Adapting Information Retrieval Techniques for a Biomedical Corpus

by

David L. Yeung

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Computer Science

Waterloo, Ontario, Canada, 2004

I hereby declare that I am the sole author of this thesis.

I authorize the University of Waterloo to lend this thesis to other institutions or individuals for the purpose of scholarly research.

David L. Yeung

I further authorize the University of Waterloo to reproduce this thesis by photocopying or other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

David L. Yeung

The University of Waterloo requires the signatures of all persons using or photocopying this thesis. Please sign below, and give address and date.

Abstract

We investigated the application of a variety of text retrieval techniques to the problem of retrieving biomedical journal articles from the MEDLINE database which are relevant to a particular gene. Our experiments were motivated by the University of Waterloo's participation in the Genome Track of the 2003 Text REtrieval Conference (TREC 2003), and conducted using the MultiText search engine developed at the University of Waterloo.

In adapting the MultiText search engine to MEDLINE, we did not incorporate domain expertise into the engine, nor did we use external biomedical resources such as dictionaries of synonyms or gene ontologies. Instead, we used techniques which have been shown to improve retrieval in a wide range of applications: shortest substring retrieval, query tiering, fusion, and query expansion. We experimented with query formulation using the Okapi BM25 retrieval model and examined different fusion techniques for combining retrieval methods. Metadata information in the MEDLINE records were used both for the construction of query tiers and for generating query expansions for feedback.

We discovered that a general purpose retrieval system can be successfully adapted for biomedical document retrieval by integrating the following features: a strategy for dealing with ambiguities in gene names, the ability to recognize the topic species of a particular document, and exploitation of metadata and other characteristics of the corpus. Our results showed that approaches that do not primarily involve domain-specific techniques can be effective for improving retrieval in a biomedical corpus, and hint at future directions for research in information retrieval in the genomics domain.

Acknowledgements

Thanks to my supervisor, Gordon V. Cormack, for his time and assistance, and my readers, Charles L. A. Clarke and Forbes J. Burkowski, for their suggestions. Thanks also to Egidio L. Terra for his help with some of the experiments described in this thesis.

The research in this thesis was supported in part by a Graduate Scholarship from the University of Waterloo.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Overview and Motivation

In recent years, there has been an enormous amount of discovery in genomics and related fields, which has been accompanied by a proportionate increase in the scientific literature. As a result of this growth, the information needs of researchers in biology-related fields have changed, and there is an increasingly urgent demand for the ability to isolate and locate relevant information in a sea of data. In particular, researchers often need to find documents related to the function of a particular gene.

The Text REtrieval Conference (TREC) introduced its Genomics Track in 2003 to encourage research in IR for bioinformatics applications. The primary task for the track is the ad hoc retrieval of documents from MEDLINE, a database of biomedical journal articles maintained by the National Library of Medicine (NLM), which are relevant to some particular genes. Although it appears to be a conventional ad hoc document retrieval task, this search task is made more difficult by the prevalence of lexical ambiguity in biomedical literature, where the meaning of a particular term is heavily dependent on context. The problem is mitigated by metadata associated with each document in the MEDLINE records, which supply the needed context through extensive annotation and by linkage to other documents or databases. The characteristics of the MEDLINE corpus and the structure of the genomics-related queries distinguish this task from previous IR problems, and suggest that techniques which have been especially fitted to the corpus would be effective.

This thesis describes our adaptation of the MultiText search engine for the ad hoc retrieval of biomedical documents from the MEDLINE database, carried out as part of our participation in the Genomics Track of TREC 2003. In tailoring the MultiText system for MEDLINE, we did not use any external bioinformatics resources, nor did we incorporate explicit domain expertise into our system. Our approach was to take an existing general purpose retrieval system and adapt it to the MEDLINE corpus by making use of the special characteristics of that corpus. We discover that certain elements are crucial to an effective retrieval system for biomedical documents, namely: a strategy for dealing with ambiguities in gene names, the ability to recognize the topic species of a particular document, and exploitation of metadata and other features of the corpus.

1.2 Thesis Outline

In the next chapter, we provide some background information on the field of information retrieval, describe the MultiText search engine, and give an overview of TREC. We explain the design of our experiments in Chapter 3 and present the results in Chapter 4. We conclude in Chapter 6 with some directions for future work.

Chapter 2

Background

2.1 Document Retrieval for Bioinformatics

There is a long history of research into document retrieval and information retrieval. Research into the automatic indexing of text started with experiments in the 1960s on index languages, such as the Cranfield tests [Cle67, Cle91]. The widespread availability of computers and the explosive growth in the popularity of the Internet has spurred research into the retrieval of information from large collections of documents. It is beyond the scope of the current thesis to give a complete overview of the current state of retrieval research. Surveys of the field may be found in Faloutsos and Oard [FO95], Voorhees [Voo99], and Greengrass [Gre00] .

Biomedical journal articles have certain characteristics which differentiate them from the types of documents previously considered in IR research. Within a biomedical corpus, *polysemy* (in which the same term refers to different objects) and *synonymy* (in which different terms refer to the same object) are major problems. Additional complications are caused by the inconsistent application of abbreviations and acronyms. Thus, acronym recognition and anaphora resolution are extremely important for document retrieval in the biomedical domain. Furthermore, the hierarchical relationships between the entities described in a biomedical corpus suggest that this structure can be used to improve retrieval performance. Research has been done on anaphora resolution [CZP02], the mapping of abbreviations to their full forms [YHF02], and on the recognition of gene and protein names [TW02b, TW02a, NSA02b] in a biomedical corpus, and the automatic construction of an acronym database from MEDLINE [PCC$^+$01, NSA02a]. The Medstract project [PCS$^+$02] has the ambitious goal of automatically extracting information from abstracts and articles in the MEDLINE database, using the latest techniques in natural language processing and text analysis. Research is also under way to investigate methods of transferring information found in the free text of scientific literature into ontologies and knowledge bases [CA02]. Due to the information-rich content of biomedical documents, much recent research into bioinformatics IR has focused on building expert knowledge, such as entity and relation identification, into the retrieval systems.

In addition to the above techniques which are based on bioinformatics-specific knowledge, a number of more general techniques based on expanding the query show promise for improving document retrieval in the bioinformatics domain. It has been shown that different IR systems and even different representations of a query retrieve different document sets [BCCC93, KJ98].

Automatic query expansion using blind feedback has been shown to improve retrieval performance in some situations [MSB98, SB90, Rob90]. In this type of feedback (also called "pseudo-relevance" feedback because input is not required from the user of the IR system), some number of the highest ranked documents retrieved using the original query are *assumed* to be relevant. These top documents are then used to expand the original query, and the modified query is used to retrieve another set of documents which is returned to the user. This type of feedback can improve or worsen performance, depending on the proportion of relevant documents in the documents used to generate the query expansion. Mitra et al. showed that refining the set of documents used in the feedback, using term co-occurrence information to estimate word correlation, often prevents query drift caused by blind expansion [MSB98]. Xu and Croft have shown that local feedback using only documents retrieved by the query is generally more effective than global techniques based on the entire corpus [XC96].

Documents which are ranked highly by disparate systems are much more likely to be actually relevant. Thus, instead of relying on the output of any single IR system, performance can be improved by merging the results of different systems using a fusion technique [Lee97, BCB94, FS93]. Fox and Shaw proposed a number of rules for combining evidence from multiple retrieval systems, by assigning weights to each and combining the weighted scores in different ways [FS93]. Lee performed experiments on these rules and developed the ideas further [Lee97]. Bartell et al. proposed a method by which the relevance estimates made by different systems can be automatically combined, using a parametrized mixture of the relevance scores produced by each system [BCB94].

2.2 The Text REtrieval Conference (TREC)

The Text REtrieval Conference (TREC) is an annual event co-sponsored by the U.S. National Institute of Standards and Technology (NIST), the Information Awareness Office of the Defense Advanced Research Projects Agency (DARPA/IAO), and the U. S. Department of Defense Advanced Research and Development Activity (ARDA) [TRE03, Voo02]. Each year at TREC, groups from academia and industry develop information retrieval (IR) systems for performing various tasks, for the purpose of evaluating and comparing different IR techniques and systems in a standard and unbiased manner. The tasks are grouped into various areas of focus called "tracks", each of which is devoted to a particular subject of interest. Typically, each track deals with some specific information need.

The first TREC was held in 1992 [Har92], and the conference has been held every year since then. The number of group taking part in the conference has increased from 25 at the first TREC to 93 at TREC 2003, which took place in November of that year, and includes participants from academic, commercial, and government institutions.

The purpose of TREC is to provide a common platform for the comparison of different IR systems, in a standard and unbiased manner. TREC has four main goals [Voo02]:

- to encourage retrieval research based on large test collections;

- to increase communication among industry, academia, and government by creating an open forum for the exchange of research ideas;
- to speed the transfer of technology from research labs into commercial products by demonstrating substantial improvements in retrieval methodologies on real-world problems; and
- to increase the availability of appropriate evaluation techniques for use by industry and academia, including development of new evaluation techniques more applicable to current systems.

Every year at TREC there are a number of areas of focus called "tracks". In 2003, these consisted of the Ad Hoc Track, the Genomics Track, the HARD (High Accuracy Retrieval from Documents) Track, the Interactive Track, the Novelty Track, the Question Answering Track, the Robust Track, and the Web Track.

TREC is based on the Cranfield paradigm, in which different retrieval systems are evaluated on the same *test collection* [Cle67, Cle91]. A test collection consists of a *document set* (called the "corpus"), a set of *information need statements* (the "topics"), and a set of *relevance judgments* (called "qrels" in TREC lingo). The relevance judgments consist of a list of documents that have been judged relevant for each topic and hence should be retrieved by an IR system for that topic. Given the corpus and topics, the retrieval task is then to retrieve all of the relevant documents and none of the non-relevant ones. The effectiveness of each IR system is evaluated based on precision and recall.

Precision measures a system's ability to find *only* relevant documents (or equivalently, to filter out non-relevant documents):

$$\text{Precision} = \frac{\text{number of relevant documents retrieved}}{\text{total number of documents retrieved}}$$

Recall measures a system's ability to find *all* relevant documents:

$$\text{Recall} = \frac{\text{number of relevant documents retrieved}}{\text{number of relevant documents in the collection}}$$

The *average precision* (AP) for each topic is the average of the precision scores after each relevant document retrieved. The *mean average precision* (MAP) is the average of the AP over the entire set of topics. This value is computed in the standard TREC manner by using the `trec_eval` program written by Chris Buckley.

Two sets of topics are supplied to the participants, the *training topics* and the *test topics*. Relevance judgments are provided to the participants for the training topics, but not for the test topics. The training topics are assumed to be similar in characteristic to the test topics. Participants can adjust their systems using the training data (topics and relevance judgments) in order to improve the performance of their systems on the test data.

2.2.1 TREC 2003 Genomics Track

The first year of the TREC Genomics Track took place in 2003. Its purpose is to provide a forum for evaluating IR systems in the genomics domain. An overview of this track is given by Hersh and Bhupatiraju [HB03]. The track featured two tasks, and a total of 29 groups participated in one or both of these.

The secondary task for the TREC 2003 Genomics Track was an information extraction and document summarization task. We did not participate in this track.

The primary task for the TREC 2003 Genomics Track was the ad hoc document retrieval of journal articles from MEDLINE which discuss the basic biology or protein products of a particular gene. The task is officially defined as follows:

> "For gene X, find all MEDLINE references that focus on the basic biology of the gene or its protein products from the designated organism. Basic biology includes isolation, structure, genetics and function of genes/proteins in normal and disease states." [Her03]

MEDLINE is the bibliographical database of biomedical journal articles maintained by the National Center for Biotechnology Information (NCBI), a division of the National Library of Medicine (NLM). A subset of this database, consisting of 525,938 records for which indexing was completed between April 1, 2002 and April 1, 2003, was used as the corpus for this track. The corpus was made available in both standard NLM MEDLINE format and in XML. Each MEDLINE record comprises a number of fields, each of which is designated by a 2 to 3 letter abbreviation. These include the document's title (TI), its abstract (AB), and a PubMed Identifier (PMID) which uniquely labels the document. The full journal articles are not included in the database, although some of them are available from other sources. There are also fields containing controlled vocabulary which provide a linkage between the document and structured data. Two of these fields that were particularly important are the MeSH Heading (MH) and Registry Number (RN) fields.

MeSH (Medical Subject Heading) is a lexical hierarchy for describing medical concepts. Each MeSH concept may be referred to by a number of synonymous terms. However, the MeSH Heading field in the MEDLINE record uses a controlled vocabulary to ensure that a standard nomenclature is maintained throughout the corpus. The Registry Number field is used to list the chemicals mentioned in the document, which are also reported using a controlled vocabulary and which may be mapped to the MeSH concepts. The MEDLINE metadata tags are explained in detail on the PubMed web site [NCB03b].

Training and test topic sets of 50 genes each were distributed to each of the participating groups. Each group was to develop and test its IR system on the training data, and was allowed to submit up to two official runs with the test data. To assist the groups in developing their systems, relevance judgments were made available for the training topics. Participating groups were to develop and test their IR systems on the training data, and to submit two official runs on the test data to NIST for evaluation and analysis. Relevance judgments for the test topics were not released until after the official result submission deadline.

Each topic consists of a single gene, identified by its LocusLink ID number, and a target organism. A list of variant ways of referring to the gene is also supplied, and each given gene name is tagged with one of the following gene name types: official gene name, preferred gene name, official symbol, preferred symbol, or preferred product. The target organism was limited to four species: *Homo sapiens* (human), *Mus musculus* (mouse), *Rattus norvegicus* (rat), and *Drosophila melanogaster* (fruit fly).

1	1026	Homo sapiens	OFFICIAL_GENE_NAME	"cyclin-dependent kinase inhibitor 1A (p21, Cip1)"
1	1026	Homo sapiens	OFFICIAL_SYMBOL	CDKN1A
1	1026	Homo sapiens	ALIAS_SYMBOL	P21
1	1026	Homo sapiens	ALIAS_SYMBOL	CIP1
1	1026	Homo sapiens	ALIAS_SYMBOL	SDI1
1	1026	Homo sapiens	ALIAS_SYMBOL	WAF1
1	1026	Homo sapiens	ALIAS_SYMBOL	CAP20
1	1026	Homo sapiens	ALIAS_SYMBOL	CDKN1
1	1026	Homo sapiens	ALIAS_SYMBOL	MDA-6
1	1026	Homo sapiens	PREFERRED_PRODUCT	cyclin-dependent kinase inhibitor 1A
1	1026	Homo sapiens	PRODUCT	cyclin-dependent kinase inhibitor 1A
1	1026	Homo sapiens	PRODUCT	cyclin-dependent kinase inhibitor 1A
1	1026	Homo sapiens	ALIAS_PROT	DNA synthesis inhibitor
1	1026	Homo sapiens	ALIAS_PROT	CDK-interaction protein 1
1	1026	Homo sapiens	ALIAS_PROT	wild-type p53-activated fragment 1
1	1026	Homo sapiens	ALIAS_PROT	melanoma differentiation associated protein 6

Figure 2.1: An example topic for the Genomics Track (training topic 1).

PubMed ID	Statement of Function (GeneRIF Text)
12388558	role of PIN1 in transactivation
11642719	expression is related to apoptosis in thymus
12459877	p21(waf1) has a role in aortal endothelial cell aging
11762751	expression inhibited by Hepatitis C virus core protein
12474524	Codon 31 polymorphism is associated with bladder cancer
11748297	induced after DNA damage and plays a role in cell survival
11781193	expression in normal, hyperplastic and carcinomatous human prostate
12513833	p21(WAF1) transfection decreases sensitivity of K562 cells to VP-16

Figure 2.2: A subset of the GeneRIFs for training topic 1, LocusLink ID 1026 (cyclin-dependent kinase inhibitor 1A).

For example, training topic 1 is the gene identified by the LocusLink ID 1026, "cyclin-dependent kinase inhibitor 1A". Figure 2.1 shows the format of the training topics file. The first two columns contain the topic number and the LocusLink ID, and the third column is the name of the organism (i.e. its species). The fourth column indicates the gene name type, and the actual gene name is found in the fifth column. The topic is provided in this format for convenience. It is sufficient to supply only the LocusLink ID, and the rest of the information may be obtained from LocusLink using this ID number.

In order to produce a large number of relevance judgments in a short amount of time, the track steering committee decided that GeneRIF (Gene Reference Into Function) data from NLM's LocusLink database [NCB03a] would be used as relevance judgments. Each GeneRIF for a gene consists of a PubMed ID pointing to a MEDLINE article which discusses some function of the gene, along with a brief statement about that function. GeneRIFs have been systematically assigned since April 2002. A document was judged to be relevant to a gene if a GeneRIF existed for that gene and the GeneRIF pointed to that document. Because relevance judgments for the track were based on GeneRIFs, groups were not allowed to use GeneRIF data in their retrieval systems.

One potential problem with using GeneRIFs as a "gold standard" is that they are incomplete, in the sense that there were some documents which are related to a gene but which have not yet been assigned a GeneRIF. As a result, there are many *false negatives* (documents which are relevant but which are not judged to be relevant).

Figure 2.2 shows some of the GeneRIFs for training topic 1. One of the GeneRIFs for this gene points to the document with PubMed ID 12388558 and has the statement of function "role of PIN1 in transactivation". Therefore, a retrieval system searching on training topic 1 is expected to retrieve this particular document.

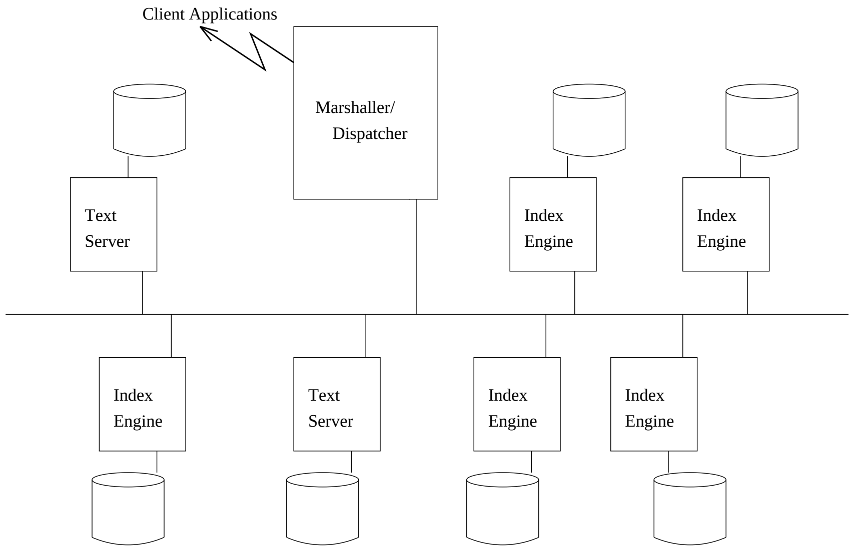

Figure 2.3: The architecture of the MultiText retrieval system.

2.3 The MultiText Search Engine

The MultiText search engine is a general purpose information retrieval system developed at the University of Waterloo. The system has been in development since 1993, and since its inception the project has centred around the development of scalable technologies for distributed information retrieval. The MultiText research group has participated in TREC annually since TREC-4 in 1995, performing retrieving experiments with a passage-based ranking algorithm called Shortest Substring Ranking, developing a precise query language called GCL that yields and combines arbitrary intervals of text, and taking part in various tracks [CCB94, CCB95, CC96, CCPT00, CPVC98, CCKP99, CCKL00, CCL+01, CCK+02, YCC+03].

The MultiText retrieval system is based on the federated architecture shown in Figure

2.3. It comprises the index engines (which maintain the index file structures and provide search capabilities), the text servers (which are specialized by document type and provide retrieval capabilities for arbitrary text passages specified at the word level), and the marshaler/dispatcher (which interacts with clients and coordinates query and update activities).

We adapted the MultiText retrieval system for the Genomics Track by loading the XML version of the MEDLINE database into MultiText and building a number of additional subsystems on top of the basic MultiText engine. We call the resultant system MultiText for Genomics.

2.3.1 GCL

The MultiText retrieval system models the text in a database as a continuous sequence of *terms*, and indicates document structure by indexing structural markers, called *metadata tags*, in between the terms. Metadata tags generally occur in pairs (the *start* and *end* tags). For example, the text of a document is enclosed between the tags `<DOC>` and `</DOC>`, while the text forming the document's title are further enclosed between the tags `<ArticleTitle>` and `</ArticleTitle>`. Text terms and metadata tags are together referred to as *tokens*, and each token in the database is assigned an integer value indicating its position.

The query language used in the MultiText retrieval system is based on the Generalized Concordance Lists (GCL) of Clarke, Cormack, and Burkowski [CCB94]. The GCL

GCL Expression	Query Represented by Expression
`"any_phrase"`	Any phrase (the underscore character is matched to whitespace and punctuation).
`"head*"`	Any term starting with "head".
`"$stem"`	Any term with the same (Porter) stem as "stem".
`gcl1..gcl2`	An interval containing `gcl1` followed by `gcl2`.
`gcl1^gcl2`	A solution containing both `gcl1` and `gcl2`.
`gcl1+gcl2`	A solution containing either `gcl1` or `gcl2`.
`gcl1>gcl2`	A solution to `gcl1` containing a solution to `gcl2`.
`gcl1<gcl2`	A solution to `gcl1` contained in a solution to `gcl2`.
`gcl1/>gcl2`	A solution to `gcl1` *not* containing a solution to `gcl2`.
`gcl1/<gcl2`	A solution to `gcl1` *not* contained in a solution to `gcl2`.
`1/gcl`	Solutions of the form (n, n) where (n, m) is a solution to `gcl`.
`2/gcl`	Solutions of the form (m, m) where (n, m) is a solution to `gcl`.
`1^(gcl1,gcl2,gcl3,...)`	Equivalent to `(gcl1+gcl2+gcl3+...)`.
`2^(gcl1,gcl2,gcl3,...)`	Equivalent to `((gcl1^gcl2)+(gcl1^gcl3)+...)` (i.e. any 2 of the solutions).
`n^(gcl1,gcl2,gcl3,...)`	Generalization of the previous rule, with n any positive integer.
`all^(gcl1,gcl2,gcl3,...)`	Equivalent to `(gcl1^gcl2^gcl3^...)`.
`(gcl1^gcl2)<[n]`	An interval with `gcl1` and `gcl2` within `n` words.
`gcl1<([n]>gcl2)`	An interval with `gcl1` that is within n words of `gcl2`.
`gcl<{n,m}`	Find `gcl` within the range (n, m).

Table 2.1: The syntax of GCL. In the above, `gclX` stands for any GCL sub-expression.

query algebra expresses searches on structured text using a number of operators, such as *boolean AND* (^), *boolean OR* (+), *containing* (>), *contained in* (<), *not containing* (/>), *not contained in* (/<), *followed by* (..), and so on. Table 2.1 gives a list of example GCL expressions and the query represented by the expression.

The algebra manipulates arbitrary intervals of text, and provides for queries that harness document structure by allowing metadata tags to be used in the query. GCL expressions can be combined and nested to form more complex queries. The result or solution to a GCL query is a set of intervals from the text, with each interval represented by an ordered pair (n, m) with $n < m$, corresponding to the integer values of the first and last token of a passage in the text satisfying the query. The solution set includes all passages in the corpus that satisfy the query, and which do not contain shorter substrings also satisfying the query. This *shortest substring rule* limits the number of passages that must be considered by the algorithm, and is the foundation behind the passage-based document ranking technique described below. For example, the GCL query ("<doc>".."</doc>")>"cdkn1a" has as its result the set of all documents containing the term "cdkn1a". The shortest substring rule ensures that the solution set contains only single documents. Start and end tags which occur in separate documents are not linked.

As another example, for "phospholipase C, gamma 1" (training topic 23), the MultiText for Genomics system generates, along with other queries, the following query: "c_gamma"^("phospholipase"+ "phospholipases"). Since the algorithm locates the shortest substrings that satisfy the query, a passage located by the algorithm will begin (or end) with the phrase "c_gamma" (where the underscore character is matched to whitespace or punctuation) and end (or begin) with one of the words "phospholipase" or "phospholipases". None of these terms will appear elsewhere in the passage, since otherwise the passage would contain a shorter substring that also satisfies the query.

Other structural constraints (metadata tags) can be applied to the query. For example, the query ("<NameOfSubstance>".."</NameOfSubstance>")>"cip1" identifies instances of the NameOfSubstance metadata field that contain the term "cip1". The GCL query ("<docno>".."</docno>")<(("<doc>".."</doc>")>"cdkn1a") retrieves the document numbers of all documents containing the term "cdkn1a". (In the case of the MEDLINE corpus, the document number for each document is its PubMed ID.)

2.3.2 Shortest Substring Ranking (SSR)

A solution to a GCL query is a set of intervals satisfying the query from the text. Ideally, intervals in which the query terms occur densely together should be favoured or ranked more highly. The Shortest Substring Ranking (SSR) method is a ranked retrieval method that assigns scores to the passages retrieved based on this idea. SSR is a technique that has been successfully deployed by the MultiText group in a number of applications [CCPT00, CC00].

Given a query and the resulting passages satisfying the SSR rule, a document's score is computed based on the lengths of all such passages contained within it. Suppose that document d contains n passages satisfying the query under the SSR rule, labelled $P_1, P_2, \ldots, P_n$ in order of increasing length. We compute a score for d that rewards higher values of n

and shorter passages. For a passage P corresponding to the extents (p, q), we define

$$I(P) = \begin{cases} \frac{\mathcal{K}}{l(P)} & \text{if } l(P) \geq \mathcal{K} \\ 1 & \text{if } l(P) \leq \mathcal{K} \end{cases}$$

where $l(P)$ is the length of P in alphanumeric tokens; that is, $l(P) = q - p + 1$. Note that for any passage P, we have $0 < I(P) \leq 1$. The score for d is then computed by the formula:

$$\sum_{i=0}^{n} I(P_i)^{\gamma}$$

For the MultiText for Genomics system, the parameters we used for SSR were $\mathcal{K} = 16$ and $\gamma = 0.5$. The exact details of the scoring function may be found in Clarke and Cormack [CC00], where an efficient algorithm for implementing SSR is also given.

2.3.3 The Okapi Measure

The Okapi measure is a well-known probabilistic retrieval model that uses weighting functions based on term frequencies [RWJ+94, RW94]. The MultiText system also has a special implementation of the Okapi BM25 retrieval model, which as an extension also allows phrases to be used as query terms. Otherwise, the implementation of Okapi BM25 used in the MultiText for Genomics system follows the description of Robertson et al. [RWB98] with the the standard parameters $k_1 = 1.2$, $b = 0.75$, $k_2 = 0$, and $k_3 = \infty$.

Specifically, given an Okapi term set Q, a document d is assigned the score

$$\sum_{t \in Q} w^{(1)} q_t \frac{(k_1 + 1) d_t}{K + d_t}$$

where

$$
\begin{aligned}
w^{(1)} &= \log\left(\frac{D - D_t + 0.5}{D_t + 0.5}\right) \\
D &= \text{number of documents in the corpus} \\
D_t &= \text{number of documents containing } t \\
q_t &= \text{frequency that } t \text{ occurs in the topic} \\
d_t &= \text{frequency that } t \text{ occurs in } d \\
K &= k_1((1 - b) + b \cdot l_d / l_{avg}) \\
l_d &= \text{length of } d \\
l_{avg} &= \text{average document length}
\end{aligned}
$$

Chapter 3

Experimental Design

The MultiText for Genomics system uses an elaborate combination of techniques, which were selected and tweaked based on experimentation with the corpus and training data. The system may be roughly divided into four subsystems:

1. Okapi
2. Query Tiering
3. Fusion
4. Feedback

Given a Genomics Track topic, the Okapi subsystem generates multiple term sets from the supplied gene name information (recall Figure 2.1), which are then used to retrieve several sets of documents using the Okapi retrieval model. Simultaneously, the Query Tiering subsystem attempts to retrieve documents by matching the gene name information against a number of query tiers. The results from from the first two subsystems are merged in the Fusion subsystem, and depending on the outcome, the Feedback subsystem may retrieve additional documents using pseudo-relevance feedback to supplement the results.

Before describing each subsystem, we describe an operation that is commonly carried out in the MultiText for Genomics system, that of appending one document list to the end of another. Let L_1 and L_2 be ranked lists of documents, and for a document d, let $s_{L_1}(d)$ be the score of the document in L_1 if $d \in L_1$, and $s_{L_2}(d)$ be the score of the document in L_2 if $d \in L_2$. Let $L_2' = L_2 \setminus L_1$ (then there are common documents between L_1 and L_2'). Let $S_{L_1,min}$ be the lowest score $s_{L_1}(d)$ for a document $d \in L_1$, and let $S_{L_2',max}$ be the highest score $s_{L_2}(d)$ for a document $d \in L_2'$. Then let $L = L_1 \cup L_2'$, with the scoring function

$$s_L(d) \begin{cases} s_{L_1}(d) & \text{if } d \in L_1 \\ s_{L_2}(d) \times \frac{S_{L_1,min}}{S_{L_2',max}} & \text{if } d \in L_2 \end{cases}$$

We say that the document list L is the result of appending L_2 to the end of L_1 with the scores appropriately scaled, and write $L = append(L_1, L_2)$.

3.1 The Okapi Subsystem: Query Formulation

Two important facts emerged during preliminary experiments on the MEDLINE corpus, which influenced the design of the experiments using the Okapi retrieval model.

First, the gene name type (official gene name, preferred gene name, official symbol, preferred symbol, or preferred product) did not seem to matter. A document discussing a

Term ends in	Action taken
-ch, -sh, -ss, -x, -z, -s	Append "-es".
-y, -ey	Replace with "-ies".
Other letter	Append "-s".

Table 3.1: Rules for "pluralization".

particular gene was as likely to use an official name as an alternate one.

Second, spacing and punctuation had a large effect on performance in some cases. The gene or protein names which have been supplied for each topic (derived from LocusLink) may differ from the gene or protein names as they actually appear in the corpus by the addition or removal of punctuation or whitespace, or by the re-arrangement of terms. In a model based on term vectors, such as Okapi, even slight variations may significantly affect the results.

We attempt to capture these morphological differences by producing three sets of Okapi term vectors with differing degrees of fidelity to the original gene and protein names, by using heuristics to process semi-colons, commas, and brackets and generating plurals for some terms. The three rules we used to generate Okapi term vectors are:

- Okapi 1:

  Each gene name in the original LocusLink-derived query, which may consist of multiple alphanumeric tokens, is considered as a phrase and treated as a single term in the Okapi term set. All punctuation is removed and replaced by whitespace. (The search engine treats punctuation and whitespace in the corpus identically.)

Example: Figure 3.1 shows the Okapi 1 term vector for training topic 1.

- Okapi 2:

  Heuristics are used to handle brackets in the gene and protein names:

  1. An internal bracket is unchanged. (Thus, the gene name "l(1)hop" for training topic 36 retains its brackets).
  2. If the terms between the brackets comprise only numbers and letters (including Greek letters), the brackets are removed. (The official gene name for training topic 12 is "tropomyosin 1 (alpha)", which is changed to "tropomyosin 1 alpha".)
  3. Otherwise, the contents of the brackets are considered to be alternate names, which are treated as separate terms in the Okapi vector. (For training topic 31, the official gene name "Tachykinin (substance P, neurokinin A, neuropeptide K, neuropeptide gamma)" is broken up into the separate gene names "Tachykinin", "substance P", "neurokinin A", "neuropeptide K", and "neuropeptide gamma".)

  Similar rules are used to break up lists separated by commas and semi-colons.

  "Plurals" are generated using the simple set of rules shown in Table 3.1. If a term consists of all alphabetical characters and is three letters or longer, and is not a Greek letter or a stop word, the "plural" of the term is generated using these rules and added to the term vector.

  Example: Figure 3.2 shows the Okapi 2 term vector for training topic 1.

- Okapi 3:

  First, the gene and protein names are separated into two sets, one containing those that comprise a single token, and another containing those comprising multiple tokens. (For training topic 1, "p21", "cip1", and so on are put into the single-token set, while "cyclin-dependent kinase inhibitor 1A" is put into the multiple-token set.)

  For the single-token set, all pairs of distinct elements are taken, and each pair is concatenated together, with and without a space between them, to form terms which are then included in the Okapi term vector. (For training topic 1, the terms "p21 cip1", "p21cip1", "cip1 p21", and "cip1p21" are generated among others for the Okapi term vector.)

  For the multiple-token set, for each term comprising multiple tokens, all bigrams of the terms are generated and added to the Okapi term vector. (For training topic 1, the term "cyclin-dependent kinase inhibitor 1A" generates "cyclin dependent", "dependent kinase", "kinase inhibitor", and "inhibitor 1A".)

  Example: Figure 3.3 shows the Okapi 3 term vector for training topic 1.

In addition to the above rules, the name of the topic species was also included in each of the Okapi term vectors. We attempted other variations on the above rules, but experiments on the training data found that the above rules gave the best overall results.

The three rules are in decreasing order of strictness. Documents retrieved by Okapi 1 will contain the terms exactly as given in the original query (ignoring punctuation), while those retrieved by Okapi 2 will contain terms which are similar to but not exactly like

“cap20”, “cdk interaction protein 1”, “cdkn1”, “cdkn1a”, “cip1”, “cyclin dependent kinase inhibitor 1a p21 cip1”, “cyclin dependent kinase inhibitor 1a”, “dna synthesis inhibitor”, “mda 6”, “melanoma differentiation associated protein 6”, “p21”, “sdi1”, “waf1”, “wild type p53 activated fragment 1”, “Homo sapiens”, “humans”, “human”

Figure 3.1: Okapi 1 term vector for training topic 1.

“cap20”, “cdk interaction protein 1”, “cdkn1”, “cdkn1a”, “cip1”, “cyclin dependent kinase inhibitor 1a”, “dna synthesis inhibitor”, “mda 6”, “mda6”, “melanoma differentiation associated protein 6”, “p21”, “sdi1”, “waf1”, “wild type p53 activated fragment 1”, “Homo sapiens”, “humans”, “human”

Figure 3.2: Okapi 2 term vector for training topic 1.

“activated fragment”, “activatedfragment”, “associated protein”, “associatedprotein”, “cap20”, “cap20 cdkn1”, “cap20 cdkn1a”, “cap20 cip1”, “cap20 mda 6”, “cap20 mda6”, “cap20 p21”, “cap20 sdi1”, “cap20 waf1”, “cap20cdkn1”, “cap20cdkn1a”, “cap20cip1”, “cap20mda 6”, “cap20mda6”, “cap20p21”, “cap20sdi1”, “cap20waf1”, “cdk interaction”, “cdk interaction protein 1”, “cdkinteraction”, “cdkn1”, “cdkn1 cap20”, “cdkn1 cdkn1a”, “cdkn1 cip1”, “cdkn1 mda 6”, “cdkn1 mda6”, “cdkn1 p21”, “cdkn1 sdi1”, “cdkn1 waf1”, “cdkn1a”, “cdkn1a cap20”, “cdkn1a cdkn1”, “cdkn1a cip1”, “cdkn1a mda 6”, “cdkn1a mda6”, “cdkn1a p21”, “cdkn1a sdi1”, “cdkn1a waf1”, “cdkn1acap20”, “cdkn1acdkn1”, “cdkn1acip1”, “cdkn1amda 6”, “cdkn1amda6”, “cdkn1ap21”, “cdkn1asdi1”, “cdkn1awaf1”, “cdkn1cap20”, “cdkn1cdkn1a”, “cdkn1cip1”, “cdkn1mda 6”, “cdkn1mda6”, “cdkn1p21”, “cdkn1sdi1”, “cdkn1waf1”, “cip1”, “cip1 cap20”, “cip1 cdkn1”, “cip1 cdkn1a”, “cip1 mda 6”, “cip1 mda6”, “cip1 p21”, “cip1 sdi1”, “cip1 waf1”, “cip1cap20”, “cip1cdkn1”, “cip1cdkn1a”, “cip1mda 6”, “cip1mda6”, “cip1p21”, “cip1sdi1”, “cip1waf1”, “cyclin dependent”, “cyclin dependent kinase inhibitor 1a”, “cyclindependent”, “dependent kinase”, “dependentkinase”, “differentiation associated”, “differentiationassociated”, “dna synthesis”, “dna synthesis inhibitor”, “dnasynthesis”, “fragment 1”, “fragment1”, “inhibitor 1a”, “inhibitor1a”, “interaction protein”, “interactionprotein”, “kinase inhibitor”, “kinaseinhibitor”, “mda 6”, “mda 6 cap20”, “mda 6 cdkn1”, “mda 6 cdkn1a”, “mda 6 cip1”, “mda 6 mda6”, “mda 6 p21”, “mda 6 sdi1”, “mda 6 waf1”, “mda 6cap20”, “mda 6cdkn1”, “mda 6cdkn1a”, “mda 6cip1”, “mda 6mda6”, “mda 6p21”, “mda 6sdi1”, “mda 6waf1”, “mda6”, “mda6 cap20”, “mda6 cdkn1”, “mda6 cdkn1a”, “mda6 cip1”, “mda6 mda 6”, “mda6 p21”, “mda6 sdi1”, “mda6 waf1”, “mda6cap20”, “mda6cdkn1”, “mda6cdkn1a”, “mda6cip1”, “mda6mda 6”, “mda6p21”, “mda6sdi1”, “mda6waf1”, “melanoma differentiation”, “melanoma differentiation associated protein 6”, “melanomadifferentiation”, “p21”, “p21 cap20”, “p21 cdkn1”, “p21 cdkn1a”, “p21 cip1”, “p21 mda 6”, “p21 mda6”, “p21 sdi1”, “p21 waf1”, “p21cap20”, “p21cdkn1”, “p21cdkn1a”, “p21cip1”, “p21mda 6”, “p21mda6”, “p21sdi1”, “p21waf1”, “p53 activated”, “p53activated”, “protein 1”, “protein 6”, “protein1”, “protein6”, “sdi1”, “sdi1 cap20”, “sdi1 cdkn1”, “sdi1 cdkn1a”, “sdi1 cip1”, “sdi1 mda 6”, “sdi1 mda6”, “sdi1 p21”, “sdi1 waf1”, “sdi1cap20”, “sdi1cdkn1”, “sdi1cdkn1a”, “sdi1cip1”, “sdi1mda 6”, “sdi1mda6”, “sdi1p21”, “sdi1waf1”, “synthesis inhibitor”, “synthesisinhibitor”, “type p53”, “typep53”, “waf1”, “waf1 cap20”, “waf1 cdkn1”, “waf1 cdkn1a”, “waf1 cip1”, “waf1 mda 6”, “waf1 mda6”, “waf1 p21”, “waf1 sdi1”, “waf1cap20”, “waf1cdkn1”, “waf1cdkn1a”, “waf1cip1”, “waf1mda 6”, “waf1mda6”, “waf1p21”, “waf1sdi1”, “wild type”, “wild type p53 activated fragment 1”, “wildtype”, “Homo sapiens”, “humans”, “human”

Figure 3.3: Okapi 3 term vector for training topic 1.

those in the original query. Documents retrieved by Okapi 3 contain the same bigrams as found in the original query.

Each query formulation has its own advantages and disadvantages. The top documents returned by Okapi 1 are likely to be relevant, since they contain the query exactly, but many relevant documents may be missed because the gene name in the document appears differently than in the query. On the other hand, Okapi 3 retrieves many relevant documents in which the gene name does not appear exactly as in the query. However, it also retrieves many irrelevant documents. The documents retrieved by Okapi 2 are intermediate between the two.

We found that the document sets retrieved using the term vectors generated by the three rules were quite different. Therefore, a document that is retrieved by *all* three term vectors was very likely to be relevant, and it was decided that the three result sets should be fused together to produce the final result. After experimenting with a number of fusion techniques, it was decided that the fusion was to be accomplished in the following manner:

- Okapi Fusion:

  The document sets retrieved by Okapi 1, Okapi 2, and Okapi 3 are combined by taking the intersection of the three sets. A document's score is taken to be the product of the three scores. This list is then followed by the remainder of Okapi 3, with the scores appropriately scaled.

  More formally, let O_1, O_2, and O_3 be the document sets retrieved by the Okapi 1, Okapi 2, and Okapi 3 term vectors respectively. Let d be a document, and let

$s_i(d)$ be the score assigned to d by Okapi i, for $i = 1, 2, 3$. Then $F' = O_1 \cap O_2 \cap O_3$ is the intersection of the three document sets. For each document $d \in F'$, define $s_{F'}(d) = s_1(d) \times s_2(d) \times s_3(d)$ to be the score of that document in F'. Then the Okapi Fusion is $F = append(F', O_3)$.

The rationale behind the fusion is that a document that scores highly on all three query formulations is very likely to be relevant. Taking the product of the scores allows each of the three document sets to vote on the relative distance (in terms of rank) between retrieved documents. Since Okapi 3 is the most relaxed of the three query formulations, it retrieves most if not all of the documents retrieved by Okapi 1 and 2. Thus, the intersection of the three document sets likely contains most of the relevant documents in the document sets returned by Okapi 1 and 2, while it might miss relevant documents retrieved by Okapi 3. For that reason, the remainder of the Okapi 3 document set is appended to the end of the combined list.

While there are other standard fusion techniques, the above seemed to work very well in preliminary trials, and thus was the only technique used in the final completed runs. It would be interesting to experiment with other fusion techniques for combining the Okapi document sets.

The performance of the Okapi 1 term vector set alone was considered to be the baseline run for comparison purposes with our other runs.

3.2 The Query Tiering Subsystem: Use of Metadata

The MEDLINE records are highly structured, and some of the metadata fields are more useful indicators than others of a document's relevance. Preliminary experiments showed that there was a correlation between some of the metadata fields in the MEDLINE record and the relevance of the document. In particular, there was a strong correspondence between the query terms and the terms that appeared in the RN (registry number) field of the MEDLINE record. The RN field contains a list of the chemicals discussed in the document. Many of these chemical names can be matched to the gene names found in query. The chemical list is a better indicator of a document's relevance than the document's title, which in turn is a better indicator than the abstract. To capture this hierarchy of relevance between the metadata fields, we used a number of query tiers. In particular, the RN field of each MEDLINE record contains a list of chemicals mentioned in that document. Many of these chemical names can be matched to the gene names given in the query, and thus there is a high degree of correlation between the contents of the RN field and the relevance of that document.

Through experimentation, we arrived at the following system of six query tiers, which are given in decreasing order of relevance. The first query tier attempts to match the query against the chemical list exactly (except for stop words, spacing, and punctuation). The second and third tiers are relaxations of the first. The query is converted into a boolean expression by turning each gene name into the conjunction of its terms, and taking the disjunction of all gene names. This expression is then applied to the title for the fourth tier, to the chemical list for the fifth tier, and to the abstract for the sixth tier.

- Tier 1:

  The gene name is found in the chemical list, or it is found in the chemical list preceded or followed by the word "protein", optionally followed by the name or description of the species. Spaces and punctuation are ignored for the purposes of comparison.

  Examples: For training topic 1, all documents with "cip1 protein" in the chemical list are retrieved. For training topic 5, "glycine receptor, alpha 1" is considered to be equivalent to "glycine receptor alpha1".

- Tier 2:

  This tier is similar to Tier 1, except that the chemical name is allowed to have additional terms.

  Examples: For training topic 1, the gene name "p21" is matched to the phrase "p21-activated kinase 1" in the chemical list. For training topic 11, "RAC1" retrieves documents in which "rac1 GTP-Binding Protein" appears in the chemical list.

- Tier 3:

  An attempt is made to find the conjunction of the terms from the gene name in the chemical list. If the gene name consists of a class name followed by a sequence of letters and numbers that specifies an object of that class, the name is successively (“sdi1”+(“cyclin”ˆ“dependent” ˆ “kinase” ˆ “inhibitor”ˆ“1a”)+(“cdk”ˆ “interaction”ˆ“protein”ˆ“1”)+“cdkn1”+“cip1”+(“mda”ˆ“6”)+(“dna”ˆ“synthesis”ˆ “inhibitor”)+“cap20”+“p21”+(“wild”ˆ“type”ˆ“p53”ˆ“activated”ˆ“fragment”ˆ “1”)+“mda6”+“cdkn1a”+(“melanoma”ˆ“differentiation”ˆ“associated”ˆ “protein”ˆ“6”)+“waf1”)

Figure 3.4: Boolean expression for training topic 1.

weakened until a match is made. Heuristics are also used to recognize plurals.

Example: From training topic 32, “estrogen receptor 1” is weakened until the documents retrieved contain “Receptors, Estrogen” in the chemical list.

- Tier 4:

  The query is converted into a boolean expression by turning each gene name into the conjunction of its terms, and taking the disjunction of all gene names. The boolean expression is matched against the title metadata field.

  Example: Figure 3.4 shows the boolean expression generated for training topic 1. Among other documents, this expression retrieves the document with the title “An immunohistochemical study of p21 and p53 expression in primary node-positive breast carcinoma”.

- Tier 5:

  The boolean expression from Tier 4 is matched against the chemical list metadata

> The histological grade of chondrosarcoma correlates well with their clinical behavior and with the patient's survival duration. We have previously demonstrated that **p21** was expressed in the hypertrophic chondrocytes of the growth plate. To assess the relationship of **p21** (**waf1/cip1**) to cell differentiation in chondrosarcoma, we examined the **p21** expression in 14 cases of chondrosarcoma immunohistochemically and the induction of **p21** by insulin-like growth factor-I (IGF-I) during cell differentiation in SW1353 chondrosarcoma cells. **p21** immunoreactivity was seen in well-differentiated chondrosarcoma cells and was mutually exclusive with MIB1 reactivity in grade-1 chondrosarcoma. ...

Figure 3.5: Part of the abstract for a document retrieved using the boolean expression.

field.

Example: The boolean expression in Figure 3.4 retrieves documents in which the phrase "Cip1 protein" appears in the chemical list.

- Tier 6:

  The boolean expression from Tier 4 is matched against the abstract metadata field.

  Example: Figure 3.5 shows part of the abstract of a document retrieved by matching the boolean expression of Figure 3.4 against the abstract metadata field.

In addition, the documents are restricted to those in which the name of the species appears in the MeSH Heading metadata field. This filtering does not completely eliminate documents which are not relevant to the species, since it is possible for the name of the species to appear in the MeSH field even if the focus of the paper is another species. It is quite common for an article about a gene in one species to mention a homologue in a related species. Nevertheless, if the name of the wanted species does not appear in the MeSH heading, then the article is (almost certainly) not relevant. Thus, using species data in the MeSH metadata field may result in false positives but not (or rarely) in false negatives.

The Query Tiering subsystem can return three types of results:

- All Tiers:

  Retrieve documents from all the tiers. Documents retrieved by each tier are ranked ahead of all documents retrieved by the next tier. (A document that is retrieved in more than one tier is counted towards only its highest tier.)

- Best Tier:

  Retrieve the documents in the first tier that contains a non-zero number of documents. Subsequent tiers are ignored.

- Exact:

  Retrieve only documents in Tier 1. No documents are retrieved if there are no documents in Tier 1.

Note that for some topics, this subsystem may retrieve no documents. In the complete

MultiText for Genomics system, the complete runs supplement the document sets retrieved by the Query Tiering subsystem with documents from other subsystems.

3.3 The Fusion Subsystem: Multiple Evidence Combination

The Okapi and Query Tiering subsystems are essentially autonomous and retrieve two independent sets of documents. By merging the two result sets, we obtain a single set of documents with a high precision. We implemented two different methods of combining the two document sets from the two previous subsystems:

In the following, assume that we have two lists of documents, $M = \{m_1, m_2, m_3, \ldots\}$ and $N = \{n_1, n_2, n_3, \ldots\}$, where m_i and n_j are documents and the subscript denotes the rank of the document within the list.

- Interweave:

  The two document sets are combined by taking one document from each set successively. That is, the interweave of M and N is $L = \{l_1, l_2, l_3, \ldots\}$ where

$$l_i = \begin{cases} m_{\frac{i+1}{2}} & \text{if } i \text{ is odd} \\ n_{\frac{i}{2}} & \text{if } i \text{ is even} \end{cases}$$

  Duplicate occurrences of the same document are removed from L; that is, if $l_i = l_j$ and $i < j$, then l_j is removed from L.

- Rank Fusion:

  To merge two sets of documents using rank fusion, the documents which were retrieved by both methods are first merged together. Each document is assigned a score that is the weighted sum of its (reverse) rank in each document set. The combined documents are followed by interweaving the remainder of the two document sets.

  More formally, if i is the rank of the document $m_i \in M$ and j is the rank of the document $n_j \in N$, then $L_1 = M \cap N$, and the score of a document $d \in L_1$ is $s_{L_1}(d) = k_1 \times (R - i) + k_2 \times (R - j)$, where k_1 and k_2 are weights and $R = 1000$ is the number of documents retrieved by each method. Let $M' = M \setminus L_1$ and $N' = N \setminus L_2$ be the remainders of the documents from M and N respectively (the documents retrieved by each method but not by both). Let L_2 be the interweave (as defined above) of M' and N'. Then the (weighted) rank fusion of M and N is $L = \mathit{append}(L_1, L_2)$.

  Merging three document sets is done in an analogous manner.

We also attempted other types of fusion, based on those of Fox and Shaw [FS93]. However, the above techniques seemed to work very well during testing and were the only ones which were fully implemented. Note that the weight rank fusion is a weighted version of the CombSUM formula described by Fox and Shaw.

3.4 The Feedback Subsystem: Query Expansion

The entries of the RN metadata field in the MEDLINE record comprise a list of chemicals mentioned in the document. A match between one of these chemicals and the query is a very good indication that the document is relevant. However, because a gene may have an alias that differs significantly from any of its known names, it is not always possible to identify the query gene in the chemical list using string matching alone.

Instead of attempting to recognize these name variants, we try to learn the variant name by using pseudo-relevance feedback. If the gene name was matched in the first tier in the Query Tiering subsystem, then the chemical list in the top retrieved documents already contains the gene name, and so feedback is unnecessary. Otherwise, we score the chemicals in the top retrieved documents using a Tf-Idf formula, and retrieve an additional set of documents containing the top chemical. The chemical names in the top documents were assigned a score using the formula:

$$w_i = R_i \cdot \left(\log\left(\frac{N}{f_i}\right)\right)^{\alpha}$$

For a chemical i, R_i is the number of times the chemical name appears in the chemical list of the top documents, f_i is the number of times it appears in the corpus, N is the total length of all documents in the corpus, and w_i is the score assigned to i. The formula was developed experimentally, based on the standard Tf-Idf (term frequency, inverse document frequency) idea [SB88]. The chemical names that appear frequently in the top documents

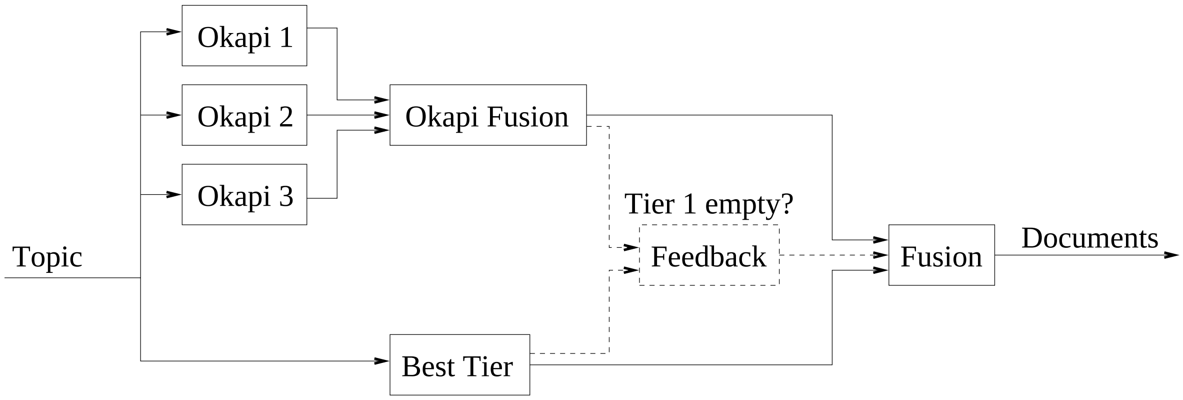

Figure 3.6: Flow diagram for the combined MultiText for Genomics system.

are more likely to be relevant, which is reflected in the "term frequency" part of the equation. On the other hand, those chemical names that appear frequently in the corpus (such as "DNA" which is ubiquitous) are unlikely to be uniquely relevant to the top documents, and their scores are attenuated by the "inverse document frequency" part of the equation. For the MultiText for Genomics system, we used a value of $\alpha = 3$.

The highest scoring chemical name is then used to retrieve a set of documents containing that name. This document set is then merged with the results from the previous subsystems to produce the final document set.

3.5 The Combined MultiText for Genomics System

The combined MultiText for Genomics system consists of the four subsystems described above. The Okapi and Query Tiering subsystems occur in parallel, and depending on the outcome of the Query Tiering subsystem, the Feedback subsystem may be activated. The resultant document sets are then merged to produce the final output of the system.

Each combination of techniques and parameters is called a *run*. Following the TREC standard procedure, 1000 documents were retrieved for each run. The runs which we used in our final system are as follows:

- Okapi 1, 2, 3, and Fusion: These are the document sets retrieved by the procedure described in Section 3.1.

- All Tiers (AT): This is the set of documents retrieved by using the All Tiers method as described in Section 3.2. The documents retrieved by Okapi Fusion are appended to the end.

- All Tiers Interweave-fusion (ATI): The set of documents retrieved by All Tiers is interweaved with the Okapi Fusion document set as described in Section 3.3.

- All Tiers Rank-Fusion (ATR): The set of documents retrieved by All Tiers is combined with the Okapi Fusion document set using the weighted rank fusion method as described in Section 3.3. It was experimentally determined that good results can be obtained if the Okapi rank was weighted 4 times as heavily as the Query Tiering rank.

- All Tiers Interweave/Rank-fusion with Feedback (ATIF, ATRF): These are the same as ATI and ATR, respectively, except that the feedback procedure described in Section 3.4 was used if no documents were retrieved in Tier 1.

- Best Tier (BT, BTI, BTR, BTIF, BTRF): These are analogous to the above, except that the Query Tiering subsystem retrieved only documents from the first tier with non-zero documents.

- Exact: Instead of all the tiers or the best tier, only Tier 1 was used to retrieve documents. The Okapi Fusion document set was then appended to the end. (If no documents were retrieved in Tier 1 for a topic, then the final set of retrieved documents is just the set retrieved by Okapi Fusion.)

- ExactI: The set of documents retrieved by Tier 1 is interweaved with the Okapi Fusion set.

Figure 3.6 shows the flow diagram of the combined system for the BTRF (Best Tier, Rank-fusion, Feedback) run. The topic is sent to both the Okapi and Query Tiering subsystems, each of which returns a set of documents. If the first tier to retrieve a non-zero number of documents is Tier 1, then the two document sets are fused in the Fusion subsystem. Otherwise, a third set of documents is retrieved using the Feedback subsystem, and the three sets of documents are merged together. The other runs follow a similar logic flow.

The parameters of the various runs were optimized for the training data, using the supplied relevance judgments. Thus, the performance of the IR system on the training data is not necessarily reflective of its performance on the test data, especially if the training and test data have different characteristics. In particular, the relative performance of some of the runs that relied on a single retrieval technique may not be necessarily preserved. Nevertheless, the runs involving fusion and feedback do seem to consistently outperform the systems on which they are based. The parameters for these runs were adjusted not only to maximize performance, but to increase stability as well.

The performance of feedback is dependent on the number of top documents used to determine the most relevant chemical name, and on the type of fusion used to merge the three document sets. These parameters are in turn dependent upon the query tiering technique used. For the All Tiers technique, it was determined that using the top 25–30 documents to determine the most relevant chemical name produced the best performance. (The value of 27 was used in the experiments.) The three document sets are fused using rank fusion with equal weights. For the Best Tier technique, the top 42 documents were used, and the three document sets were merged using weighted rank fusion with a weight of 5 for the query tiers document set, 28 for the feedback document set, and 20 for the Okapi Fusion document set. These numbers were determined experimentally.

The reason for the difference between the feedback parameters of the AT and BT runs is that more of the top documents retrieved by the Best Tier technique are relevant compared to those retrieved by All Tiers. Since feedback is only used when no documents are retrieved in Tier 1, the set of documents retrieved using the top chemical name will be far more relevant than the documents retrieved by the Best Tier, and slightly more relevant than retrieved by Okapi.

The Exact and ExactI runs were experiments designed to test the effects of ignoring all subsequent tiers if no documents are retrieved by Tier 1. Early experiments showed that it performed better than All Tiers on those topics for which a match was found in Tier 1, and worse otherwise. Because the performance was unstable, and because Best Tier seemed to always perform better, the full set of fusion and feedback experiments were not performed on the Exact run.

We examine the experimental results on the training and test data in further detail in the next chapter.

Chapter 4

Experimental Results

4.1 Results on Training Topics

The values of the parameters of the MultiText for Genomics system were tuned using the training data. Once these values had been decided upon, we conducted each of the runs on the training data to obtain the final results which are shown in Table 4.1. The results of the Wilcoxon paired-T significance test for certain pairs of runs on the training data are shown in Table 4.2.

As can be seen from Table 4.1, the best average precision belonged to the BTRF run, at 0.4821. This is a 47.3% improvement over the baseline Okapi 1 ($p < 0.001$), which had an average precision of 0.3273. The BTIF run had an average precision of 0.4812, a 47.0% improvement ($p < 0.001$), and the ATRF run had an average precision of 0.4598, a 40.5% ($p < 0.001$) improvement. The ATRF run retrieved 291 relevant documents, which was

Method Used	Rel. & Ret.	Avg. Precision	R-Precision
Okapi 1	224	0.3273	0.3077
Okapi 2	245	0.3193	0.2917
Okapi 3	261	0.3157	0.2700
Okapi Fusion	261	0.3321	0.3173
AT	282	0.3819	0.3452
ATI	282	0.4394	0.3836
ATIF	289	0.4429	0.3844
ATR	284	0.4519	0.4324
ATRF	**291**	0.4598	0.4448
BT	279	0.4003	0.3818
BTI	279	0.4528	0.4236
BTIF	286	0.4812	0.4448
BTR	279	0.4452	0.4216
BTRF	286	**0.4821**	**0.4579**
Exact	277	0.3981	0.3820
ExactI	277	0.4246	0.3959

Table 4.1: Summary of Results on Training Data: 50 topics, 1000 retrieved per query, 335 total relevant.

the most relevant documents retrieved of all the runs. This is slightly more than the 286 retrieved by BTRF and BTIF, and considerably more than the 224 retrieved by the Okapi 1 run.

Among the Okapi runs, more relevant documents were retrieved by Okapi 3 than by Okapi 2, which in turn retrieved more relevant documents than Okapi 1. Performance, however, was in the reverse order, with Okapi 1 having the best average precision of the three. Figure 4.1 shows the precision-recall curves[1] for the Okapi runs on the training data. As is typical for such curves, the precision and recall are inversely related for each of the Okapi runs. As can be seen, at lower recall levels (when fewer documents have been retrieved) Okapi 1 has the highest precision, and Okapi 3 has the lowest, with Okapi

[1]Note that this and subsequent precision-recall curves have been scaled to show the precision range $0.1 - 0.7$ for the sake of clarity.

	Runs Compared		p-value
Okapi	Okapi 1	Okapi Fusion	0.061
	Okapi 2	Okapi Fusion	0.097
	Okapi 3	Okapi Fusion	0.23
Query Tiering	Okapi 1	AT	0.14
	Okapi 1	BT	0.089
	Okapi 1	Exact	0.012
	Okapi Fusion	AT	0.15
	Okapi Fusion	BT	0.14
	Okapi Fusion	Exact	0.057
Fusion	AT	ATI	0.035
	AT	ATR	0.074
	BT	BTI	0.037
	BT	BTR	0.091
	Exact	ExactI	0.18
Feedback	ATI	ATIF	0.45
	ATR	ATRF	0.50
	BTI	BTIF	0.083
	BTR	BTRF	0.025
Feedback vs. Baseline	Okapi 1	ATIF	< 0.001
	Okapi 1	ATRF	< 0.001
	Okapi 1	BTIF	< 0.001
	Okapi 1	BTRF	< 0.001
Chosen Runs	ATRF	BTRF	0.11

Table 4.2: Wilcoxon paired-T test results on runs for training data.

2 in the middle. However, as the recall level increases (when more documents have been retrieved) the relative positions of the three runs are reversed. By using bigrams, the Okapi 3 system was able to retrieve more relevant documents, but they were ranked lower as it also retrieved many irrelevant documents which it ranked highly. This suggests that the gene names in the corpus are actually very close to how they appear in the LocusLink-derived query.

The Okapi Fusion run both retrieved more relevant documents and achieved a better performance than each of the individual Okapi runs. It retrieved as many documents as Okapi 3 did, while its average precision was 0.3321, a 1.4% ($p = 0.061$) improvement over Okapi 1. While the gain is not significant, it nevertheless demonstrates that an improvement in retrieval can be made simply by reformulating the query and merging the documents retrieved using different query formulations. More importantly, retrieval using the fusion technique is more stable than any of Okapi 1, Okapi 2, or Okapi 3 alone. The first three rows of the Table 4.2 compare each of the Okapi runs to the Okapi Fusion run. As can be seen in Figure 4.1, the Okapi Fusion run is outperformed by Okapi 1 when the recall level is low, and by Okapi 3 when the recall level is high, but performs better than each of the individual Okapi runs at the intermediate recall level. This suggests that a good strategy for merging the results of the individual Okapi runs should weigh Okapi 1 more heavily at first but gradually increase the dominance of Okapi 3 as more documents are retrieved.

Feedback and fusion improved performance in every case, and the systems with the best performance made use of both. It isn't clear which fusion method is better, since ATR outperformed ATI, but BTI did better than BTR. However, when fusion is used with feedback, the rank fusion method outperformed the interweave fusion method in both cases. Figures 4.2 and 4.3 show the precision-recall curves for the All-Tiers and Best-Tier runs. At low recall levels, ATI and ATIF outperform ATR and ATRF, but the reverse is true at high recall levels. Similarly, BTI and BTIF outperform BTR and BTRF at low recall levels, but while BTRF outperforms BTIF at high recall levels, the performance of BTI and BTR are similar. This suggests that, when few documents have been retrieved, the Okapi and Query Tiering subsystems retrieve different relevant documents, and good results may be obtained simply by interweaving the two document sets. However, as more documents

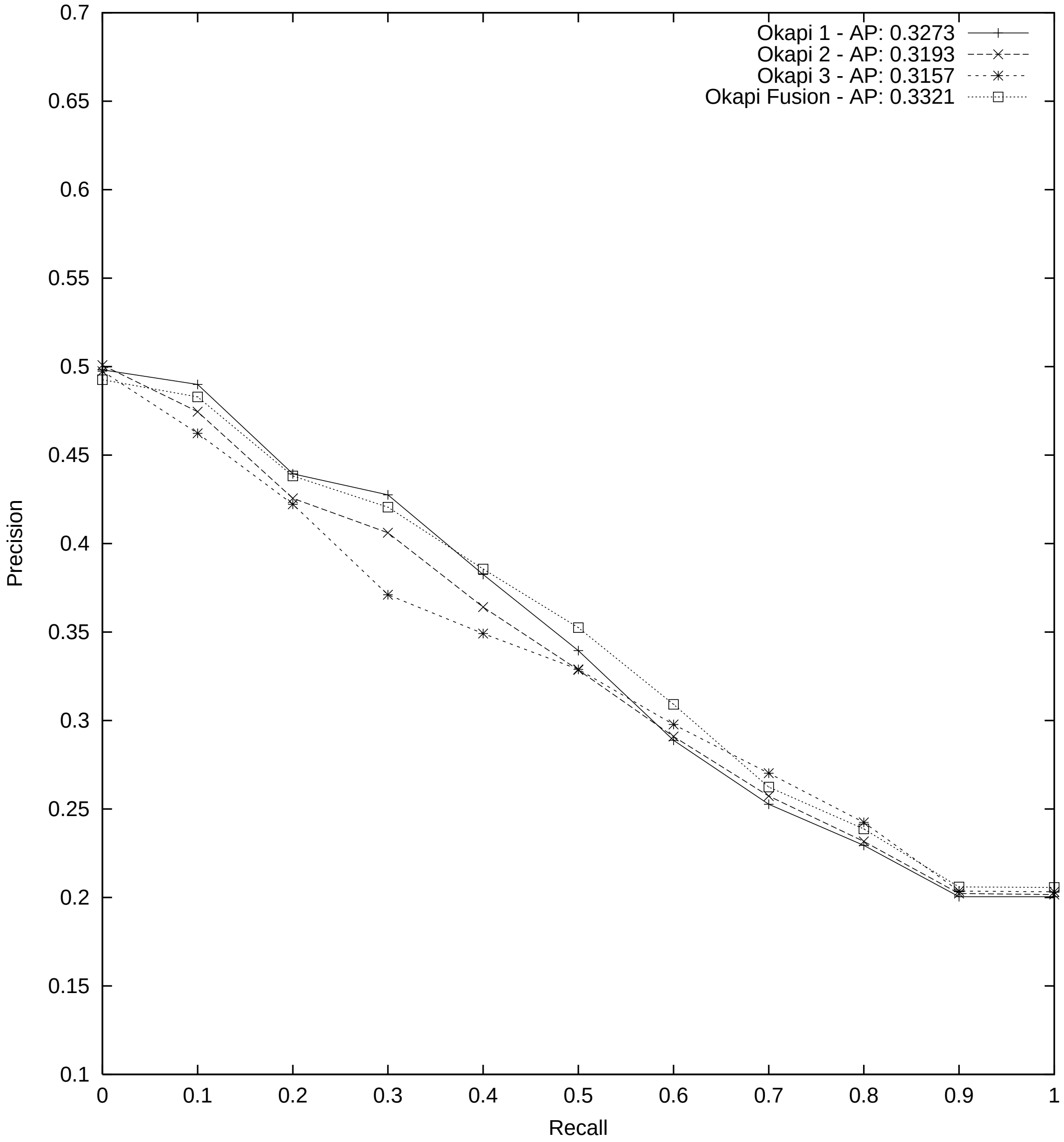

Figure 4.1: Precision-recall curves for the Okapi runs on the training data.

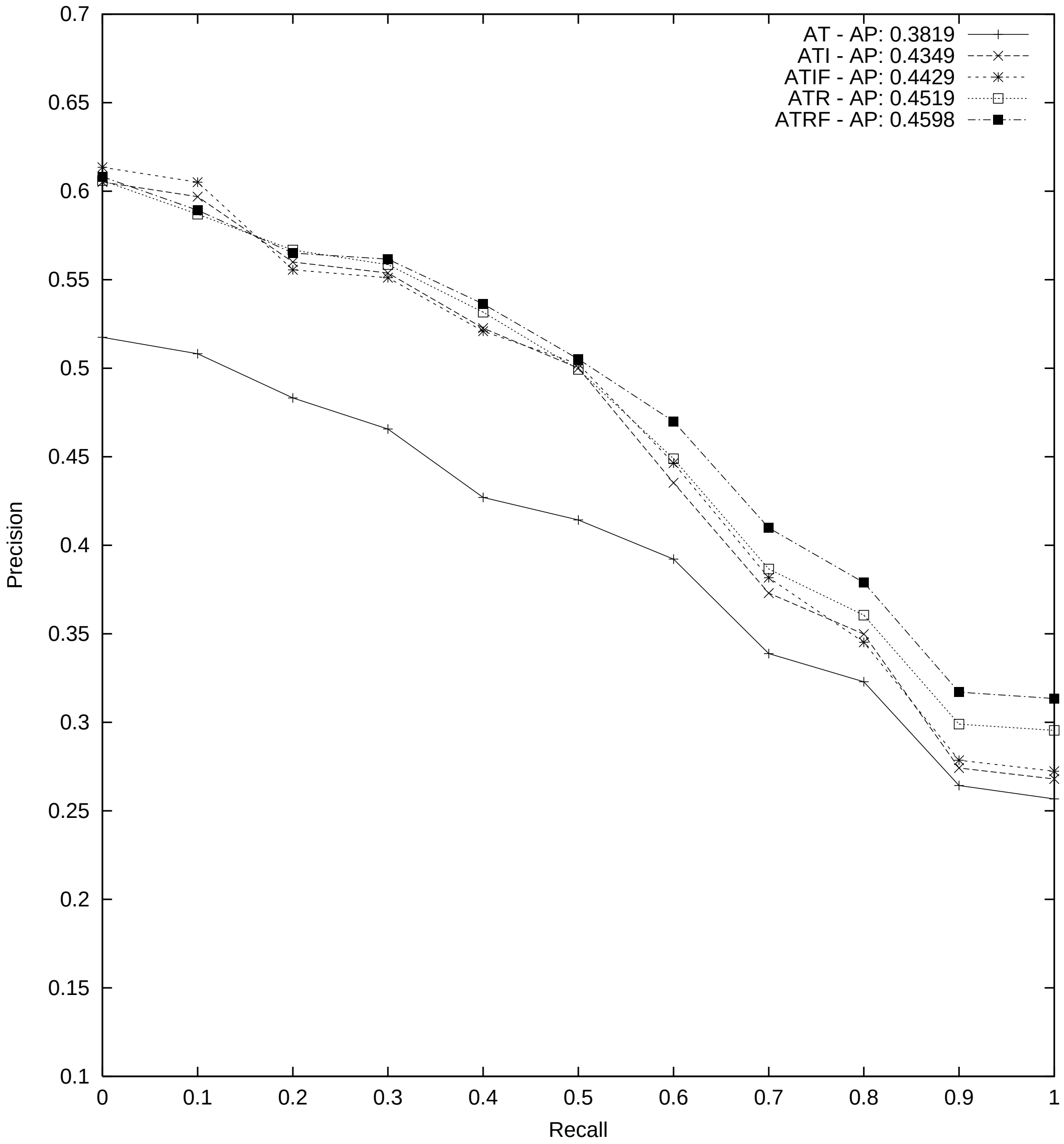

Figure 4.2: Precision-recall curves for the All-Tiers runs on the training data.

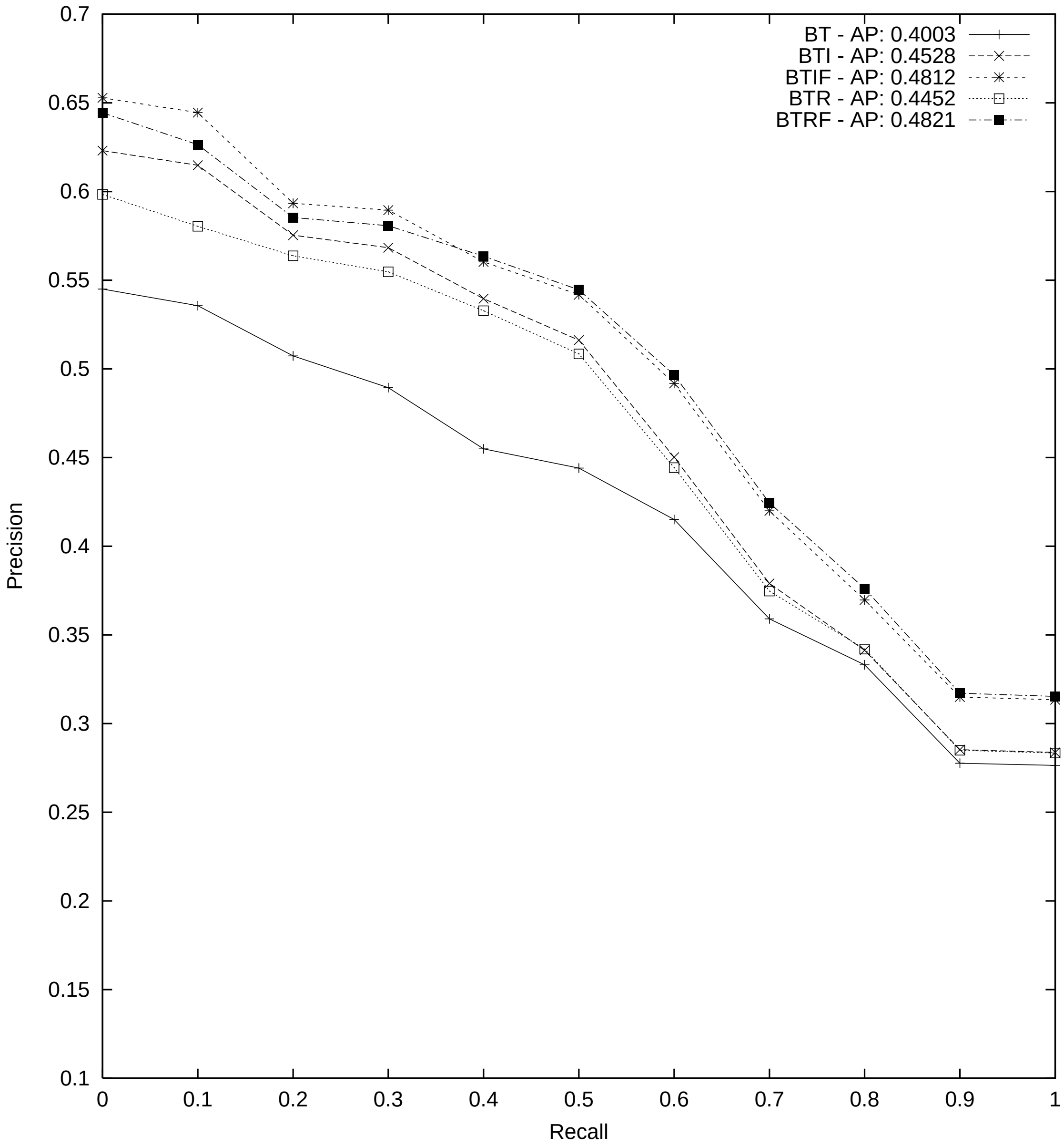

Figure 4.3: Precision-recall curves for the Best-Tier runs on the training data.

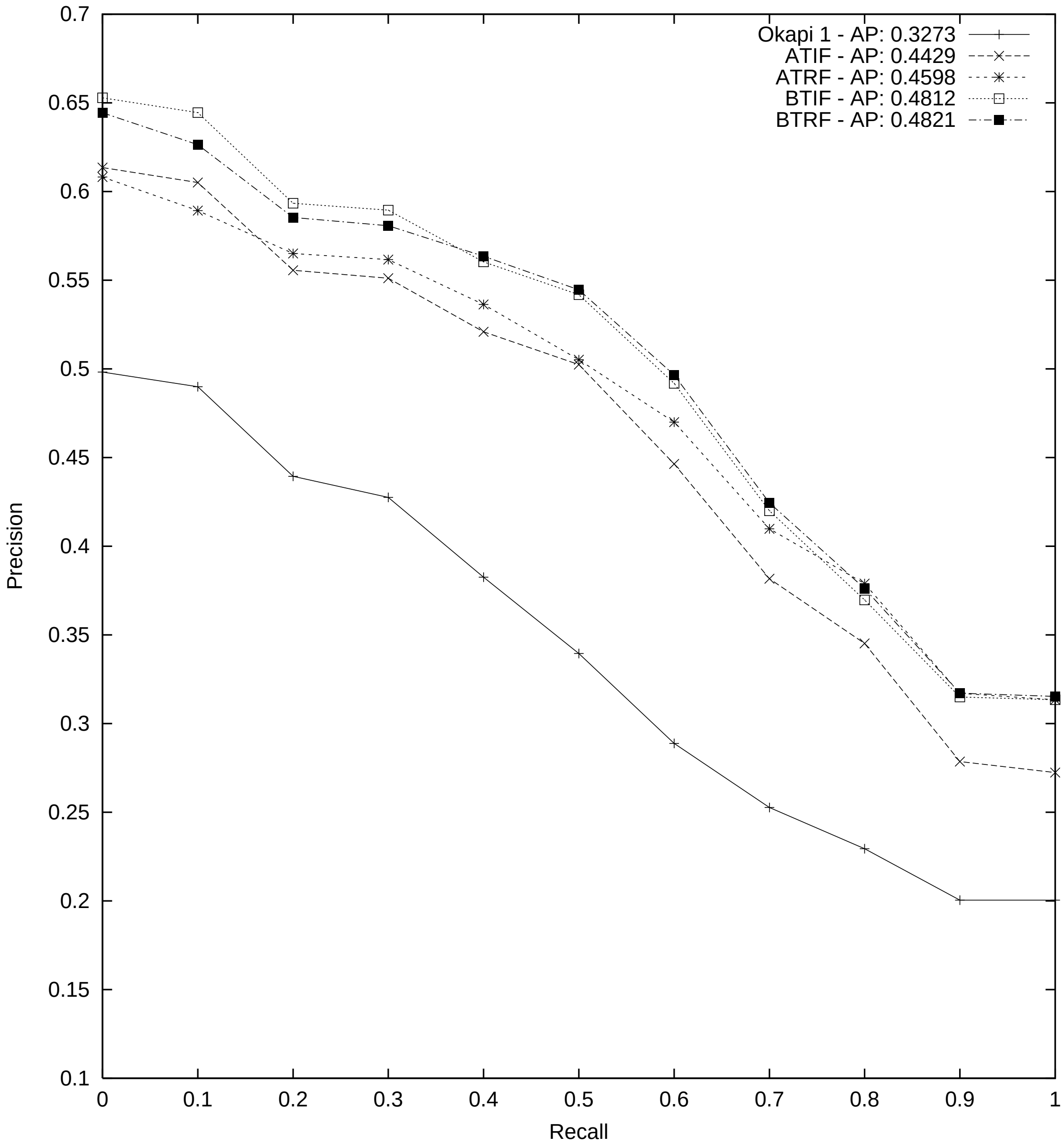

Figure 4.4: Precision-recall curves for the training runs using feedback.

are retrieved, the Okapi methods retrieve more relevant documents than the Query Tiering methods, even if it gives these relevant documents a low rank, and so documents in the Okapi document set should be favoured.

Table 4.2 shows that the improvement to the retrieval due to the fusion of the Okapi and Query Tiering subsystems over the Query Tiering by itself is not very significant. Furthermore, it shows that the improvements due to using feedback are not significant at all for the All-Tiers runs while they are somewhat significant for the Best-Tier runs. Since feedback is used only when the set of documents retrieved by Tier 1 in the Query Tiering subsystem is empty, the Best Tier runs are more sensitive to its effects.

There is a high level of correspondence between the metadata fields and the relevance of the documents. This is clear from the fact that retrieval using query tiers based on the information in the metadata fields outperformed the Okapi runs, including the Okapi Fusion run. Before fusion and feedback, the best technique that is based on query tiers is BT, with an average precision of 0.4003, which is a 22% improvement ($p = 0.089$) over Okapi 1. The Exact run had an average precision of 0.3981, a 21% improvement ($p = 0.012$), while the AT run had an average precision of 0.3819, which close to 17% ($p = 0.14$) over Okapi 1. Note that both Best Tier and Exact had a better average precision than the All Tiers method. It appears that once a match has been found in a tier, it was a better strategy to append the Okapi Fusion list rather than documents from lower tiers. The experimental results suggest that the performance of the Okapi Fusion method was between that of Tier 1 and Tier 2.

Figure 4.4 shows the recall-precision curve for the runs with feedback, with the Okapi 1 run shown as a baseline for comparison. Table 4.2 shows that each of Query Tiering, Fusion, and Feedback improve upon the Okapi runs.

Table 4.3 shows the documents retrieved in each tier for the 50 training topics. The topic number is shown in the first column, followed by six columns showing the number of documents retrieved in each of the six tiers. The last column contains the expression or expressions used in the first tier in which a match was made.

In 32 out of 50 topics, the best tier was Tier 1. Of the remaining topics, Tier 2 was the best tier in 4 topics, Tier 3 was best in 8, and Tier 4 was best in 4. No documents were retrieved at all in Tier 5, and Tier 6 was the best tier for 1 topic. The reason that Tier 5 was included at all is that the tiers were developed independently and had been re-arranged during training. In the final arrangement of the query tiers, it happened that every document retrieved by Tier 5 had already been retrieved in a higher tier.

Because Tier 1 had a better performance on its own than Okapi or even feedback, performance can be improved by recognizing relevant chemical names in the chemical list metadata, even in cases where the name of the gene and the relevant chemical name are different.

Table 4.4 shows the chemical names produced by the pseudorelevance feedback for those topics in which no documents were retrieved in Tier 1, for the BTRF run. The first column gives the topic number, and the second column gives a gene name from the query. The third column shows the chemical name that was found using automatic query expansion.

Topic	Number of Documents Retrieved						Matches in Best Tier
	T1	T2	T3	T4	T5	T6	
1	**438**	120	0	19	0	482	“cip1 protein”
2	**6**	13	38	4	0	28	“rna dependent atpase”, “protein p68”
3	**19**	31	0	5	0	43	“tel protein”
4	**35**	2	499	2	0	75	“keratinocyte growth factor”, “fibroblast growth factor 7 precursor”, “fibroblast growth factor 7”
5	**16**	0	23	0	0	6	“glycine receptor alpha1”
6	**93**	10	0	2	0	101	“hla dqb1”
7	**56**	3	44	0	0	39	“janus kinase 2”
8	–	–	–	**8**	0	50	((“luteinizing”ˆ“hormone”ˆ“choriogonadotropin”ˆ“receptor”)+“lhcgr”+“lcgr”+“lhr”+ (“luteinizing”ˆ“hormone”ˆ“receptor”)+(“lutropin”ˆ“choriogonadotropin”ˆ“receptor”)+“lcgrs”+ “lhcgrs”+(“luteinizing”ˆ“choriogonadotropin”ˆ“receptor”)+“lgr2”+“lhrs”+(“lutropin”ˆ“receptor”)+ (“choriogonadotropin”ˆ“receptor”))
9	**15**	1	68	12	0	345	“growth inhibitory factor”
10	**161**	360	757	480	0	785	“protein c”
11	–	**80**	0	0	0	117	“rac1”
12	**3**	0	41	0	0	11	“tropomyosin 1”
13	**3**	0	3	7	0	163	“gpcr protein”, “frizzled 4 protein vertebrate”
14	–	–	–	**10**	0	408	((“tyrosyl”ˆ“trna”ˆ“synthetase”)+“tyrrses”+“ytses”+“yts”+ (“tyrosyl”ˆ“trna”ˆ“ligase”)+“yars”+“tyrrs”+“yarses”+“yrses”+“yrs”)
15	**11**	1	0	13	0	109	“major vault protein”
16	**4**	0	80	0	0	0	“adrenergic receptor alpha 1d”, “adrenergic receptor alpha 1a”
17	–	**10**	0	0	0	0	“rhob”
18	**213**	0	205	2	0	73	“cpp32 protein”
19	**6**	0	0	0	0	6	“ctcf protein”
20	**162**	0	979	2	0	68	“fasl protein”
21	–	–	**1**	2	0	44	(((“ig”)))
22	–	–	–	**4**	0	14	(“ihhs”+(“indian”ˆ“hedgehog”)+“ihh”)
23	–	–	**47**	1	0	16	(((“phospholipase”+“phospholipases”))ˆ“c gamma”)
24	–	–	**3**	0	0	0	(((“seven”+“sevens”)ˆ(“absentia”+“absentias”)))
25	–	–	–	**3**	0	112	(“dntts”+“tdt”+“dntt”+(“terminal”ˆ“deoxynucleotidyl”ˆ“transferase”)+ (“deoxynucleotidyltransferase”ˆ“terminal”)+“tdts”)
26	–	–	–	**1**	0	1	((“rho”ˆ“related”ˆ“btb”ˆ“domain”ˆ“containing”ˆ“2”)+“rhobtb2”+“kiaa0717”+“dbc2”)
27	–	–	–	–	–	**19**	((“cholinergic”ˆ“receptor”ˆ“muscarinic”ˆ“3”)+“chrm3”)
28	–	**11**	0	9	0	57	“egr1”, “ngfi”
29	**19**	1	0	0	0	8	“glucokinase”
30	**2**	0	40	0	0	1	“retinoic acid receptor gamma”
31	**149**	4	460	9	0	93	“neurokinin a”, “substance p”, “neuropeptide k”
32	–	–	**186**	4	0	75	(((“estrogen”+“estrogens”)ˆ(“receptor”+“receptors”)))
33	–	–	**70**	0	0	21	(((“guanylate”+“guanylates”)ˆ(“cyclase”+“cyclases”)))
34	**20**	1	0	0	0	2	“cocaine and amphetamine regulated transcript protein”
35	–	–	–	–	–	–	–
36	**5**	0	9	2	0	6	“hop protein”
37	**1**	0	0	0	0	1	“slob protein”
38	**3**	0	0	0	0	0	“eiger protein drosophila”
39	**32**	1	7	1	0	15	“cadherins”
40	**6**	0	0	3	0	2	“stat92e protein”
41	**3**	0	0	0	0	3	“ebony protein”
42	**10**	0	0	0	0	5	“crb protein drosophila”
43	–	–	**3**	11	0	422	(((“calcineurin”+“calcineurins”)))
44	**3**	0	4	0	0	0	“gp73 protein”
45	**5**	1	3	2	0	5	“sh3px1 protein”, “wisp protein”
46	–	**7**	0	5	0	16	“hanks”, “ank”
47	**2**	0	0	0	0	0	“dda3 protein”
48	**10**	0	0	10	0	323	“artemis protein human”
49	–	–	**1000**	67	0	947	(((“transcription”+“transcriptions”)ˆ(“factor”+“factors”)))
50	**1**	0	2	0	0	1	“pax 8 protein”
Total	32	4	8	4	0	1	

Table 4.3: Matches in the query tiers for the training topics.

Topic	Query Term/Phrase	Feedback Chem. Name	Ret.	R.&R.	MAP	R-P	MAP Fb.	R-P Fb.	Imp.
8	luteinizing hormone/ choriogonadotropin receptor	Receptors, LH	49	7	0.2917	0.4286	0.4305	0.4286	+47%
11	ras-related C3 botulinum toxin substrate 1 (rho family, small GTP binding protein Rac1)	rac1 GTP-Binding Protein	80	13	0.2302	0.4118	0.1977	0.1765	-14%
14	tyrosyl-rRNA synthetase	Tyrosine-rRNA Ligase	10	6	0.5872	0.5000	0.8238	0.6667	+40%
17	ras homolog B (RhoB)	rhoB GTP-Binding Protein	6	2	0.3333	0.3333	0.3889	0.6667	+17%
21	immunoglobulin heavy chain 6 (heavy chain of IgM)	Immunoglobulins, mu-Chain	21	0	–	–	–	–	–
22	Indian hedgehog	hedgehog protein, vertebrate	69	6	0.4703	0.5000	0.6723	0.5000	+43%
23	phospholipase C, gamma 1	phospholipase C gamma	47	9	0.6503	0.5556	0.5262	0.4444	-19%
24	seven in absentia 2	seven in absentia protein	3	2	1.0000	1.0000	1.0000	1.0000	0%
25	terminal deoxynucleotidyl transferase	DNA Nucleotidylexotransferase	8	2	1.0000	1.0000	1.0000	1.0000	0%
26	Rho-related BTB domain containing 2	QM protein, Trypanosoma brucei	0	0	1.0000	1.0000	1.0000	1.0000	0%
27	cholinergic receptor, muscarinic 3	Receptors, Muscarinic	153	2	0.0312	0.0000	0.0747	0.0000	+139%
28	Early growth response 1	Krox-24 protein	40	8	0.0258	0.1250	0.2523	0.1250	+878%
32	estrogen receptor 1	Receptors, Estrogen	163	11	0.1039	0.0909	0.1354	0.0000	+30%
33	guanylate cyclase 1, soluble, beta 3	Guanylate Cyclase	70	1	0.0774	0.0000	0.0569	0.0000	-26%
35	CG3599	Drosophila Proteins	638	0	–	–	–	–	–
43	Calcineurin B	Calcineurin	3	1	0.5000	0.0000	1.0000	1.0000	+100%
46	ankylosis, progressive homolog	ankylosis protein	5	3	0.1595	0.0000	0.7500	0.7500	+370%
49	transcription factor 23	Transcription Factors	1000	0	–	–	–	–	–

Table 4.4: Analysis of the effects of feedback on performance for the training topics.

The next two columns show the number of documents retrieved using the chemical name and the number of these which were also relevant. The next two columns show the mean average precision and the interpolated recall-precision, respectively, for that topic without using feedback. (These are equivalent to the MAP and recall-precision for the BTR run.) The next two columns give the mean average precision and interpolated recall-precision with feedback, and the last column gives the percentage improvement (or degradation) due to using feedback. It is apparent that most of the chemical names are related in some way to the gene name, and a better way of recognizing the relationship between a gene and a chemical name will clearly improve performance.

For topic 28, the top chemical name "Krox-24 protein" was produced for the "Early growth response 1". In fact, "Krox-24 protein" is another name for "Early growth response 1". By searching on "Krox-24 protein", which does not appear in the original query, the average precision was improved by an incredible 878%. Of course, the original performance for this topic was very poor, but there is clearly a lot of potential for improving performance by recognizing the alternate names of a gene or a substance related to a gene.

In some cases, this is relatively simple. For topic 14, for example, the chemical name "Tyrosine-rRNA Ligase" was generated for the gene name "tyrosyl-rRNA synthetase". A system that understood the relationship between "tyrosine" and "tyrosyl" and "ligase" and "synthetase" can determine that the two expressions refer to the same thing (or closely related things), and even assign a score for the degree of similarity. In other cases, this is complicated by the fact that more than one chemical name generated by the automatic expansion might be relevant to the query. For topic 27, searching on the gene name "cholinergic receptor, muscarinic 3" resulted in the top chemical name "Receptors, Muscarinic". However, the chemical name "muscarinic receptor M3", which is clearly more relevant, was overlooked. Choosing this chemical name instead of the more general "Receptors, Muscarinic" would have resulted in an improvement of 534%.

As the table shows, in most cases the performance was improved by using feedback to find the most relevant chemical, though in some cases there was a degradation in performance. Determining the conditions under which feedback improved or degraded performance would allow feedback to be used more effectively.

The results on the training data show that the mixture of techniques and the parameters used in the MultiText for Genomics system performs quite well for genomics document retrieval from the MEDLINE corpus.

4.2 Results on Test Topics

Even though the Genomics Track allowed for the submission of only two official runs, we performed the same runs using the test data as we did on the training data, for the purposes of comparing the characteristics of the test and training data as well as to verify the properties we believe the various combinations of techniques to have.

The two runs chosen for official submission to TREC were the BTRF and ATRF runs. The first of these used the Best Tier retrieval method in the Query Tiering subsystem, while the second used the All Tiers retrieval method. Both runs used the Rank Fusion method in the Fusion subsystem. The BTRF run was chosen because it had the highest average precision on the training data, while the ATRF run was chosen partly because it had one of the highest average precisions, but also because it had the highest number of relevant documents retrieved. Even though BTIF had a better mean average precision than ATRF on the training data, it was too similar to the BTRF run in that it differed only in the fusion method used. It was found that by adjusting the fusion weights, it was always possible for the rank-fusion to outperform the interweave fusion. It was also suspected that the ATRF run might be more stable, in the sense that the performance would not be too adversely affected by an incorrect match in Tier 1. Both the ATRF and BTRF runs had

Method Used	Rel. & Ret.	Avg. Precision	R-Precision
Okapi 1	447	0.2060	0.1965
Okapi 2	473	0.2155	0.1948
Okapi 3	524	0.2169	0.2095
Okapi Fusion	524	0.2323	0.2138
AT	550	0.2542	0.1967
ATI	550	0.3334	0.2723
ATIF	559	0.3379	0.2680
ATR	552	0.3425	0.3050
ATRF	**562**	0.3479	0.3013
BT	535	0.2443	0.2010
BTI	535	0.3066	0.2581
BTIF	556	0.3322	0.2745
BTR	535	0.3161	0.2852
BTRF	556	**0.3534**	**0.3113**
Exact	528	0.2500	0.2194
ExactI	528	0.2803	0.2449

Table 4.5: Summary of Results on Test Data: 50 topics, 1000 retrieved per query, 566 total relevant.

a p-value much less than 0.001 when compared with the Okapi 1 baseline run. It would be interesting to examine the trade-off between retrieving more relevant documents and having a better precision.

The results for the various runs on the test data are shown in Table 4.5, and the results of the Wilcoxon paired-T tests shown in Table 4.6. Some similarities and differences between the training and test results may be noted. The two official runs turned out to be excellent choices, as the BTRF and ATRF runs on the test data had the two highest average precisions, at 0.3534 and 0.3479 respectively, corresponding to improvements of 71.5% ($p < 0.001$) and 68.9% ($p < 0.001$) over the Okapi 1 baseline result of 0.2060. The ATRF run retrieved the most relevant documents, with 562 relevant documents retrieved, which is 25.7% more than the 447 retrieved by Okapi 1. Furthermore, ATRF performed better than BTIF, which had an average precision of 0.3322, a 61.3% improvement ($p < 0.001$)

	Runs Compared		p-value
Okapi	Okapi 1	Okapi Fusion	< 0.001
	Okapi 2	Okapi Fusion	0.0086
	Okapi 3	Okapi Fusion	0.056
Query Tiering	Okapi 1	AT	0.088
	Okapi 1	BT	0.016
	Okapi 1	Exact	0.0046
	Okapi Fusion	AT	0.38
	Okapi Fusion	BT	0.68
	Okapi Fusion	Exact	0.20
Fusion	AT	ATI	< 0.001
	AT	ATR	< 0.001
	BT	BTI	< 0.001
	BT	BTR	< 0.001
	Exact	ExactI	0.015
Feedback	ATI	ATIF	0.76
	ATR	ATRF	0.35
	BTI	BTIF	0.017
	BTR	BTRF	0.0077
Feedback vs. Baseline	Okapi 1	ATIF	< 0.001
	Okapi 1	ATRF	< 0.001
	Okapi 1	BTIF	< 0.001
	Okapi 1	BTRF	< 0.001
Official Runs	ATRF	BTRF	0.80

Table 4.6: Wilcoxon paired-T test results on runs for test data.

over Okapi 1.

The distance between ATRF and BTRF was also smaller. Although BTRF showed 4.8% higher average precision in training, the difference was not significant ($p = 0.11$). On the test data the difference diminishes to 1.6% ($p = 0.80$). Thus these tests do not demonstrate any real difference in effectiveness between ATRF and BTRF as measured by average precision. Whereas for the training data, the BT run slightly outperformed the AT run, for the test data the situation is reversed. For the test data, the relevant documents were more likely to be distributed between the tiers rather than be concentrated in the best tier. This suggests that there is more variation in the characteristics identifying relevant documents for the test data than for the training data.

The precision-recall curves for the Okapi runs are shown in Figure 4.5. The Okapi Fusion run performed better than any individual Okapi run, and has a higher precision for almost all recall levels below 0.65, above which Okapi 3 has a higher precision. Of the individual Okapi runs, Okapi 3 had the highest average precision, followed by Okapi 2, and then Okapi 1. This is the *reverse* of the order with the training data. Using bigrams rather than the original query resulted in better performance on the test data. This suggests that with the test data, the gene and protein names in the corpus are less like the LocusLink-derived queries than is the case with the training data. This would also explain the reversal in performance between the AT and BT runs for the training and test data described above.

Figures 4.6 and 4.7 show the precision-recall curves for these runs. The Rank-fusion method seemed to work better for the test data than for the training data. The ATRF run had a better precision than the ATIF run for both high and low recall levels, with ATIF outperforming ATRF only in the range of recall levels from 0.2 and 0.3. The BTRF run outperformed the BTIF run at every recall level. Since the parameters are set such that the Rank-fusion algorithm assigns a heavier weight to the Okapi Fusion document set than it does to the Query Tiering document set, this means that the Okapi subsystem is ranking relevant documents more highly with the test data than with the training data.

A comparison of Table 4.3 with Table 4.7 shows that the matches in the query tiers are more distributed among the tiers for the test data than for the training data. Whereas 32 out of 50 training topics retrieved documents in Tier 1, only 25 test topics did so, with

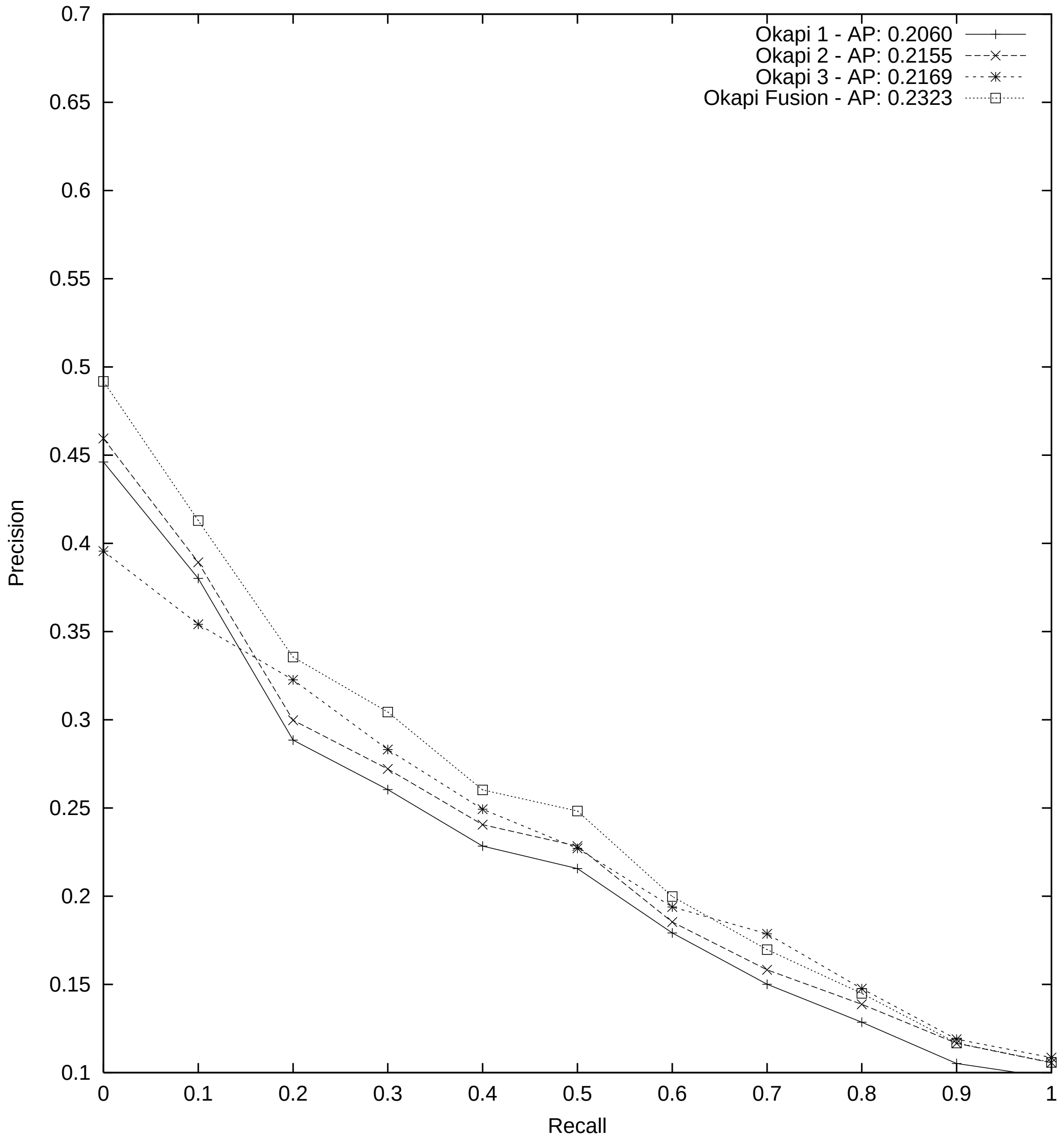

Figure 4.5: Precision-recall curves for the Okapi runs on the test data.

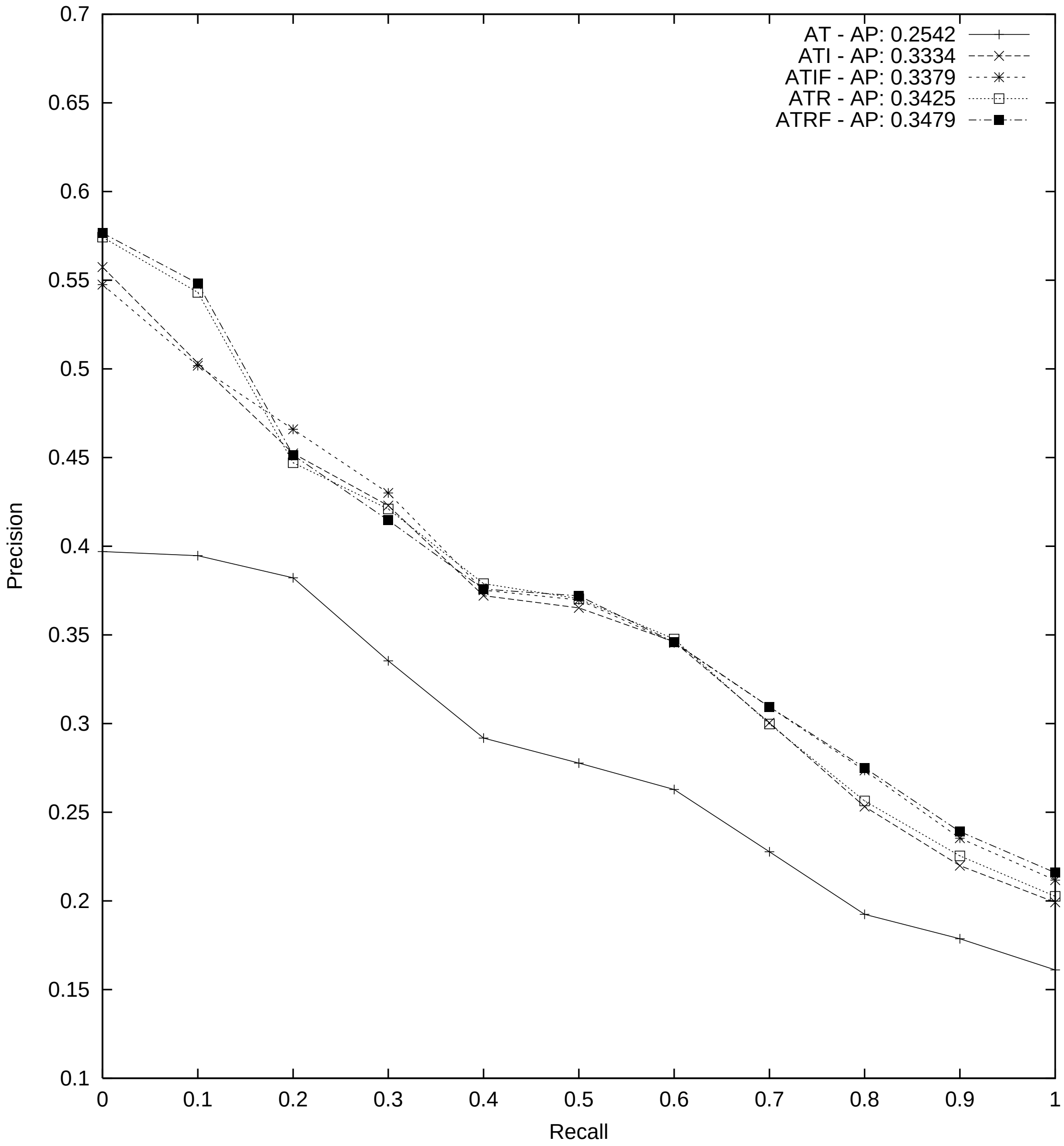

Figure 4.6: Precision-recall curves for the All-Tiers runs on the test data.

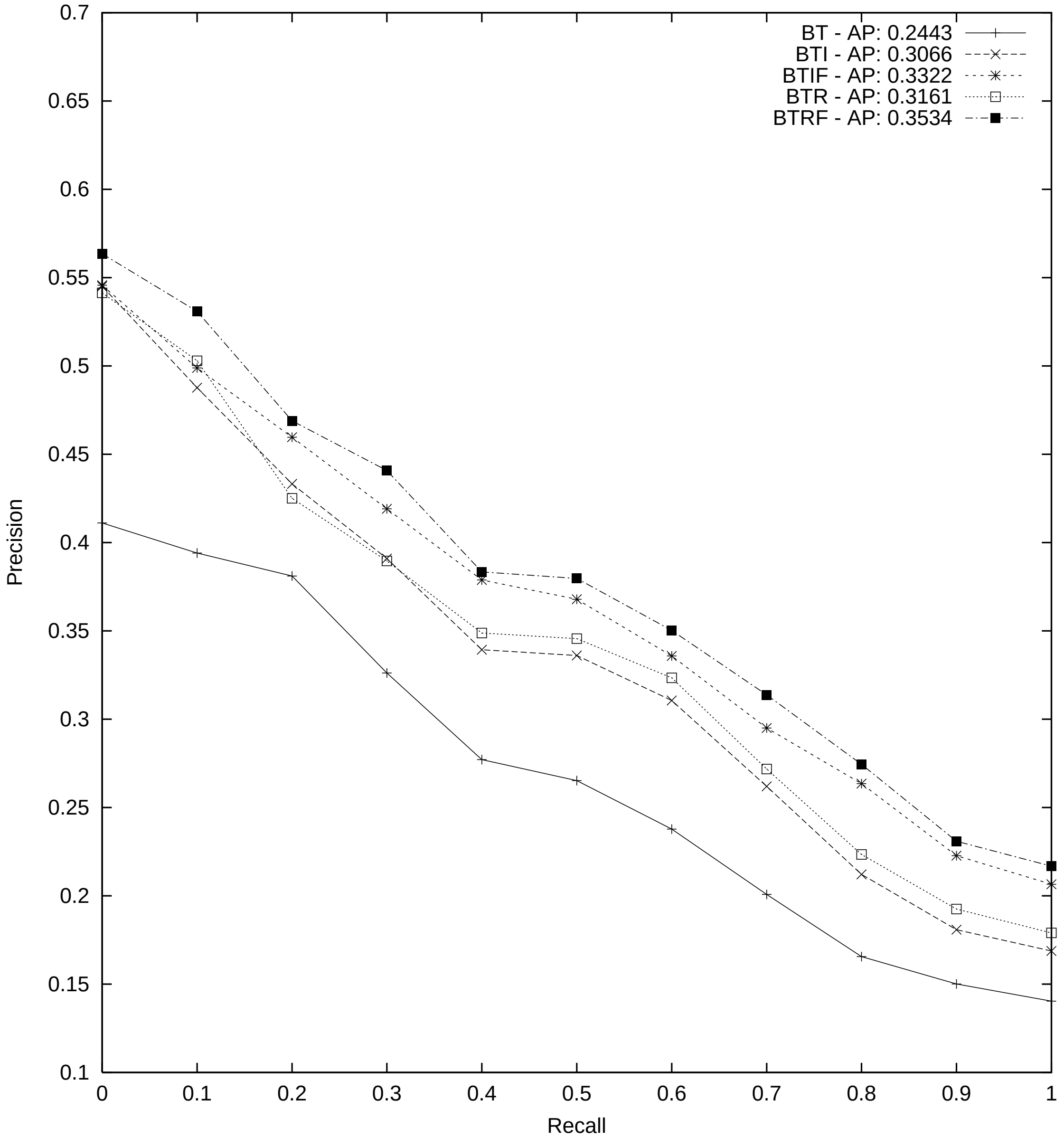

Figure 4.7: Precision-recall curves for the Best-Tier runs on the test data.

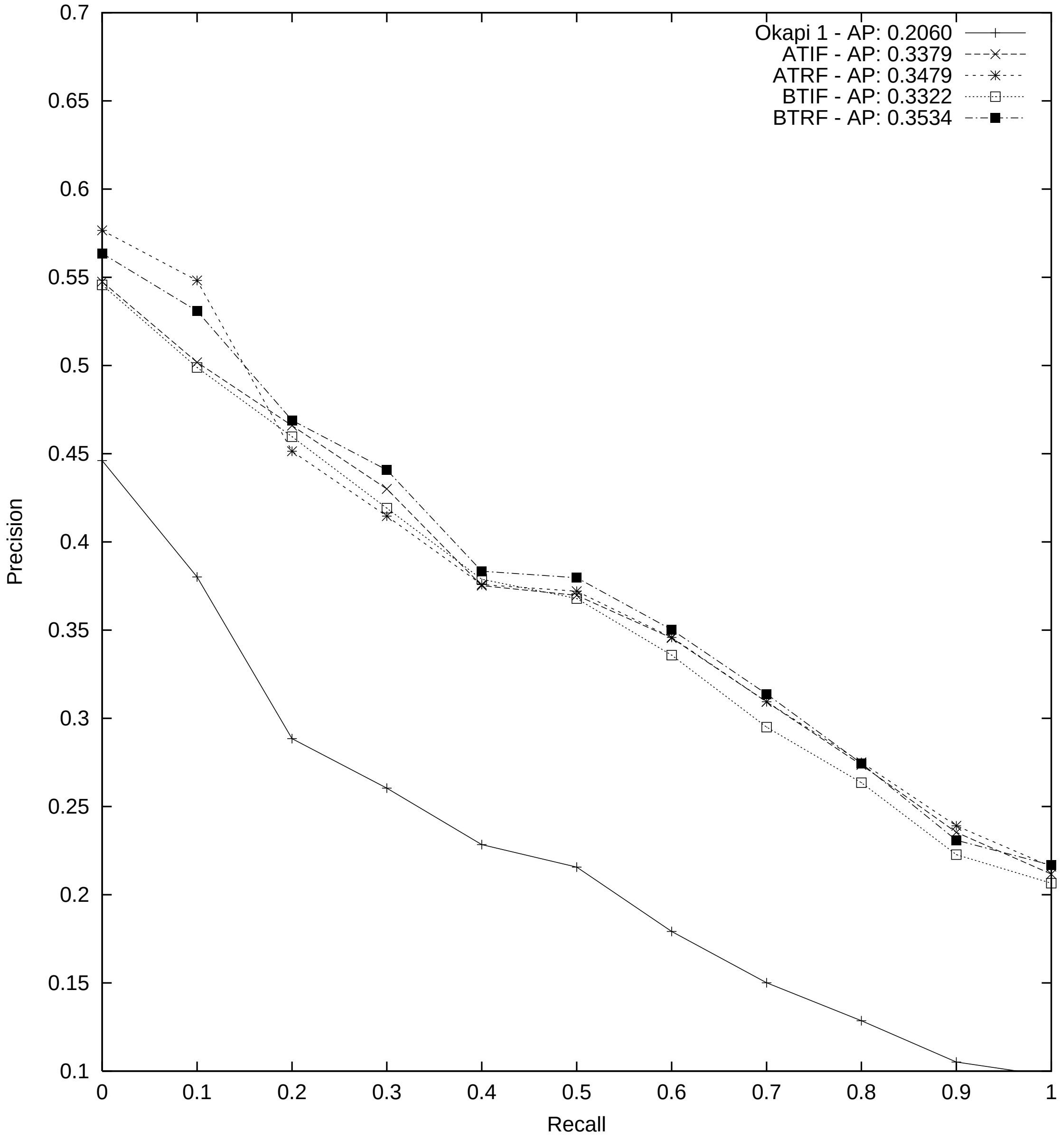

Figure 4.8: Precision-recall curves for the test runs using feedback.

Topic	Number of Documents Retrieved						Matches in Best Tier
	T1	T2	T3	T4	T5	T6	
1	**18**	0	29	1	0	45	“activating transcription factor 2 protein”
2	–	**1**	118	2	0	48	“e2f1”
3	–	**2**	0	13	0	40	“eif4e”
4	–	**146**	62	214	0	891	“g protein”
5	**79**	0	36	2	0	64	“heme oxygenase 1”
6	**145**	4	0	1	0	47	“pten protein”
7	**8**	0	31	0	0	6	“syndecan 4”
8	**18**	0	5	1	0	16	“excitatory amino acid transporter 2”
9	–	**3**	0	28	0	79	“stat5”
10	**102**	0	0	3	0	92	“thrombopoietin”
11	**103**	0	201	0	0	20	“tissue inhibitor of metalloproteinase 2”
12	–	**12**	380	92	0	94	“vdr”, “1 25 dihydroxyvitamin d3”
13	–	**10**	48	1000	0	604	“ah”, “ahr”, “in”
14	**1**	269	**7**	11	0	211	“bcl2 protein mouse”
15	–	**97**	0	2	0	63	“cd34”
16	–	–	–	**5**	0	22	((“heterogeneous”^“nuclear”^“ribonucleoprotein”^“a1”)+“hnrnpa”+“hnrnpa1”+(“hnrnp”^“a”))
17	**33**	0	28	1	0	29	“interleukin 1 receptor antagonist protein”
18	**81**	2	968	4	0	363	“interleukin 5”
19	**12**	0	0	0	0	1	“ptp 1b protein”
20	–	**20**	16	5	0	152	“spa”, “surfactant associated protein a”
21	–	–	**61**	1	0	4	(((“adenylate”+“adenylates”)^(“cyclase”+“cyclases”)^(“activating”+“activatings”)^ (“polypeptide”+“polypeptides”)))
22	–	**86**	0	54	0	205	“di”, “vas”
23	**35**	0	390	1	0	61	“protein kinase c alpha”
24	–	–	–	7	0	17	(“glutbs”+“gtg3”+“glutb”+(“solute”^“carrier”^“family”^“2”^“a”^“1”^“brain”)+“slc2a1”+ “ratgtg1”+(“facilitated”^“glucose”^“transporter”)+“gtg1”+(“solute”^“carrier”^“family”^“2”^ “member”^“1”)+“glut1”)
25	–	**19**	577	0	0	219	“tnf”
26	**1**	6	0	4	0	27	“fat protein drosophila”
27	**10**	1	0	0	0	6	“numb protein”
28	**9**	29	0	1	0	19	“epidermal growth factor”
29	**5**	0	0	1	0	4	“brahma protein”
30	–	**26**	0	0	0	7	“reaper”
31	–	–	–	**7**	0	12	“((“gonadotropin”^“releasing”^“hormone”^“receptor”)+“gnrhrs”+“gnrhr”)”
32	–	**523**	488	70	1	537	“fas”, “cd95”
33	**10**	0	63	0	0	10	“edg 1 protein”
34	–	–	–	**5**	0	27	(“her3”+(“v”^“erb”^“b2”^“erythroblastic”^“leukemia”^“viral”^“oncogene”^“homolog”^“3”)+ (“v”^“erb”^“b2”^“avian”^“erythroblastic”^“leukemia”^“viral”^“oncogene”^“homolog”^“3”)+ “erbb3”+(“transformation”^“gene”^“erbb”^“3”))
35	**93**	734	65	975	0	838	“interleukin 3”
36	–	**13**	4	9	0	580	“ing1”
37	–	–	–	**29**	0	134	(“ppargs”+“humpparg”+“pparg2”+(“peroxisome”^“proliferative”^“activated”^“receptor”^ “gamma”^“isoform”^“2”)+“pparg1”+“humppargs”+“pparg”+(“ppar”^“gamma”)+(“peroxisome”^ “proliferative”^“activated”^“receptor”^“gamma”^“isoform”^“1”)+(“peroxisome”^“proliferative”^ “activated”^“receptor”^“gamma”)+“nr1c3”)
38	–	–	–	**15**	0	403	((“mip”^“1”^“alpha”)+“scya3”+“ld78alpha”+“g0s191”+(“small”^“inducible”^“cytokine”^“a3”)+ “mip1a”+(“chemokine”^“ligand”^“3”)+“mip1alpha”+(“c”^“c”^“motif”)+“ccl3”+(“g0s19”^“1”))
39	–	**170**	0	4	0	103	“sp1”
40	–	**22**	0	8	0	56	“tie 2”
41	**1**	78	0	45	0	227	“cash protein”
42	–	**6**	257	3	0	40	“app”
43	**23**	5	0	5	0	78	“creb binding protein”
44	**23**	0	38	2	0	45	“fibroblast growth factor receptor 1”
45	**73**	23	0	20	0	156	“growth hormone”
46	**19**	0	7	0	0	9	“hepatocyte nuclear factor 3beta”
47	–	–	**3**	0	25	19	(((“purkinje”+“purkinjes”)^(“cell”+“cells”)^(“protein”+“proteins”)))
48	**49**	0	0	0	0	30	“stat6 proteinstat6 protein”
49	–	–	**1**	14	0	113	(((“tcr”+“tcrs”)))
50	**164**	1	11	3	0	169	“interleukin 6”
Total	25	16	3	6	0	0	

Table 4.7: Matches in the query tiers for the test topics.

Topic	Query Term/Phrase	Feedback Chem. Name	Ret.	R.&R.	MAP	R-P	MAP Fb.	R-P Fb.	Imp.
2	E2F transcription factor 1	transcription factor E2F	111	10	0.1515	0.0909	0.2559	0.1818	+69%
3	eukaryotic translation initiation factor 4E	Eukaryotic Initiation Factor-4E	36	12	0.5180	0.4615	0.6803	0.6154	+31%
4	guanine nucleotide binding protein (G protein), alpha activating activity polypeptide, olfactory type	G-Protein, Stimulatory Gs	42	0	0.0085	0.0000	0.0109	0.0000	+28%
9	signal transducer and activator of transcription 5A	mammary gland-specific nuclear factor	80	8	0.1244	0.1250	0.2508	0.2500	+102%
12	vitamin D (1,25- dihydroxyvitamin D3) receptor	Receptors, Calcitriol	134	24	0.1481	0.0800	0.2174	0.2000	+47%
13	aryl-hydrocarbon receptor	Receptors, Aryl Hydrocarbon	49	8	0.2650	0.2500	0.4342	0.3750	+64%
15	CD34 antigen	Antigens, CD34	97	3	0.3333	0.3333	0.7222	0.6667	+117%
16	heterogeneous nuclear ribonucleoprotein A1	hnRNP A1	7	3	0.3344	0.3333	0.6667	0.6667	+99%
20	surfactant associated protein A	Pulmonary Surfactant-Associated Protein A	19	6	0.4511	0.5000	0.3480	0.1667	+77%
21	adenylate cyclase activating polypeptide 1	pituitary adenylate cyclase activating polypeptide	61	7	0.1503	0.1429	0.1796	0.0000	+19%
22	arginine vasopressin	8-Hydroxy-2-(di-n-propylamino) tetralin	63	0	0.0255	0.0000	0.0200	0.0000	-22 %
24	Glut1	GLUT-1 protein	27	5	0.5821	0.7143	0.6596	0.7143	+13%
25	tumor necrosis factor superfamily, member 2	Tumor Necrosis Factor	575	25	0.0411	0.0769	0.0727	0.1154	+77%
30	reaper	reaper peptide, Drosophila	26	7	0.7760	0.6250	0.6955	0.5000	-10%
31	gonadotropin-releasing hormone receptor	Receptors, LHRH	23	4	0.7500	0.7500	0.7857	0.7500	+5%
32	CD95	Antigens, CD95	516	65	0.2353	0.2121	0.1964	0.1970	-17%
34	ERBB3	Receptor, erbB-3	31	5	0.2958	0.3333	0.4062	0.3333	+37%
36	p33ING1	p33(ING1) protein	13	4	0.3405	0.0000	0.4155	0.5000	+22%
37	peroxisome proliferative activated receptor, gamma	peroxisome proliferator-activated receptor	385	61	0.1281	0.1311	0.2106	0.1639	+64%
38	MIP1A	Macrophage Inflammatory Protein-1	83	9	0.0370	0.1111	0.1516	0.2222	+309%
39	Sp1 transcription factor	Transcription Factor, Sp1	168	35	0.3533	0.3421	0.2633	0.2105	-25%
40	TEK tyrosine kinase, endothelial	TIE-2 receptor tyrosine kinase	22	4	0.4946	0.6000	0.4413	0.4000	-11%
42	amyloid beta (A4) precursor protein	Amyloid beta-Protein Precursor	131	10	0.0414	0.0588	0.0749	0.0588	+81%
47	inositol 1,4,5-triphosphate receptor 1	inositol-1,4,5-triphosphate receptor	25	6	0.1535	0.1429	0.4149	0.4286	+170%
49	T-cell receptor alpha chain	Receptors, Antigen, T-Cell, alpha-beta	166	5	0.0833	0.1429	0.1115	0.1429	+34%

Table 4.8: Analysis of the effects of feedback on performance for the test topics.

another 16 topics having Tier 2 as their best tier. This confirms that the test data differs from the training data in that the query gene and protein names are not as similar to the relevant gene and protein names in the corpus.

Table 4.8 shows the chemical names associated by the Feedback subsystem with each topic for which no documents were retrieved in Tier 1, for the BTRF run on the test data. Whereas for the training data feedback was used for only 18 topics, for the test data feedback was used for 25 topics, or half of the 50 topics. This is due to fewer topics having an exact match, i.e. a match in Tier 1 in the Query Tiering subsystem. In 20 cases, feedback improved the performance, while the performance was degraded in 5 of the cases. As with the training data, it is apparent that there is a clear relationship between most of the query terms and the feedback term chosen by the feedback system. The ability to recognize this relationship using domain-specific knowledge would definitely improve retrieval.

The runs on the test data confirm that the combination of techniques and parameters chosen for the MultiText for Genomics system improves retrieval performance. The results showed that there are some differences between the characteristics of the training and test data, but our system was robust enough to have a very good performance on the test data.

There were a total of 49 official runs, submitted by 25 groups. The final results may be found in Hersh and Bhupatiraju [HB03]. Our system placed 4th among the 25 competing systems, with our two runs having mean average precision (MAP) scores of 0.3534 and 0.3479. Table 4.9 shows the top 15 official runs, sorted by MAP, along with the number of

Run Tag	Run Type	Mean Average Precision	Relevant @ 10 documents retrieved	Relevant @ 20 documents retrieved
NLMUMDSE	automatic	0.4165	3.16	4.84
NLMUMDSRB	manual	0.3994	3.20	4.56
nrc1	automatic	0.3941	2.94	4.38
biotext1	automatic	0.3912	3.06	4.46
nrc2	automatic	0.3771	2.76	4.36
biotext0	automatic	0.3753	2.92	4.30
uwmtg03btrf	automatic	0.3534	2.28	3.68
uwmtg03atrf	automatic	0.3479	2.48	4.00
axon2	automatic	0.3173	2.50	3.86
axon1	automatic	0.3118	2.40	3.78
CSUSM2	automatic	0.3079	2.68	3.76
edstanrecall	automatic	0.3015	2.60	3.74
edstanprec	automatic	0.2984	2.60	3.74
KUBIOIRNE	automatic	0.2980	2.32	3.42
KUBIOIRRAW	automatic	0.2937	2.24	3.38
Mean (all runs)		0.2313	1.85	2.85
Median (all runs)		0.1960	1.58	2.60

Table 4.9: The top 15 official runs by mean average precision.

relevant documents at 10 and 20 documents retrieved.

Chapter 5

Discussion

5.1 Analysis of Results

We have identified three features which appear to be vital to a successful biomedical document retrieval system, namely: 1) the ability to deal with variants of gene names; 2) recognition of the subject species of a document; and 3) use of metadata fields and structured data. Furthermore, we have also identified a fourth feature which, while not crucial, may have increased the performance of some systems for the TREC genomics track: 4) identification of documents which are cited by GeneRIFs.

We explain each of these features below.

5.1.1 Recognition of Gene Name Variants

A strategy for dealing with ambiguities in biomedical nomenclature seems to be the one defining feature separating a successful biomedical document retrieval system from a failure. While we have not examined every system that participated in the genomics track, it is clear that any system which did not implement this feature, or was unsuccessful in doing so, would fail to find the majority of relevant documents.

In our system, we used two different strategies for recognizing gene name variants. In the Okapi subsystem, gene names are relaxed and converted into term vectors, while in the Query Tiering subsystem, gene names are matched by relaxing the gene name and by using a boolean expression. These steps together served the same purpose as the hand-crafted gene variant generation rules and decision trees used by some other systems. Our approach may be described as a "shotgun" approach: we simply generated many re-arrangements of the given gene name in the belief that those which corresponded to sensible gene names would retrieve relevant documents. While unorthodox, this approach seemed to have paid off, rewarding us with a high precision in our retrieval system.

An issue related to the recognition of gene name variants is the disambiguation of acronyms. In our system, we do not attempt to disambiguate acronyms explicitly, leaving that function to the statistics of the corpus and our scoring functions, which weigh terms that co-occur frequently with the query terms more heavily. Acronyms in the MeSH controlled vocabulary may also be recognized by our Feedback subsystem. For example, on training topic 8, the query gene name is "luteinizing hormone/choriogonadotropin receptor", and our Feedback subsystem correctly deduced "Receptors, LH" as the most relevant chemical. This was accomplished without any recognition on the part of our system that the acronym "LH" stood for "luteinizing hormone".

5.1.2 Species Filtering

Because the same gene might exist in many different organisms, a retrieval system might retrieve many documents which are relevant to the gene but for the wrong species. Filtering out documents about species other than the topic species would therefore greatly increase the precision of the retrieval.

In our Query Tiering subsystem, documents in which the name of the species does not appear in the MeSH Heading metadata are removed from consideration. This does not completely eliminate documents which are not relevant to the species, since it is possible for the name of the species to appear in the MeSH Heading field even if the focus of the paper is another species. It is quite common for an article about a gene in one species to mention a homologue in a related species. Nevertheless, if the name of the wanted species does *not* appear in the MeSH Heading metadata, then the article is (almost certainly) not relevant. Thus, using species data in the MeSH metadata field may result in false positives but not (or rarely) in false negatives.

In our Okapi subsystem, the name of the topic species is added to the term vectors. However, we do not filter documents by species in this subsystem. Nevertheless, because the Okapi result set is fused with the Query Tiering result set which does not contain any documents where the topic species is not mentioned in the MeSH Heading metadata, the combined document set has effectively been filtered by species.

5.1.3 Use of Structured Data

Each MEDLINE record is divided into a number of metadata fields, and not every field is equally useful for determining the relevance of a document. A match between the query and the title, for example, appeared to be slightly more indicative of a document's relevance than a match between the query and its abstract, since the title is more tightly focused on the subject of the document. The top groups in the TREC genomics track were unanimous in according pride of place to the structured data and controlled vocabulary portions of the MEDLINE records, although each group used the data differently.

Our system makes use of the chemical list both in our Query Tiering subsystem and in our Feedback subsystem, as explained in Sections 3.2 and 3.4 above.

5.1.4 GeneRIF Identification

Not every document in MEDLINE is cited by a GeneRIF, and in fact the distribution of GeneRIFs is quite sparse. Because GeneRIFs are used as pseudo-relevance judgments for the TREC genomics track, the ability to determine which documents are cited by GeneRIFs confers a big advantage in the task of finding "relevant" documents.

At first, however, this might seem to be a case of overfitting the solution to the problem, since GeneRIFs were chosen to be the qrels for the genomics track merely for the sake of convenience. But there are, in fact, some quite legitimate reasons for wanting to distinguish between documents cited by GeneRIFs and those which are not. The goal of the task is to find all documents related to the function of a gene. Only a portion of MEDLINE documents are about gene function, of which a portion have been assigned GeneRIFs. The removal of documents which are not about gene function at all from the search pool would greatly reduce the effort needed for finding relevant documents. While there is no easy way to determine whether a document is about gene function, documents which have been assigned GeneRIFs is characteristic of this class of documents.

We did not make any attempt to classify documents which have been assigned GeneRIFs. However, GeneRIF identification was a component of several other systems which achieved high performance in the Genomics Track.

Chapter 6

Conclusions

6.1 Summary

To summarize, we adapted an "off the shelf" general purpose retrieval system to a genomics corpus. In doing so, we solved a number of problems which are essential for anyone wanting to construct a biomedical document retrieval system. We handled ambiguities in gene and protein names by generating term vectors containing relaxed versions, and also by matching them against a number of query tiers. We attempted to restrict our search to documents about the topics species by removing some documents in which the species is not mentioned in the MeSH Heading metadata field. We made use of structured data and controlled vocabulary by using the chemical list metadata for our query tiers and for pseudo-relevance feedback. All in all, we tuned our retrieval system to the specific features and characteristics of the MEDLINE corpus. Our system had an excellent performance in the TREC Genomics Track primary task, placing 4^{th} among 25 participating systems.

The research reported here is a preliminary study in the field of genomics information retrieval. Our experimental results demonstrate that it is possible to achieve very good retrieval performance, even without using expert knowledge, by tailoring standard IR techniques to the task and taking advantage of the corpus characteristics. Through our experimentations with the MultiText for Genomics system, we have determined some key features of a successful biomedical document retrieval system for the TREC Genomics Track, namely: a strategy for dealing with ambiguities in gene names, the ability to recognize the topic species of a particular document, and exploitation of metadata and other features of the corpus. We showed that a general purpose retrieval system can be successfully adapted to a biomedical corpus by incorporating each of these features.

As the TREC Genomics Track has generated a phenomenal amount of interest and appears poised to become a very active track in the future, we have provided potential future track participants with a recipe for constructing a good baseline system quickly. Future research into biomedical document retrieval can be built upon the foundations described in this thesis.

6.2 Future Work

There are a number of areas in which further work can be done. Due to time constraints, it was not possible to test every combination of techniques, or even a very wide range of parameters for each combination of techniques. Many experiments, such as fusion techniques other than the ones used in the final MultiText for Genomics system, were abandoned early on due to unsatisfactory preliminary results. Potentially, experimental parameters or combinations of techniques other than those we used might improve further retrieval.

The implementation of the Okapi retrieval model in our Okapi subsystem applies the retrieval model to entire documents, and does not distinguish between metadata fields. It would be an interesting experiment to add query tiering to the basic Okapi retrieval model by applying the retrieval model to each of the metadata fields separately. This can be implemented in our system by splitting the corpus into separate databases, with each database containing the data from one metadata field across all documents.

The metadata fields of the MEDLINE records contain information which we have shown to be highly relevant to retrieval. An avenue of exploration that is likely to be fruitful is to take advantage of the metadata more fully, in particular the hierarchical relationship inherent in the metadata. We have found that the most effective technique for finding the relevant documents in MEDLINE is to find a matching chemical name in the metadata. Currently, our system attempts to generate phrases and boolean expressions from the topic gene and protein names which are then checked against the contents of these metadata fields. This procedure may be improved in a number of ways. For example, the Feedback subsystem often retrieves a chemical name which actually corresponds to a class of chemicals. By recognizing that the topic gene or protein is a part of a broader family of genes or proteins, the search may be narrowed or broadened as necessary depending on the number of relevant documents retrieved. Furthermore, instead of using heuristics to guess at a chemical name or using feedback to find it, it might be possible to learn the chemical name from the corpus using pattern matching. A system with built-in genomics domain-specific knowledge can produce a list of candidate chemicals before any documents have been retrieved. Some intriguing possibilities for learning this domain knowledge include data mining the MEDLINE corpus and exploiting external databases, which we have not done in our system.

It was assumed due to preliminary tests that the gene name type (such as whether a gene name is its official name or an alias) made no difference to retrieval. However, further analysis is required to confirm or refute this assumption. It may be that the gene name type is relevant in a way that is not evident to the statistical techniques we have used in our experiments.

Another area requiring further inquiry is an assessment of the suitability of the GeneRIF data as the "gold standard" for relevance judgment. The use of GeneRIFs for this purpose is somewhat problematic, as the GeneRIFs are incomplete, in the sense that there were some documents which are related to a gene but which have not yet been assigned a GeneRIF. As a result, there are many *false negatives* (documents which are relevant but which are not judged to be relevant). Even though these false negatives should not affect the *relative performance* of different IR systems with respect to each other, according to the assumptions of the Cranfield paradigm, it would be instructive to pool the results from the various Genomics Track participants to obtain a list of the most relevant documents, in order to determine the extent to which the incompleteness of the GeneRIF data affected the evaluations of the performance of the various systems.

While our investigations were specific to the MEDLINE corpus and the requirements of the TREC Genomics Track, the lessons we have learned may have broader applications to other biomedical databases and other specialized forms of retrieval.

Bibliography

[BCB94] Bartell, Brian T., Cottrell, Garrison W., and Belew, Richard K. Automatic Combination of Multiple Ranked Retrieval Systems. In *Proceedings of the 17th Annual International ACM SIGIR Conference*. ACM, 1994.

[BCCC93] Belkin, N. J., Cool, C., Croft, W. B., and Callan, J. P. The effect of multiple query representations on information retrieval performance. In *Proceedings of the 16th Annual International ACM SIGIR Conference*. ACM, 1993.

[CA02] Chang, Jeffrey T. and Altman, Russ B. Promises of Text Processing: Natural Language Processing Meets AI. *Drug Discovery Today*, 7(19):992–993, October 2002.

[CC96] Clarke, Charles L. A. and Cormack, G. V. Interactive Substring Retrieval (MultiText Experiments for TREC-5). In *NIST Special Publication 500-238: The Fifth Text REtrieval Conference (TREC-5)*, pages 267–278, Gaithersburg, MD, 1996. National Institute of Standards and Technology.

[CC00] Clarke, Charles L. A. and Cormack, Gordon V. Shortest-Substring Retrieval and Ranking. *ACM Transactions on Information Systems*, 18(1):44–78, January 2000.

[CCB94] Clarke, Charles L. A., Cormack, G. V., and Burkowski, F. J. An Algebra for Structured Text Search and A Framework for its Implementation. Technical Report CS-94-30, Dept. of Computer Science, University of Waterloo, August 1994.

[CCB95] Clarke, Charles L. A., Cormack, G. V., and Burkowski, F. J. Shortest Substring Ranking (MultiText Experiments for TREC-4). In *NIST Special Publication 500-236: The Fourth Text REtrieval Conference (TREC-4)*, pages 295–304, Gaithersburg, MD, 1995. National Institute of Standards and Technology.

[CCK+02] Clarke, C. L. A., Cormack, G. V., Kemkes, G., Laszlo, M., Lynam, T. R., Terra, E. L., and Tilker, P. L. Statistical Selection of Exact Answers (MultiText Experiments for TREC 2002). In *NIST Special Publication 500-251: The Eleventh Text REtrieval Conference (TREC 2002)*, Gaithersburg, MD, 2002. National Institute of Standards and Technology.

[CCKL00] Clarke, C. L. A., Cormack, G. V., Kisman, D. I. E., and Lynam, T. R. Question Answering by Passage Selection (MultiText Experiments for TREC-9). In *NIST Special Publication 500-249: The Ninth Text REtrieval Conference (TREC-9)*, pages 673–654, Gaithersburg, MD, 2000. National Institute of Standards and Technology.

[CCKP99] Cormack, G. V., Clarke, C. L. A., Kisman, D. I. E., and Palmer, C. R. Fast Automatic Passage Ranking (MultiText Experiments for TREC-8). In *NIST Special Publication 500-246: The Eighth Text REtrieval Conference (TREC-8)*, pages 735–742, Gaithersburg, MD, 1999. National Institute of Standards and Technology.

[CCL+01] Clarke, C. L. A., Cormack, G. V., Lynam, T. R., Li, C. M., and McLearn, G. L. Web Reinforced Question Answering (MultiTest Experiments for TREC 2001). In *NIST Special Publication 500-250: The Tenth Text REtrieval Conference (TREC 2001)*, pages 673–679, Gaithersburg, MD, 2001. National Institute of Standards and Technology.

[CCPT00] Cormack, Gordon V., Clarke, Charles L. A., Palmer, Christopher R., and To, Samuel S. L. Passage-Based Refinement (MultiText Experiments for TREC-6). *Information Processing & Management*, 36(1):133–153, 2000.

[Cle67] Cleverdon, C. W. The Cranfield tests on index language devices. In *Aslib Proceedings*, volume 19, pages 173–193, 1967.

[Cle91] Cleverdon, C. W. The significance of the Cranfield tests on index languages. In *Proceedings of the 14th Annual International ACM SIGIR Conference*. ACM, 1991.

[CPVC98] Cormack, G. V., Palmer, C. R., Van Biesbrouck, M., and Clarke, C. L. A. Deriving Very Short Queries for High Precision and Recall (MultiText Experiments for TREC-7). In *NIST Special Publication 500-242: The Seventh Text REtrieval Conference (TREC-7)*, pages 121–132, Gaithersburg, MD, 1998. National Institute of Standards and Technology.

[CZP02] Castaño, José, Zhang, Jason, and Pustejovsky, James. Anaphora Resolution in Biomedical Literature. In *International Symposium on Reference Resolution*, Alicante, Spain, 2002.

[FO95] Faloutsos, Christos and Oard, Douglas W. A Survey of Information Retrieval and Filtering Methods. Technical Report CS-TR-3514, University of Maryland at College Park, 1995.

[FS93] Fox, Edward A. and Shaw, Joseph A. Combination of Multiple Evidence. In *The Second Text REtrieval Conference (TREC-2)*, pages 243–252, Gaithersburg, MD, 1993. National Institute of Standards and Technology.

[Gre00] Greengrass, Ed. Information Retrieval: A Survey, 2000.

[Har92] Harman, Donna. Overview of the First Text REtrieval Conference (TREC-1). In *NIST Special Publication 500-207: The First Text REtrieval Conference (TREC-1)*, Gaithersburg, MD, 1992. National Institute of Standards and Technology.

[HB03] Hersh, William and Bhupatiraju, Ravi Teja. TREC Genomics Track Overview. In *The Twelth Text Retrieval Conference (TREC 2003)*, Gaithersburg, MD, 2003. National Institute of Standards and Technology.

[Her03] Hersh, William. TREC 2003 Genomics Track – Protocol. Online: http://medir.ohsu.edu/˜genomics/, September 2003.

[KJ98] Kekäläinen, Jaana and Järvelin, Kalervo. The impact of query structure and query expansion on retrieval performance. In *Proceedings of the 21st Annual International ACM SIGIR Conference*. ACM, 1998.

[Lee97] Lee, Joon Ho. Analyses of Multiple Evidence Combination. In *Proceedings of the 20th Annual International ACM SIGIR Conference*. ACM, 1997.

[MSB98] Mitra, M., Singhal A., and Buckley, C. Improving Automatic Query Expansion. In *Proceedings of the 21st Annual International ACM SIGIR Conference*. ACM, 1998.

[NCB03a] NCBI. LocusLink. Online: http://www.ncbi.nlm.nih.gov/LocusLink/, September 2003.

[NCB03b] NCBI. PubMed Help / MEDLINE Display Format. Online: http://www.ncbi.nlm.nih.gov/, September 2003.

[NSA02a] Nenadić, Goran, Spasić, Irena, and Ananiadou, Sophia. Automatic Acronym Acquisition and Term Variation Management within Domain-Specific Texts. In

Proceedings of the 3rd International Conference on Language, Resources, and Evaluation (LREC-3), 2002.

[NSA02b] Nenadić, Goran, Spasić, Irena, and Ananiadou, Sophia. Automatic Discovery of Term Similarities Using Pattern Mining. In *Proceedings of CompuTerm 2002*, pages 43–49, Taipei, Taiwan, 2002.

[PCC+01] Pustejovsky, James, Castaño, José, Cochran, Brent, Kotecki, Maciej, Morrell, Michael, and Rumshisky, Anna. Linguistic Knowledge Extraction from Medline: Automatic Construction of an Acronym Database. *10th World Congress on Health and Medical Informatics (Medinfo 2001)*, 2001.

[PCS+02] Pustejovsky, James, Castaño, José, Saurí, R., Rumshisky, A., Zhang, Jason, and Luo, W. Medstract: Creating Large-scale Information Servers for biomedical libraries. In *ACL 2002 Workshop on Natural Language Processing in the Biomedical Domain*, Philadelphia, PA, 2002.

[Rob90] Robertson, S. E. On term selection for query expansion. *Journal of Documentation*, 46(4):359–364, 1990.

[RW94] Robertson, S. E. and Walker, S. Some Simple Effective Approximations to the 2-Poisson Model for Probabilistic Weighted Retrieval. In *Proceedings of the 17th Annual International ACM SIGIR Conference*. ACM, 1994.

[RWB98] Robertson, S. E., Walker, S., and Beaulieu, M. Okapi at TREC-7. In *The Seventh Text REtrieval Conference (TREC-7)*, Gaithersburg, MD, 1998. National Institute of Standards and Technology.

[RWJ+94] Robertson, S. E., Walker, S., Jones, S., Hancock-Beaulieu, M. M., and Gatford, M. Okapi at TREC-3. In *Overview of the Third Text REtrieval Conference (TREC-3)*, pages 109–126, Gaithersburg, MD, 1994. National Institute of Standards and Technology.

[SB88] Salton, G. and Buckley, C. Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, 24(5):513–523, 1988.

[SB90] Salton, G. and Buckley, C. Improving retrieval performance by relevance feedback. *Journal of the American Society for Information Science*, 41:288–297, 1990.

[TRE03] TREC Website. Text REtrieval Conference (TREC). Online: http://trec.nist.gov/, September 2003.

[TW02a] Tanabe, Lorraine and Wilbur, John W. Tagging Gene and Protein Names in Biomedical Text. *Bioinformatics*, 18(8):1124–1132, 2002.

[TW02b] Tanabe, Lorraine and Wilbur, John W. Tagging Gene and Protein Names in Full Text Articles. In *Proceedings of the Workshop on Natural Language Pro-*

cessing in the Biomedical Domain, pages 9–13. Association for Computational Linguistics, July 2002.

[Voo99] Voorhees, Ellen M. Natural Language Processing and Information Retrieval. In *SCIE*, pages 32–48, 1999.

[Voo02] Voorhees, Ellen. Overview of TREC 2002. In *The Eleventh Text Retrieval Conference (TREC 2002)*, Gaithersburg, MD, 2002. National Institute of Standards and Technology.

[XC96] Xu, Jinxi and Croft, W. Bruce. Query Expansion Using Local and Global Document Analysis. pages 4–11, August 1996.

[YCC+03] Yeung, David L., Clarke, Charles L. A., Cormack, Gordon V., Lynam, Thomas R., and Terra, Egidio L. Task-Specific Query Expansion (MultiText Experiments for TREC 2003). In *The Twelth Text Retrieval Conference (TREC 2003)*, Gaithersburg, MD, 2003. National Institute of Standards and Technology.

[YHF02] Yu, Hong, Hripcsak, George, and Friedman, Carol. Mapping Abbreviations to Full Forms in Biomedical Articles. *Journal of the American Medical Informatics Association*, 9(3):262–272, 2002.