.04%**	0 / 0	0 / 0	
ClassTrad	0.6622		-0.0064 / -0.97%	-0.1172 / -21.50%‡		-0.0452 / -6.83%
MNZTrad	0.6558				-0.1107 / -16.88%‡	-0.0388 / -5.92%
ClassMix	0.5451				0 / 0	0.0719 / 13.19%†
MNZMix	0.5451					0.0719 / 13.19%†
Traditional	0.6170					

† indicates significant differences ($\alpha \leq 0.05$, two-sided)

‡ indicates significant differences ($\alpha \leq 0.001$, two-sided)

* indicates significant differences ($\alpha \leq 0.0005$, two-sided)

** indicates significant differences ($\alpha \leq 0.0001$, two-sided)

(20.31%, $\alpha \leq 0.001$; and 21.48%, $\alpha \leq 0.001$, respectively). Additionally, MNZTrad and ClassTrad also generated significantly better results than MNZClass (22.04%, $\alpha \leq 0.0001$; and 21.50%, $\alpha \leq 0.0005$, respectively).

This last result suggests that the runs—MNZMix, ClassMix, and Traditional—are sufficiently effective at ranking the relevant documents that merging them will not harm recall by "pushing" relevant documents out of the window of evaluation. This is supported by the fact that "Mix" runs merging *all* available runs achieved the best recall scores in the TREC 2008 Legal Relevance Feedback evaluations. But merging the Traditional run with the classifier-based ranker or with the rCombMNZ results does not necessarily improve precision measures for runs that are already the result of prior merges.

6.2.3 Ranking Measures

In the previous section, we used estimated Precision @ 5 documents as our precision measure. To extend the precision question beyond the initial five documents, we used two ranking measures, Average Precision, and Area Under the ROC Curve (AUC). In Table 22, we compare MNZMix, ClassMix, and MNZClass with the Traditional run by identifying documents that had been evaluated for relevance from each of the runs. In the case of MNZMix, ClassMix, and MNZClass, each represents a reordering of the same documents. For each of these documents, we used the similarity score from the Traditional run to construct the rank ordering of the Traditional run. Using this approach, the Traditional run generated higher scores against the three other methods, but none of these improvements were significant in either Average Precision or AUC.

In summary, we find that constructing these higher level merges allow us to draw conclusions about the effects of different runs on the different information retrieval measures. The Traditional run, which retrieved and ranked a set of documents using a relevance feedback-influenced query, has a different pool of document than the two runs created using different merging strategies applied to the same pool of documents. Thus, adding these distinct document sets increases recall significantly against these runs. This conclusion is consistent with the TREC 2008 relevance feedback track results, where merging large results together generated the highest Recall at 100000 measurements. We also found that merging an outside run will not alter significant differences in Precision at 5, and may eliminate a significant improvement in the balanced F-measure. From an e-discovery perspective, high recall is desirable, especially if a second, precision or rank optimizing procedure can be applied prior to document review.

Table 22. Improvement in performance by the "Traditional" run over three alternatives formed through rank aggregation.

	Traditional		***MNZClass***		***ClassMix***		***MNZMix***	
	AUC	**AP**	**AUC**	**AP**	**AUC**	**AP**	**AUC**	**AP**
Average	0.6548	0.5248	0.6442	0.5089	0.5620	0.4294	0.6819	0.5230
% Improvement			1.65	3.11	16.51	22.22	-3.98	0.33

6.3 Optimizations

Social choice theorists have identified that one of the limitations of the Borda Count is the fact that it is possible to give a better ranking to item *a* over item *b*, even if item *b* is preferred in a majority of contributing ranked lists [32, 47, 62]. This makes "strategic" voting possible when using this rank aggregation method for elections. The classifier-based ranker, too, can produce an aggregated ranking wherein an item with a particularly good ranking in one contributing run can outscore items that outrank it in all other runs.

The weighted Borda Count, the best performing heuristic rank aggregator in our study, attempts to address this by weighting ranking scores by the performance of the contributing run. An extraordinarily high ranking of a document in an underperforming run would be discounted in the aggregate. But a similar performance in a high quality run is, in fact, accentuated by the weighting. As mentioned earlier, using knowledge of contributing run performance—while not as explicit as using knowledge of individual document relevance—is still a form of supervision.

6.3.1 Local Kemenization

An optimal aggregate ranking, from a social preference perspective, will only rank an item higher than another item if it is ranked higher than that item in at least half

of the rankings [25, 100]. The task of finding this optimal ranking has been proven to be NP-hard, largely because each potentially ranked item in the list must be compared against all other items [32]. One approach to addressing this problem is to modify a previously-aggregated run post-hoc to correct for non-optimal rankings. The procedure we implemented, "local Kemenization" is described in [32]. Taking an existing rank aggregation, we start at the tail of the list, and create a new ordering of the same list. We do this by inserting each document at the point when they cease to have a majority of "votes" from the contributing runs over any previously inserted documents.

Dwork, et al., identify properties of local Kemenization that minimize the presence of "spam" web pages in metasearch results. If a "spam" web page is incorporated into the results of less than half of queried search engines, than the assumption is that a majority of these search engines will prefer a real result. This same property is desirable as a result for e-discovery review. We hope to exploit this property to push relevant documents to the top of an aggregated list, while non-relevant documents are pushed into lower "loser" partition.

We applied this method to the MNZMix and ClassMix rankings generated from the TREC 2008 Legal Track relevance feedback task submissions, the data used in the previous section. Note, we do not consider local Kemenization as an alternative to rank aggregation methods, but as a post-aggregation optimizing technique—it requires an initial aggregated ranking. We demonstrate below that the effectiveness of local Kemenization is subject to implementation decisions and the ordering of the initial rank aggregation.

6.3.2 Procedures

Overall, we found that a voting-based optimization may need to be configurable to generate improvements depending on characteristics of both the contributing runs and the merged ranking. Local Kemenization, developed independently as Condorcet-fuse by

other researchers, proceeds in the manner of an InsertionSort [62]. A document in the Kemenized ranking will have a majority, or at least a tie vote with each document that follows it. When it is time to place document *A* in the Kemenized list, we start by comparing it with document *Z*, the document with the lowest ranking in the new list. In order to compare documents *A* and *Z*, we find each of the contributing runs that contain either *A* or *Z*. We assign a point to document *A* for each of these runs which has assigned a higher ranking to *A* than *Z*, or if it contains *A* and not *Z*. Points are assigned to document *Z* in the same manner. After all points are assigned, if document *A* has more points than document *Z*, it "wins" and is then compared with the document that is immediately ranked higher than *Z*. If there are no more comparisons to make, it is inserted at the head of the Kemenized list. If document *A* has less than or the same number of points as document *Z*, it is inserted into the Kemenized list just after document *Z*. In all of our implementations of the Kemenization procedure, a contributing run is considered to have "voted" for a document if it and not the other document in the challenge are included in the run. The potential result of this technique is a Kemenized run that is reordered in such a way as to significantly reduce the influence of the original ranking method. It requires high-ranking documents in the original list to "justify" their position—in order to maintain their original high rankings, items must challenge and defeat each item that originally had a lower ranking.

Applying local Kemenization to the runs we have labeled MNZMix and ClassMix leads to a couple of challenges, which we observed in retrospect. Recall that MNZMix and ClassMix are constructed from four contributing runs that "perform well" and four runs that do not. ClassMix, constructed from the Classifier-Based Ranker, uses supervised learning in the form of a trained classifier to adjust for the quality of the different contributing runs. MNZMix applies run-count multipliers to Borda Count scores in order to calculate new ranking scores for the merged list.

In contrast, Kemenization considers each of the contributing runs to have equal weight. This democratic approach falters when half of the contributors are selected specifically for their poor performance. Consider two documents in the MNZMix run that will be evaluated using the Kemenization procedure. Let us say that each document is included in only one contributing run; document *A* has a high ranking in *run0*, and document *B* has a low ranking in *run1*. When calculating the MNZMix ranking, document *A* benefits from its high ranking in its original run. Likewise, document *B* is handicapped due to its low ranking. Thus, in MNZMix document *A* is ranked higher than document *B*.*

Applying the Kemenization procedure described by Dwork, et al., document *B* is examined first due to its lower ranking in the aggregated ranking. To attain its original ranking relative to document *B*, document *A* must have a higher ranking than *B* in a majority of voting runs. But in this instance, *Run0* includes document *A* and not *B*, and *Run1* includes document *B* and not *A*. The final tally in this challenge is 1-1. As *A* has failed to "justify" its ranking ahead of *B*, it ends up in the new aggregated list behind document *B*. In this way, when there is a tie vote between documents, the local Kemenization procedure obliterates the benefit that MNZMix derives from the original rankings. In this instance, local Kemenization has reversed the signals provided to the MNZMix aggregating method by the original rankings of the documents. It seems that the potential benefit of local Kemenization is limited under any of the following conditions:

* We note that, because the Classifier-Based Ranker is a "black box," we cannot make any inferences as to how it would rank *A* and *B*. It relies on data beyond the ranking of the document in contributing runs, and it generates a different model for each topic based on training data.

- there are few "electors," which both increases the ability of a single contributor to insert items at higher levels of the aggregated list and the possibility of tie votes;
- the contributing ranked lists do not necessarily include the same items—as votes are dispersed against a larger population, the knowledge to be gained from a single vote is minimized;
- and the contributors are not of equal quality, allowing poor quality runs to "subvert" the voting.

Two approaches can be taken to mitigate these consequences of the Local Kemenization postprocessing. One is to reverse the rule concerning tied challenges. In the case of the tie from the previous example, document *A* would be placed ahead of document *B*. The other strategy is to start examining documents for positioning in the new aggregated list from the head of the original list. We evaluated the effect of different Local Kemenization procedures on runs ClassMix and MNZMix, where we tested the effect of reversing the ordering procedure and switching the victory allocation for tied vote counts. Specifically, we tested the following Kemenization-based strategies:

- No Kemenization – the original ranking
- Original Local Kemenization (KEM-L): procedure begins with the tail of the list, and in the case of tied votes, the examined item is assigned a rank behind the item that was first placed in the new list. Thus, this procedure reverses the original ordering within clusters of tied votes.
- Procedure begins with the tail of the list, and tied votes are treated as wins to the item that is being examined (KEM-W). This procedure preserves the original ordering within clusters of tied votes.
- Procedure begins with the head of the list, and tied votes are assigned to the item that is examined first (KEMTOP-L). This procedure preserves the original ordering within clusters of tied votes.

- Procedure begins with the head of the list, and tied votes are assigned to the item that is being examined (KEMTOP-W). This procedure reverses the original ordering within clusters of tied votes.

6.3.3 Results

We note from the results presented in Table 23 that Local Kemenization under any configuration fails to generate significant improvements for the MNZMix merge. In fact, the original Local Kemenization (KEM-L) significantly hurts the performance for MNZMix in both AUC and AP. In contrast, for the classifier-based ranker, Local Kemenization improves performance in AUC by 11.3% and in AP by 17.0% when the procedure begins at the top, and the tie-breaking mechanism is set to preserve the original ordering. When the tie-breaker is set to encourage re-ordering, the improvement in AP reaches 20.5%. Likewise, in this situation, the percentage improvement in AUC reaches 16.1%, though the result is not significant.

We note that in the three cases where Local Kemenization postprocessing improves the ranking effectiveness of the classifier-based ranker, it fails to elevate it past the baseline MNZMix ranking with the same procedure applied. After Kemenization, with the procedure beginning at the tail and the tie-breaking mechanism is set to preserve the original ordering, the classifier-based ranker achieves a significant decline from MNZMix in AUC. The decline is repeated when Kemenization begins with the top of the original ranking. We also note that the improvement of the classifier-based ranker over MNZMix when the original Kemenization procedure is applied, 16.9%, is not significant.

In short, the classifier-based ranker achieves its best performance when Local Kemenization is applied in reverse order, but with a limited and diverse pool of contributing runs, MNZMix would be the rank merging approach that is most effective. Local Kemenization does not improve MNZMix performance, and will likely hurt performance if it is not arranged to minimize reordering from the original ranking.

Table 23. Change in performance after Local Kemenization is applied to ClassMix and MNZMix

					ClassMix vs MNZMix	
	AUC	**Improvement**	**AP**	**Improvement**	**AUC**	**AP**
ClassMix	0.5620		0.4294			
KEM-L	0.5247	-6.6%	0.4013	-6.5%	16.9%	3.8%
KEM-W	0.6167	9.7%	0.4673	8.8%	-5.1%†	-4.5%
KEMTOP-L	0.6255	11.3%†	0.5022	17.0%‡	-8.4%	-6.0%
KEMTOP-W	0.6523	16.1%	0.5174	20.5%†	-1.1%	1.5%
MNZMix	0.6819		0.5230			
KEM-L	0.4490	-34.2%‡	0.3865	-26.1%‡		
KEM-W	0.6500	-4.7%	0.4891	-6.5%		
KEMTOP-L	0.6832	0.2%	0.5341	2.1%		
KEMTOP-W	0.6595	-3.3%	0.5099	-2.5%		

† indicates significant differences ($\alpha \leq 0.05$, two-sided)

‡ indicates significant differences ($\alpha \leq 0.01$, two-sided)

For these experiments, we selected the merges constructed from a smaller pool of contributing runs with a maximum variance of performance. We selected the "diversity" runs based on their performance at TREC 2008, where the runs were evaluated using an F-score that balances recall and precision measures. This limited amount of contributing runs may prevent the Kemenization optimization from having a desired effect. A larger pool of contributing runs may mitigate the distorting effects of outlier runs that share few documents with the other contributors.

The Condorcet literature describes the phenomenon that results in these different orderings, especially when we consider the changes given simply by changing the

Table 24. Change in performance when the Local Kemenization is changed to reverse the ordering within a partition

	AUC	**AP**
ClassMix		
Head-first Placement	-4.1%	-2.9%
Tail-first Placement	17.5%†	16.4%†
MNZMix		
Head-first Placement	3.6%	4.7%
Tail-first Placement	44.8%‡	26.5%‡

† indicates significant differences ($\alpha \leq 0.05$, two-sided)

‡ indicates significant differences ($\alpha \leq 0.001$, two-sided)

ordering of the sort. Local Kemenization ensures that the resulting aggregation meets the extended Condorcet criteria [32, 92]. A locally Kemeny-optimal aggregation will consist of a set of partitions where a partition (A, $\bar{A}$) of $\mathcal{S}$ exists when for any $x \in A$ and an $y \in \bar{A}$, the majority of contributing runs rank x higher than y. All documents in A must then be ranked higher than all documents in $\bar{A}$. No method for ordering items within a partition is prescribed by Condorcet principles. With eight "voters" and hundreds of documents, there are only 24 possible partitions in our Kemenized aggregations.

We can see that modifying the Kemenization procedure alters the handling of partitions in a manner that has a significant impact on overall effectiveness. Adjusting the original Kemenization procedure so that the tie-breaker favors the original ordering results in significant improvements (Table 24). Results were mixed and not significant

Table 25. Change in performance when Local Kemenization changes to place documents from the start, rather than the end, of the original merged list

	AUC	**AP**
ClassMix		
Favors original ordering within partition	24.3%†	28.9%‡
Favors reversing ordering within partition	1.4%	7.5%
MNZMix		
Favors original ordering within partition	46.9%*	31.9%‡
Favors reversing ordering within partition	5.1%	9.2%

† indicates significant differences ($\alpha \leq 0.05$, two-sided)

‡ indicates significant differences ($\alpha \leq 0.01$, two-sided)

* indicates significant differences ($\alpha \leq 0.005$, two-sided)

when making the same switch with the Kemenization procedure beginning at the head of the original ranking. When favoring the original ordering, switching the Kemenization procedure to begin at the head of the list generated significant improvements (Table 25). Improvements were not significant if the Kemenization favored reordering, but the Kemenization procedure was switched to begin at the top of the list.

Aslam and Montague describe this aspect of Kemeny-optimal aggregations as a feature rather than a bug. They suggest that using QuickSort rather than InsertionSort model for inserting documents into the new aggregation creates a probabilistic ordering within the partition. Assigning weights to the "votes" based on the performance of the contributing runs significantly increases the number of possible partitions by creating

many different possible permutations of voting calculations. With such weights, the example of the 1-1 partition described earlier would not be an issue.

If we consider the performance of the other submissions to the TREC 2008 Legal Track relevance feedback task, we should note that submissions compiled from all available and eligible contributing runs generated the best recall performance. Since this reflects the initial goal for e-discovery and other high recall tasks, it may be appropriate for us to consider the effect of these optimizations. We maximize recall to gather as much evidence as is reasonable, and then reorder to improve precision for evaluation. For the purpose of future studies, it may be appropriate to consider the impact of these optimizations on the larger data set.

In this chapter, we sought to investigate the fact that the classifier-based ranker did not retain its advantages in precision as the depth of the data from contributing runs increased from 100 to 25000. We began this with finding the broader applicability of the classifier-based ranker, first by increasing *d* from 100 to 1000, and then applying the same tests to two other large datasets. We determined that, at the larger scale of $d = 25000$, that recall could be improved by merging a distinct source with other previously merged runs. This step had no appreciable improvement in precision measures. We finally determined that use of local Kemenization, carefully implemented, can bring significant improvements to the results generated by the classifier-based ranker.

CHAPTER VII

CONCLUSION

As a high recall task, e-discovery represents a growing concern for large organizations. Requests for production of documents, from the demands of plaintiffs to Freedom of Information Act requests, present great costs to an organization if it fails to provide all pertinent documents. In turn, manually reviewing these documents for relevance is a costly procedure that can benefit from a high-precision ordering of the retrieved documents. This problem is explicitly adopted by the TREC Legal Track, which provides us a framework for evaluating information retrieval tasks that address both the initial retrieval step along with subsequent rerankings to improve precision for document review. We offer our approaches to the two problems of ad hoc retrieval and relevance feedback as defined by the TREC Legal Track framework.

Ad hoc retrieval, the task of retrieving documents to address a particular information need that has been expressed as a query, has long been the focal point of researchers. The information retrieval community has largely sought to find a sample of relevant documents, but e-discovery and other high-recall problems require the retrieval of "all" relevant documents. The legal community, relying on strategies that have been developed for searching citation indexes for laws and decisions that could shape legal arguments, employs keyword searches to seek evidence from enterprise document collections. For our studies, in alignment with the goals of the TREC Legal Track, we sought to improve on the performance of keyword searches by using ranked retrieval, by both retrieving more relevant documents, and by effectively ordering the documents by decreasing likelihood of relevance.

We have identified a series of highly effective strategies that can be applied to ad hoc retrieval conducted across large collections of electronically stored information. Using the Boolean keyword queries as a starting point, we stripped the queries of most of

their operators, and then expanded our set of query terms by expanding wildcards from the original Boolean query. We found that accepting all possible expansions diluted the effect of other terms in the queries as OCR error generated large quantities of variants of the same term. In order to mitigate this effect, we determined that using the two candidate terms contained in the most documents in the collection generated the best results. Adding terms from the narrative request for the production of documents improved performance. We found that we could further improve our results if we systematically removed terms from these requests if they did not apply specifically to the topic (e.g., "please", "document"). We determined that these terms, often present due to standard legal protocols, could be identified by finding the ones that appeared in 25% of the requests across all of the topics. We also found improvement by employing Okapi-BM25, a term-weighting formula that can be applied to the vector space model. BM25, which accounts for document length, is a practical mechanism for handling the wide range of document lengths present in an enterprise collection of electronically stored information.

We also improved our ad hoc results by employing pseudo-relevance feedback. We implement this by gathering the top three documents from an initial retrieval, and then selecting the five terms that appear most frequently in these documents and also meet several filtering criteria. These five terms are added to the initial query, for significant improvements. Finally, we utilize the Boolean queries one final time by identifying the intersection of documents retrieved by our ad hoc strategies, and the documents retrieved by a system implementing the lawyer-negotiated Boolean queries. We found that we could significantly improve the Mean Average Precision of our system by "boosting" these intersecting documents within our own rankings.

We also note areas where we did not generate significant improvements, and they are potential avenues for future research. The documents in the TREC Legal Track collection were digitized and converted into machine-readable text using optical character

recognition. The level of error from this process results in varying levels of noise. We attempt to build a profile of each document to estimate the level of distortion introduced to the text. Attempts to modify the ordering of our ranked results using these profiles did not generate improvements, though we believe that this information may be useful in characterizing the documents in future data mining experiments. Associated with each document in the TREC Legal Track collection are non-content metadata fields—inconsistently applied and generated—which provide information about the creation, distribution and post-settlement disposal of each document. We did not generate any improvements by directly querying these fields as part of our ad hoc retrieval strategies, though we feel that future research can mine this rich information for relevance feedback purposes. Additional research may approach the possibility of constructing social networking models to identify document traffic patterns within organizations.

"Traditional" relevance feedback is often associated with the procedure of using information from documents that are known to be relevant in order to modify the query that retrieved the documents in the first place. We interpret relevance feedback data as signals within ranking information collected from multiple information retrieval systems that may point to other relevant documents. In our experiments, we use this information to train a classifier to rank unjudged documents from multiple ad hoc retrieval systems. Documents, both judged and unjudged, from each of contributing system are combined in a pool. We use the ranking information for the documents in the pool to calculate a series of features that describe each document in terms of its position relative to documents known to be relevant or non-relevant. The profiles of the judged documents are used to train an SMO-based classifier. The classifier is then used to estimate the likelihood of relevance of the remaining unjudged documents. We identify the parameter d to specify the number of documents added to the pool by each system.

We conducted a pilot experiment where we set $d = 100$ and used different combinations of contributing systems to test the classifier-based ranker against two

variants of the heuristic Borda Count method. We found that using all available contributing systems led to the most consistently significant improvements over the baseline heuristic methods. Using smaller subsets of contributing systems, we found that significant improvements could be subject to the quality of the different systems. In particular, we noted that using contributing systems representing a mix of high precision and low precision results presented an environment where the classifier-based ranker would consistently outperform the Borda Count methods. In contrast, using a similar number of the best performing systems resulted in barely any improvement over the heuristics.

The classifier-based ranker itself is a complicated system with several configuration options, each of which opens up further venues for research. The SMO classifier is built using default parameters, each of which should be explored for potential performance improvement. Further, the SMO classifier itself could be replaced with any of several alternative classifiers or ranking methods based on supervised learning. Additionally, the features used to model documents are dependent on the availability of sufficient relevance judgments in close proximity within the ranking information. Sparse judgments reduce the ability of these features to create distinct profiles of documents; this is particularly evident at lower ranks when *d* is set to larger values. Creation of features that are not dependent on the context of the document within ranked retrieval result sets may improve performance for higher values of *d*.

Increasing *d* to 25000 enabled us to test the effectiveness of the classifier-based ranker at the high recall task. We found that our heuristic baseline method, which does not rely on relevance feedback to determine rankings, outperformed the classifier-based ranker. Both rank aggregation methods, which employed ranking information from all available contributing systems when building document pools, proved to be more effective at overall recall than traditional relevance feedback-based query expansion strategies.

We sought to assess the effectiveness of the classifier-based ranker with values of *d* increased to more than 100 but less than 25,000 documents. We further wished to establish the robustness of the system by testing it against datasets other than the one established by the organizers of the TREC Legal Track. Using measures specifically designed to assess ranking effectiveness, we increased *d* to 1000 and we found significant improvements by the classifier-based ranker over Borda Count-based heuristics when tested against an expanded TREC Legal dataset and against a dataset based on the GOV2 collection of web-based documents. However, we found no significant difference between the methods when applied to a dataset based on medical literature abstracts.

We built higher order merges by merging traditional relevance feedback results with merged rankings generated using both the classifier-based ranker and the rCombMNZ heuristic. This baseline is similar to Borda Count, except that the actual Borda Count score for a given document is multiplied by the number of contributing systems that actually retrieved the document. Consistent with our TREC based results, we found that merging rankings containing distinct sets of documents improved recall, but did not significantly affect precision. We also found that the traditional relevance feedback method did not significantly outperform the rank aggregation methods using any of our ranking-based measures.

We concluded our research with an examination of an optimization method with roots in social choice theory. Local Kemenization reflects the principle that, in a merged ranking, no document should be ranked higher than another document that is ranked higher than it in a majority of contributing ranked lists. In general, we found that this method could generate significant improvements to the classifier-based ranker, and significant declines in performance to the rCombMNZ heuristic. We found that the effectiveness of this strategy was subject to implementation details, particularly because the large number of documents under consideration combined with the relatively small number of aggregated rankings results in several “tie votes” between document pairings.

In certain implementations, applying Local Kemenization reduces the performance gap between the classifier-based ranker and rCombMNZ. Future research on the effectiveness of voting-based optimizations would center on different subsets used in document merges, as having more "voters" may reduce the size of partitions of "tied votes."

Our research on e-discovery and high recall information retrieval reflects the evaluation and development of a broad range of retrieval strategies. We further complement this approach with the development of a merging strategy that relies on training a classifier to exploit signals found in the ranking information from the results of diverse systems. The range of opportunities for continued research in these areas is expansive, with the potential for greater improvements in both recall for large scale searches, and precision for subsequent document review.

APPENDIX

PERFORMANCE MEASURES

AT VARIOUS DEPTHS,

D = 100 EXPERIMENTS

Table A-1. Recall and precision measures assessed at depth 10 for each subset of contributing runs

Contributing Runs	Precision			Recall		
	CBR	WBC	BC	CBR	WBC	BC
All60	0.6581	0.5581‡	0.5233‡	0.1292	0.1008‡	0.1003‡
MixedRuns	0.6667	0.5762‡	0.5357‡	0.1911	0.1859	0.1660
Top30Prec	0.6465	0.5419‡	0.5256‡	0.1570	0.1252‡	0.1122†
IowaS	0.5667	0.4429‡	0.4524‡	0.2765	0.1877‡	0.2022‡
CMU	0.5698	0.4884‡	0.4651‡	0.2203	0.1647‡	0.1628‡
UMass	0.6209	0.5395‡	0.5140‡	0.2858	0.2671	0.2070‡
Waterloo	0.5628	0.4953‡	0.4907‡	0.2399	0.2020‡	0.1825†
Fudan	0.5500	0.5452	0.4952	0.3502	0.3573	0.3140
UMKC	0.5833	0.5643	0.5143‡	0.3193	0.3192	0.2812
Ursinus	0.5683	0.5390	0.4780‡	0.2620	0.3338**	0.2844
IowaE	0.5500	0.5400	0.5050	0.4090	0.4500	0.4167
Top7Prec	0.5833	0.5405	0.5357	0.1691	0.1419	0.1494
Top7Recall	0.5810	0.5571	0.5429	0.1760	0.1651	0.1754
OpenText	0.5405	0.5452	0.5524	0.1636	0.1692	0.1831
Sabir	0.5122	0.5146	0.4805	0.4255	0.4378	0.3929

† indicates that the classifier-based ranker outperformed the specified baseline at $\alpha \leq 0.10$ using the randomization test.

‡ indicates that the classifying ranker outperformed the specified baseline at $\alpha \leq 0.05$ using the randomization test.

** indicates that the classifying ranker underperformed the specified baseline at $\alpha \leq 0.05$ using the randomization test.

Table A-2. Recall and precision measures assessed at depth 15 for each subset of contributing runs

Contributing Runs	Precision			Recall		
	CBR	WBC	BC	CBR	WBC	BC
All60	0.6434	0.5535‡	0.5209‡	0.1836	0.1482‡	0.1430‡
MixedRuns	0.6651	0.5698‡	0.5302‡	0.2820	0.2624	0.2368†
Top30Prec	0.6372	0.5442‡	0.5225‡	0.2229	0.1785‡	0.1666‡
IowaS	0.5460	0.4651‡	0.4460‡	0.3620	0.2774‡	0.2846‡
CMU	0.5535	0.4915‡	0.4729‡	0.3084	0.2489‡	0.2333‡
UMass	0.5876	0.5318†	0.5008‡	0.3906	0.3641	0.2922‡
Waterloo	0.5426	0.4961†	0.4915†	0.3144	0.2919	0.2722
Fudan	0.5349	0.5048	0.4698‡	0.4721	0.4617	0.4039
UMKC	0.5571	0.5429	0.4921‡	0.4200	0.4219	0.3686
Ursinus	0.5415	0.5122	0.4618‡	0.3630	0.4546**	0.3696
IowaE	0.5117	0.5017	0.4783	0.5112	0.5780*	0.5421
Top7Prec	0.5810	0.5540	0.5286	0.2411	0.2197	0.2162
Top7Recall	0.5698	0.5508	0.5524	0.2529	0.2500	0.2599
OpenText	0.5492	0.5492	0.5460	0.2484	0.2567	0.2655
Sabir	0.4813	0.4748	0.4553	0.5359	0.5688	0.5215

† indicates that the classifier-based ranker outperformed the specified baseline at $\alpha \leq 0.10$ using the randomization test.

‡ indicates that the classifying ranker outperformed the specified baseline at $\alpha \leq 0.05$ using the randomization test.

* indicates that the classifying ranker underperformed the specified baseline at $\alpha \leq 0.10$ using the randomization test.

** indicates that the classifying ranker underperformed the specified baseline at $\alpha \leq 0.05$ using the randomization test.

Table A-3. Recall and precision measures assessed at depth 20 for each subset of contributing runs

Contributing Runs	Precision			Recall		
	CBR	WBC	BC	CBR	WBC	BC
All60	0.6326	0.5442‡	0.4919‡	0.2324	0.1875‡	0.1716‡
MixedRuns	0.6417	0.5571‡	0.5060‡	0.3502	0.3253	0.2915‡
Top30Prec	0.6105	0.5372‡	0.5105‡	0.2773	0.2297‡	0.2132‡
IowaS	0.5131	0.4714†	0.4488‡	0.4355	0.3767	0.4124
CMU	0.5326	0.4884‡	0.4558	0.3627	0.3520	0.3006
UMass	0.5674	0.5128‡	0.4872‡	0.4709	0.4310	0.3629‡
Waterloo	0.5256	0.5058	0.4837	0.3837	0.3803	0.3561
Fudan	0.5214	0.4869	0.4464‡	0.5837	0.5619	0.4792‡
UMKC	0.5345	0.5179	0.4750‡	0.4933	0.5121	0.4626
Ursinus	0.5268	0.4866	0.4427‡	0.4789	0.5165	0.4266
IowaE	0.4850	0.4700	0.4425†	0.6133	0.6759	0.6224
Top7Prec	0.5655	0.5500	0.5369	0.3138	0.3009	0.3003
Top7Recall	0.5631	0.5464	0.5357	0.3202	0.3404	0.3269
OpenText	0.5310	0.5488	0.5286	0.3160	0.3490	0.3386
Sabir	0.4512	0.4378	0.4244	0.6120	0.6474	0.6225

† indicates that the classifier-based ranker outperformed the specified baseline at $\alpha \leq 0.10$ using the randomization test.

‡ indicates that the classifying ranker outperformed the specified baseline at $\alpha \leq 0.05$ using the randomization test.

Table A-4. Recall and precision measures assessed at depth 25 for each subset of contributing runs

Contributing Runs	Precision			Recall		
	CBR	WBC	BC	CBR	WBC	BC
All60	0.6186	0.5274‡	0.4837‡	0.2803	0.2287‡	0.2072‡
MixedRuns	0.6162	0.5390‡	0.4962‡	0.4134	0.3909	0.3449‡
Top30Prec	0.5888	0.5274‡	0.4967‡	0.3221	0.2907†	0.2509‡
IowaS	0.4933	0.4733	0.4467‡	0.5306	0.4855	0.4612†
CMU	0.5181	0.4763†	0.4391‡	0.4331	0.4089	0.3573†
UMass	0.5553	0.4958‡	0.4549‡	0.5503	0.5148	0.4332‡
Waterloo	0.5060	0.4986	0.4753	0.4418	0.4638	0.4309
Fudan	0.4990	0.4629‡	0.4286‡	0.6658	0.6426	0.5715‡
UMKC	0.5200	0.5048	0.4552‡	0.5835	0.6285	0.5348
Ursinus	0.5034	0.4751	0.4332‡	0.5375	0.5958*	0.5170
IowaE	0.4580	0.4420	0.4170†	0.6883	0.7665**	0.7124
Top7Prec	0.5562	0.5362	0.5067†	0.3752	0.3656	0.3599
Top7Recall	0.5543	0.5362	0.5210	0.3843	0.4177	0.3884
OpenText	0.5229	0.5352	0.5067	0.3871	0.4257*	0.3971
Sabir	0.4273	0.4234	0.4098	0.6817	0.7343	0.7154

† indicates that the classifier-based ranker outperformed the specified baseline at $\alpha \leq 0.10$ using the randomization test.

‡ indicates that the classifying ranker outperformed the specified baseline at $\alpha \leq 0.05$ using the randomization test.

* indicates that the classifying ranker underperformed the specified baseline at $\alpha \leq 0.10$ using the randomization test.

** indicates that the classifying ranker underperformed the specified baseline at $\alpha \leq 0.05$ using the randomization test.

REFERENCES

[1] Almquist, B., Ha-Thuc, V., Sehgal, A. K., Arens, R. and Srinivasan, P. Exploring the Legal Discovery and Enterprise Tracks at the University of Iowa. In Voorhees, E. M. and Buckland, L. P. eds. *The Sixteenth Text REtrieval Conference Proceedings (TREC 2007).* (Gaithersburg, MD, November 2007). National Institute of Standards and Technology (NIST) the Defense Advanced Research Projects Agency (DARPA) and the Advanced Research and Development Activity (ARDA), 2007.

[2] Almquist, B., Mejova, Y., Ha-Thuc, V. and Srinivasan, P. University of Iowa at TREC 2008 Legal and Relevance Feedback Tracks. In Voorhees, E. M. and Buckland, L. P. eds. *The Seventeenth Text REtrieval Conference Proceedings (TREC 2008).* (Gaithersburg, MD, November 18-21, 2008). National Institute of Standards and Technology (NIST) the Defense Advanced Research Projects Agency (DARPA) and the Advanced Research and Development Activity (ARDA), 2008.

[3] Ashley, K. and Bridewell, W. Emerging AI & Law approaches to automating analysis and retrieval of electronically stored information in discovery proceedings. Artificial Intelligence and Law, 18, 4 (2010), 311-320. DOI=10.1007/s10506-010-9098-4.

[4] Aslam, J. A. and Montague, M. Models for metasearch. In *SIGIR '01: Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval.* (New Orleans, Louisiana, September 9-13, 2001). ACM, New York, NY, USA, 2001, 276-284.

[5] Ataman, K., Street, W. N. and Zhang, Y. Learning to Rank by Maximizing AUC with Linear Programming. In *IJCNN '06. International Joint Conference on Neural Networks.* 2006, 123-129.

[6] Bagley, P. R. *Electronic digital machines for high-speed information searching.* M.S. Thesis, Massachusetts Institute of Technology, Boston, Mass., 1951.

[7] Barnett, T., Godjevac, S., Renders, J., Privault, C., Schneider, J. and Wickstrom, R. Machine Learning Classification for Document Review. In *Proceedings of the Global E-Discovery/E-Disclosure Workshop on Electronically Stored Information in Discovery at the 12th International Conference on Artificial Intelligence and Law (ICAIL09 DESI Workshop).* (Barcelona, Spain,). 2009.

[8] Baron, J. R. and Thompson, P. The search problem posed by large heterogenous datasets in litigations: possible future approaches to research. In *Proceedings of the 11th International Conference on Artificial Intelligence and Law.* (Stanford, CA, 2007). ACM, New York, NY, 2007, 141-147.

[9] Baron, J. R., Lewis, D. D. and Oard, D. W. TREC 2006 Legal Track Overview. In Voorhees, E. M. and Buckland, L. P. eds. *The Fifteenth Text REtrieval Conference Proceedings (TREC 2006).* (Gaithersburg, MD, November 14-17, 2006). National Institute of Standards and Technology (NIST), the Defense Advanced Research Projects Agency (DARPA) and the Advanced Research and Development Activity (ARDA), Gaithersburg, MD, 2006, 79-98.

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

[11] Bookstein, A. Information retrieval: a sequential learning process. *Journal of the American Society for Information Science* 34, 5 (September 1983), 331-342.

[12] Borda, J. C. *Mémoire sur les élections au scrutin.* Histoire de l'académie royale des sciences, 1781.

[13] Bradley, A. P. The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognition* 30, 7 (July 1997), 1145-1159. DOI: 10.1016/S0031-3203(96)00142-2.

[14] Brefeld, U. and Scheffer, T. AUC maximizing support vector learning. In *Proceedings of the ICML Workshop on ROC Analysis in Machine Learning.* (Bonn, Germany, August 11, 2005). 2005.

[15] Buckley, C. Examining Overfitting in Relevance Feedback: Sabir Research at TREC 2007. In Voorhees, E. M. and Buckland, L. P. eds. *The Sixteenth Text REtrieval Conference Proceedings (TREC 2007).* (Gaithersburg, MD, November 2007). National Institute of Standards and Technology (NIST) the Defense Advanced Research Projects Agency (DARPA) and the Advanced Research and Development Activity (ARDA), 2007.

[16] Buckley, C., Salton, G. and Allan, J. The effect of adding relevance information in a relevance feedback environment. In *SIGIR '94: Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval.* (Dublin, Ireland, July 3-6, 1994). Springer-Verlag New York, Inc., New York, NY, USA, 1994, 292-300.

[17] Bush, V. As We May Think. *The Atlantic Monthly* 176, 1 (July 1945), 101-108.

[18] Büttcher, S., Clarke, C. L. A., Cormack, G. V. and Lynam, T. R. MultiText Legal Experiments at TREC 2007. In Voorhees, E. M. and Buckland, L. P. eds. *The Sixteenth Text REtrieval Conference Proceedings (TREC 2007).* (Gaithersburg, MD, November 2007). National Institute of Standards and Technology (NIST), the Defense Advanced Research Projects Agency (DARPA), and the Advanced Research and Development Activity (ARDA), 2007.

[19] Büttcher, S., Clarke, C. and Soboroff, I. The TREC 2006 Terabyte Track. In Voorhees, E. M. and Buckland, L. P. eds. *The Fifteenth Text REtrieval Conference Proceedings (TREC 2007).* (Gaithersburg, MD, November 14-17, 2007). National Institute of Standards and Technology (NIST), the Defense Advanced Research Projects Agency (DARPA), and the Advanced Research and Development Activity (ARDA), 2007.

[20] Buckley, C., Salton, G., Allan, J., Singhal, A. Automatic Query expansion using SMART: TREC 3. In Harman, D. K., ed. *Overview of the Third Text REtrieval Conference (TREC-3).* (Gaithersburg, MD, November 2-4, 1994). The Department of Commerce and the National Institute of Standards and Technology (NIST), 1994.

[21] Caruana, R. and Mizil, A. An empirical comparison of supervised learning algorithms. In *ICML '06: Proceedings of the 23rd International Conference on Machine Learning.* Pittsburgh, Pennsylvania, 2006. ACM, 161-168.

[22] Cleverdon, C. W. *ASLIB Cranfield research project: report on the testing and analysis of an investigation into the comparative efficiency of indexing systems.* College of Aeronautics, Cranfield, Cranfield, 1962.

[23] Cleverdon, C. The Cranfield Tests on Index Language Devices. *ASLIB Proceedings* 19, 6 (1967), 173. DOI=10.1108/eb050097.

[24] Collins-Thompson, K. and Callan, J. Query expansion using random walk models. In *CIKM '05: Proceedings of the 14th ACM International Conference on Information and Knowledge Management.* (Bremen, Germany, October 31-November 5, 2005). ACM, New York, NY, USA, 2005, 704-711.

[25] Condorcet, M. -. *Essai sur l'Application de l'Analyse à la Probabilité des Décisions Rendues à la Pluralité des Voix.* L'Imprimerie Royale, Paris, 1785.

[26] Conrad, J. E-Discovery revisited: the need for artificial intelligence beyond information retrieval. Artificial Intelligence and Law, 18, 4 (2010), 321-345. DOI=10.1007/s10506-010-9096-6.

[27] Cormack, G., Clarke, C. and Buettcher, S. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *SIGIR '09: Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval.* (Boston, Mass., July 19-23, 2009). ACM, 2009, 758-759.

[28] Cristianini, N. and Shawe-Taylor, J. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods.* Cambridge University Press, Cambridge, UK, 2000.

[29] Croft, W. B. *Advances in Information Retrieval: Recent Research from the Center for Intelligent Information Retrieval.* Kluwer Academic Publishers, Dordrecht, 2000.

[30] Culotta, A., Liu, A., Cordover, M., Borden, B. and Strickland, S. IT-Discovery at TREC 2010 Legal. In Voorhees, E. M. and Buckland, L. P. eds. *The Nineteenth Text REtrieval Conference Proceedings (TREC 2010).* (Gaithersburg, MD, November 16-19, 2010). National Institute of Standards and Technology (NIST) the Defense Advanced Research Projects Agency (DARPA) and the Advanced Research and Development Activity (ARDA), 2010.

[31] Dunham, M. H. *Data mining introductory and advanced topics / Margaret H. Dunham.* Prentice Hall/Pearson Education, Upper Saddle River, N.J. :, 2003.

[32] Dwork, C., Kumar, R., Naor, M. and Sivakumar, D. Rank aggregation methods for the Web. In *WWW '01: Proceedings of the 10th International Conference on World Wide Web.* (Hong Kong, May 1-5, 2001). ACM, 613-622.

[33] Efthimiadis, E. N. Query expansion. *Annual Review of Information Systems and Technology* 31(1996), 121-187.

[34] Eichmann, D. and Chin, S. Concepts, semantics and syntax in e-discovery. In *ICAIL 2007 Workshop on Supporting Search and Sensemaking for Electronically Stored Information in Discovery Proceedings (DESI Workshop).* (Palo Alto, CA, June 4, 2007). 2007.

[35] Fellbaum, C. *WordNet: An Electronic Lexical Database (Language, Speech, and Communication).* The MIT Press, Cambridge, MA, 1998.

[36] Freund, Y., Iyer, R., Schapire, R. E. and Singer, Y. An efficient boosting algorithm for combining preferences. *Journal of Machine Learning Research* 4 (2003), 933-969.

[37] Garron, A. and Kontostathis, A. Applying Latent Semantic Indexing on the TREC 2010 Legal Dataset. In Voorhees, E. M. and Buckland, L. P. eds. *The Nineteenth Text REtrieval Conference Proceedings (TREC 2010).* (Gaithersburg, MD, November 16-19, 2010). National Institute of Standards and Technology (NIST) the Defense Advanced Research Projects Agency (DARPA) and the Advanced Research and Development Activity (ARDA), 2010.

[38] Greenwood, A. Attorney at blah. *Washington City Paper* (Nov. 7, 2007), 18-28.

[39] Harman, D. Overview of the first TREC conference. In *Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval.* (Pittsburgh, Pennsylvania, United States). ACM, New York, NY, USA, 1993, 36-47.

[40] Harman, D. Relevance feedback revisited. In *SIGIR '92: Proceedings of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval.* (Copenhagen, Denmark, June 21-24, 1992). ACM, New York, NY, USA, 1992, 1-10.

[41] Hawking, D. and Robertson, S. On Collection Size and Retrieval Effectiveness. *Information Retrieval* 6, 1 (01/01 2003), 99-105.

[42] Hersh, W., Cohen, A., Yang, J., Bhupatiraju, R., Roberts, P. and Hearst, M. TREC 2005 genomics track overview. In Voorhees, E. M. and Buckland, L. P. eds. *The Fourteenth Text REtrieval Conference Proceedings (TREC 2010).* (Gaithersburg, MD, November 15-18, 2005). National Institute of Standards and Technology (NIST) the Defense Advanced Research Projects Agency (DARPA) and the Advanced Research and Development Activity (ARDA), 2005**.**

[43] Hodge, G. M. and Milstead, J. L. *Computer Support to Indexing.* National Federation of Abstracting and Information Services, Philadelphia, Penn., 1998.

[44] Hogan, C., Bauer, R. and Brassil, D. Automation of legal sensemaking in e-discovery. *Artificial Intelligence and Law*, 18, 4 (2010), 431-457. DOI=10.1007/s10506-010-9100-1.

[45] Jones, R. and Fain, D. C. Query word deletion prediction. In *SIGIR '03: Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval.* (Toronto, Canada, July 28-August 1, 2003). ACM Press, New York, NY, USA, 2003, 435-436.

[46] Kantor, P. B. and Voorhees, E. M. The TREC-5 confusion track: comparing retrieval methods for scanned text. *Information Retrieval* 2, 2 (May 2000), 165-176.

[47] Kemeny, J. Mathematics without Numbers. *Daedalus* 88, 4 (1959).

[48] Kluck, M. and Gey, F. The Domain-Specific Task of CLEF - Specific Evaluation Strategies in Cross-Language Information Retrieval. In Peters, C. ed., *Cross-Language Information Retrieval and Evaluation.* Springer Berlin Heidelberg, , 2001, 48-56.

[49] Kotsiantis, S. B. Supervised Machine Learning: A Review of Classification Techniques. *Informatica* 31, 3 (2007).

[50] Kulp, S. and Kontostathis, A. On Retrieving Legal Files: Shortening Documents and Weeding Out Garbage. In Voorhees, E. M. and Buckland, L. P. eds. *The Sixteenth Text REtrieval Conference Proceedings (TREC 2007).* (Gaithersburg, MD, November 2007). National Institute of Standards and Technology (NIST), the Defense Advanced Research Projects Agency (DARPA), and the Advanced Research and Development Activity (ARDA), 2007.

[51] Lancaster, F. W. and Mills, J. Testing indexes and index language devices: The ASLIB cranfield project. *American Documentation* 15, 1 (1964), 4-13. DOI=10.1002/asi.5090150104.

[52] Legacy Tobacco Documents Library, http://legacy.library.ucsf.edu, UCSF Library and Center for Knowledge Management and the American Legacy Foundation. Last Updated: 2007 (Accessed: 2008 Dec. 4).

[53] Limbu, D. K., Connor, A., Pears, R. and Macdonell, S. Contextual relevance feedback in web information retrieval. In *IIiX: Proceedings of the 1st International Conference on Information Interaction in Context.* (Copenhagen, Denmark, October 18-20, 2006). ACM, New York, NY, USA, 2006, 138-143.

[54] Lin, J. and Wilbur, W. J. PubMed related articles: a probabilistic topic-based model for content similarity. *BMC Bioinformatics* 8, 1 (2007), 423. DOI=10.1186/1471-2105-8-423.

[55] Liu, Y., Liu, T., Qin, T., Ma, Z. and Li, H. Supervised rank aggregation. In *WWW '07: Proceedings of the 16th International Conference on World Wide Web.* (Banff, Canada, May 8-12). ACM Press, New York, NY, USA, 2007, 481-490.

[56] Lucene Java, 2.2, http://lucene.apache.org/java/docs/index.html, The Apache Software Foundation (2007).

[57] Manmatha, R., Rath, T. and Feng, F. Modeling score distributions for combining the outputs of search engines. In *SIGIR '01: Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval.* (New Orleans, Louisiana, September 9-13, 2001). ACM Press, New York, NY, USA, 2001, 267-275.

[58] Manning, C. D., Raghavan, P. and Schütze, H. *Introduction to Information Retrieval.* Cambridge University Press, Cambridge, UK, 2008.

[59] Maron, M. E. Probabilistic Retrieval Models. In Dervin, B. and Voigt, M. J. eds., *Progress in Communication Sciences.* Ablex Publishing, Norwood, NJ., 1984, 145.

[60] Maron, M. E. and Kuhns, J. L. On relevance, probabilistic indexing and information retrieval. *Journal of the ACM* 7, 3 (July 1960), 216-244.

[61] Mitra, M., Singhal, A. and Buckley, C. Improving automatic query expansion. In *SIGIR '98: Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval.* (Melbourne, Australia, August 24-28, 1998). ACM Press, New York, NY, USA, 1998, 206-214.

[62] Montague, M. and Aslam, J. A. Condorcet fusion for improved retrieval. In *CIKM '02: Proceedings of the Eleventh International Conference on Information and Knowledge Management.* (McLean, Virginia, November 4-9, 2002). ACM, New York, NY, USA, 2002, 538-548.

[63] Oard, D. W., Hedin, B., Tomlinson, S. and Baron, J. R. Overview of the TREC 2008 Legal Track. In Voorhees, E. M. and Buckland, L. P. eds. *The Seventeenth Text REtrieval Conference Proceedings (TREC 2008).* (Gaithersburg, MD, November). National Institute of Standards and Technology (NIST) the Defense Advanced Research Projects Agency (DARPA) and the Advanced Research and Development Activity (ARDA), 2008.

[64] Platt, J. C. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In Bartlett, P. J., Schölkopf, B., Schuurmans, D. and Smola, A. J. eds., *Advances in Large-Margin Classifiers.* The MIT Press, Cambridge, Massachusetts, 2000, 61-74.

[65] Privault, C., O'Neill, J., Ciriza, V. and Renders, J. A new tangible user interface for machine learning document review. *Artificial Intelligence and Law*, 18, 4 (2010), 459-479. DOI=10.1007/s10506-010-9090-z.

[66] Renda, E. and Straccia, U. Web metasearch: rank vs. score based rank aggregation methods. In *SAC '03: Proceedings of the 2003 ACM Symposium on Applied Computing.* (Melbourne, Florida, March 9-12, 2003). ACM Press, 2003, 841-846.

[67] Ro, J. S. An evaluation of the applicability of ranking algorithms to improve the effectiveness of full-text retrieval. I. On the effectiveness of full-text retrieval. *Journal of the American Society for Information Science* 39, 2 (1988), 73-78.

[68] Robertson, S. E. The probability ranking principle in IR. *Journal of Documentation* 33, 4 (Dec. 1977), 294-304.

[69] Robertson, S. E., Maron, M. E. and Cooper, W. S. Probability of relevance: a unification of two competing models for document retrieval. *Information Technology: Research and Development* 1, 1 (Jan. 1982), 1-21.

[70] Robertson, S. E. On term selection for query expansion. *Journal of Documentation* 46, 4 (Dec. 1990), 359-364.

[71] Robertson, S. E., Walker, S., Hancock-Beaulieu, M., Gull, A. and Lau, M. Okapi at TREC. In Harman, D. K., ed. *The First Text REtrieval Conference (TREC-1).* (Gaithersburg, MD, November 4-6, 1992). The Department of Commerce and the National Institute of Standards and Technology (NIST), 1992, 21-30.

[72] Rocchio, J. J. Relevance feedback in information retrieval. In Salton, G. ed., *The SMART Retrieval System: Experiments in Automatic Document Processing.* Prentice Hall, Englewood Cliffs, New Jersey, 1971.

[73] Roitblat, H. L., Kershaw, A. and Oot, P. Document categorization in legal electronic discovery: computer classification vs. manual review. *Journal of the American Society for Information Science and Technology (JASIST)* 61, 1 (2010), 70-80. DOI=10.1002/asi.21233.

[74] Salton, G., Wong, A. and Yang, C. S. A vector space model for automatic indexing. *Communications of the ACM* 18, 11 (November 1975), 613-620.

[75] Salton, G. Automatic text analysis. *Science* 168, 3929 (April 1970), 335-343.

[76] Salton, G. *The SMART Retrieval System: Experiments in Automatic Document Processing.* Prentice Hall, Englewood Cliffs, New Jersey, 1971.

[77] Salton, G. and Buckley, C. Term-weighting approaches in automatic text retrieval. *Information Processing and Management* 24, 5 (1988), 513-523.

[78] Shaw, J. A. and Fox, E. A. Combination of multiple searches. In Harman, D. K., ed. *The Second Text REtrieval Conference (TREC-2).* (Gaithersburg, MD, August 31-September 2, 1993). The Department of Commerce and the National Institute of Standards and Technology (NIST), 1993, 243-252.

[79] Smucker, M. D., Allan, J. and Carterette, B. A comparison of statistical significance tests for information retrieval evaluation. In *CIKM '07: Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management.* (Lisbon, Portugal, November 6-10, 2007). ACM, New York, NY, USA, 2007, 623-632.

[80] Spärck Jones, K. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation* 28, 1 (1972), 11-20.

[81] Swanson, D. R. Searching natural language text by computer. *Science* 132, 3434 (October 21, 1960), 1099-1104.

[82] Taghva, K., Borsack, J., Condit, A. and Erva, S. The effects of noisy data on text retrieval. *Journal of the American Society for Information Science* 45, 1 (January 1994), 50-58.

[83] Taghva, K., Borsack, J. and Condit, A. Effects of OCR errors on ranking and feedback using the vector space model. *Information Processing and Management* 32, 3 (May 1996), 317-327.

[84] Text REtrieval Conference (TREC) Overview, *Text REtrieval Conference (TREC),* http://trec.nist.gov/overview.html, National Institute for Standards and Technology. Last Updated: 2008 Aug. 28 (Accessed: 2008 Dec. 2).

[85] The Lemur Project. The Lemur Toolkit, http://www.lemurproject.org/.

[86] Theobald, M., Schenkel, R. and Weikum, G. Efficient and self-tuning incremental query expansion for top-k query processing. In *SIGIR '05: Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval.* (Salvador, Brazil, 15-19 August 2005). ACM, New York, NY, USA, 2005, 242-249.

[87] Thompson, P. Looking back: On relevance, probabilistic indexing and information retrieval. *Information Processing & Management* 44, 2 (March 2008), 963-970.

[88] Tomlinson, S. Experiments with the Negotiated Boolean Queries of the TREC 2007 Legal Discovery Track. In Voorhees, E. M. and Buckland, L. P. eds. *The Sixteenth Text REtrieval Conference Proceedings (TREC 2007).* (Gaithersburg, MD, November 2007). National Institute of Standards and Technology (NIST) the Defense Advanced Research Projects Agency (DARPA) and the Advanced Research and Development Activity (ARDA), 2007.

[89] Tomlinson, S., Oard, D. W., Baron, J. R. and Thompson, P. Overview of the TREC 2007 legal track. In Voorhees, E. M. and Buckland, L. P. eds. *The Sixteenth Text REtrieval Conference Proceedings (TREC 2007).* (Gaithersburg, MD, November). National Institute of Standards and Technology (NIST) the Defense Advanced Research Projects Agency (DARPA) and the Advanced Research and Development Activity (ARDA), 2007.

[90] Tomlinson, S. Experiments with the negotiated boolean queries of the TREC 2006 legal discovery track. In Voorhees, E. M. and Buckland, L. P. eds. *The Fifteenth Text REtrieval Conference Proceedings (TREC 2006).* (Gaithersburg, MD, November). National Institute of Standards and Technology (NIST) the Defense Advanced Research Projects Agency (DARPA) and the Advanced Research and Development Activity (ARDA), 2006.

[91] Tomlinson, S. Learning Task Experiments in the TREC 2010 Legal Track. In Voorhees, E. M. and Buckland, L. P. eds. *The Nineteenth Text REtrieval Conference Proceedings (TREC 2010).* (Gaithersburg, MD, November 16-19, 2010). National Institute of Standards and Technology (NIST) the Defense Advanced Research Projects Agency (DARPA) and the Advanced Research and Development Activity (ARDA), 2010.

[92] Truchon, M. *An Extension of the Concordet Criterion and Kemeny Orders.* Université Laval - Département d'économique, cahier 98-15 du Centre de Recherche en Économie et Finance Appliqueés, 1998.

[93] Turtle, H. and Metzler, D. CIIR Experiments for TREC Legal 2007. In Voorhees, E. M. and Buckland, L. P. eds. *The Sixteenth Text REtrieval Conference Proceedings (TREC 2007).* (Gaithersburg, MD, November 2007). National Institute of Standards and Technology (NIST) the Defense Advanced Research Projects Agency (DARPA) and the Advanced Research and Development Activity (ARDA), 2007.

[94] van Erp, M. and Schomaker, L. Variants Of The Borda Count Method For Combining Ranked Classifier Hypotheses. In *Seventh International Workshop on Frontiers in Handwriting Recognition.* (Amsterdam, September 11-13, 2000), 443-452.

[95] *Victor Stanley, Inc. v. Creative Pipe, Inc*. 2010.

[96] Voorhees, E. M. Overview of TREC 2007. In Voorhees, E. M. and Buckland, L. P. eds. *The Sixteenth Text REtrieval Conference Proceedings (TREC 2007).* (November). National Institute of Standards and Technology (NIST) the Defense Advanced Research Projects Agency (DARPA) and the Advanced Research and Development Activity (ARDA), 2007.

[97] Voorhees, E. M., Gupta, N. K. and Johnson-Laird, B. Learning collection fusion strategies. In *SIGIR '95: Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval.* (Seattle, Washington, July 9-13, 1995). ACM, New York, NY, USA, 1995, 172-179.

[98] Willoughby, D. H. J., Jones, R. H. and Antine, G. R. Sanctions for E-Discovery Violations: By the Numbers. *Duke Law Journal* 60, 3 (Dec. 2010 2010), 789.

[99] Witten, I. H. and Frank, E. *Data Mining: Practical Machine Learning Tools and Techniques, Second Edition (Morgan Kaufmann Series in Data Management Systems).* Morgan Kaufmann, San Francisco, California, 2005.

[100] Young, H. P. and Levenglick, A. A Consistent Extension of Condorcet's Election Principle. *SIAM Journal on Applied Mathematics* 35, 2 (Sep. 1978), pp. 285-300.

[101] Yu, H. SVM selective sampling for ranking with application to data retrieval. In *KDD '05: Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining.* (Chicago, Illinois, August 21-24, 2005). ACM, New York, NY, USA, 2005, 354-363.

[102] Zhao, F. C., Lee, Y. and Medhi, D. Evaluation of Query Formulations in the Negotiated Query Refinement Process of Legal e-Discovery: UMKC at TREC 2007 Legal Track. In Voorhees, E. M. and Buckland, L. P. eds. *The Sixteenth Text REtrieval Conference Proceedings (TREC 2007).* (Gaithersburg, MD, November 2007). National Institute of Standards and Technology (NIST), the Defense Advanced Research Projects Agency (DARPA), and the Advanced Research and Development Activity (ARDA), 2007.

[103] Zhu, Y., Zhao, L., Callan, J. and Carbonell, J. Structured Queries for Legal Search. In Voorhees, E. M. and Buckland, L. P. eds. *The Sixteenth Text REtrieval Conference Proceedings (TREC 2007).* (Gaithersburg, MD, November 2007). National Institute of Standards and Technology (NIST) the Defense Advanced Research Projects Agency (DARPA) and the Advanced Research and Development Activity (ARDA), 2007.

[104] Zobel, J. and Moffat, A. Exploring the similarity space. *SIGIR Forum* 32, 1 (Spring 1998), 18-34.