# SEMANTIC CONCEPT EXTRACTION FROM ELECTRONIC MEDICAL RECORDS FOR ENHANCING INFORMATION RETRIEVAL PERFORMANCE

DAWID KASPEROWICZ

A THESIS SUBMITTED TO THE FACULTY OF GRADUATE STUDIES IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTERS OF ARTS

GRADUATE PROGRAM IN INFORMATION SYSTEMS AND TECHNOLOGY
YORK UNIVERSITY
TORONTO, ONTARIO

JUNE 2013

© DAWID KASPEROWICZ, 2013

# Abstract

With the healthcare industry increasingly using EMRs, there emerges an opportunity for knowledge discovery within the healthcare domain that was not possible with paper-based medical records. One such opportunity is to discover UMLS concepts from EMRs. However, with opportunities come challenges that need to be addressed. Medical verbiage is very different from common English verbiage and it is reasonable to assume extracting any information from medical text requires different protocols than what is currently used in common English text. This thesis proposes two new semantic matching models: Term-Based Matching and CUI-Based Matching. These two models use specialized biomedical text mining tools that extract medical concepts from EMRs. Extensive experiments to rank the extracted concepts are conducted on the University of Pittsburgh BLULab NLP Repository for the TREC 2011 Medical Records track dataset that consists of 101,711 EMRs that contain concepts in 34 predefined topics. This thesis compares the proposed semantic matching models against the traditional weighting equations and information retrieval tools used in the academic world today.

# Acknowledgments

I would like to express the deepest appreciation to my supervisor, Dr. Jimmy Huang, who has given me the opportunity to pursue my dream of receiving a Graduate Degree; for providing support in the form of donating his time, resources, and knowledge throughout the term of my Graduate studies. Without the support and encouragement received dating back to my Undergraduate studies by Dr. Jimmy Huang, this thesis would not been possible. In addition, I deeply thank Dr. Mariam Daoud for all the time and support she has given to making this thesis a success. I would further like to thank the School of Information Technology at York University, and Dr. Cristobal Sanchez-Rodriguez from the School of Administrative Studies at York University, for providing work opportunities that aided in my ability to financially support myself during the duration of my Graduate studies. I additionally like to thank all my loved ones who have given support, encouragement, and guidance throughout my life. I also extend my special thanks to Professor Marshall Walker, Professor George Georgopoulos, and Professor Zijiang Yang for being a part of my thesis examination committee. Last but not

least, I thank God for the countless blessings He has given me throughout my life and the ability to receive an education in a wonderful country like Canada, and in a reputable University as York University.

# Table of Contents

# Abbreviations

*Baseline* – A minimum value or starting point, used for comparing standard procedures or default methodologies to various non-standard or newly developed methodologies and procedures [1].

*BB2* – Stands for ***B**ernoulli-Einstein model with **B**ernoulli after-effect and normalization **2***, and it is a DRF document-weighting model.

*BM25* – It is a bag-of-words retrieval equation that ranks a set of documents based on the query terms appearing in each document [2].

*Conceptual Search* – An automated method used to search electronically stored unstructured text for information that is conceptually similar to the information provided in a search query [3].

*CUI* – It is the National Cancer Institute[1] ***Concept Unique Identifier***. This is a unique identifier that describes a specific disease or treatment. Each CUI can be used to identify official synonyms and abbreviations, definitions, among other useful information for specific diseases or treatments on their website: http://ncim.nci.nih.gov/ncimbrowser/.

*Data Mining* – It is the process of discovering useful patterns or knowledge from the data source [4].

*DFI0* – Stands for

*DFR* – Stands for ***Divergence from Randomness***. The paradigm is a generalization of one of the very first models of Information Retrieval [5].

*DirectIndex* – It is an index, in the usual or natural course or line, immediately upwards or downwards, that contains references, alphabetically arranged, to the contents of a series or collection of volumes; or an addition to a single volume or set of volumes containing such references of its contents [6].

*EMR* – Stands for ***Electronic Health Record / Electronic Medical Record***. They are an evolving concept defined as a comprehensive longitudinal collection of electronic health information about individual patients and populations, and it integrates

[1] http://www.cancer.gov/

healthcare information currently collected in both paper and electronic media for the purpose of improving the quality of healthcare [7] [8].

*HTML* – Stands for ***HyperText Markup Language***; the authoring language used to create documents on the World Wide Web [9].

*ICD* – Stands for ***International Classification of Diseases***; used to code and classify morbidity data from the inpatient and outpatient records, physician offices, and most National Center for Health Statistics (NCHS) surveys [10].

*Index* – In its simplest form, it is a data structure that attaches each distinctive term with a list of all documents that contains the terms. Thus, in retrieval, it takes constant time to find the documents that contain a query term/terms [4].

*InvertedIndex* – An index containing terms, as keys, mapped to references to the documents they appear in. The index is sorted by its keys. "Inverted" means that the documents are found by matching on terms, rather than the other way around [11].

*JSON* – Stands for **J**ava**S**cript **O**bject **N**otation and it is a lightweight data-interchange format and is based on a subset of JavaScript Programming Language, Standard ECMA-262 3rd Edition – a text format completely language independent but uses conventions familiar to programmers of the C-family languages [12].

*Lexicon* – The repository of idiosyncratic and unpredictable facts about lexical items organized as a list.

*Metadata* – Specific forms of data about a document, such as its authors name, title and date of publication. This metadata would generally include fields such as the date of creation and the format of the document, as well as the author and possibly the tile of the document. The possible values of a field should be thought of as finite – for instance, the set of all dates of authorship [13].

*MSH* – Stands for **M**edical **S**ubject **H**eadings and is a controlled vocabulary used for indexing articles, for cataloging books and other holdings, and for searching MSH-indexed databases, including MEDLINE [14].

*Natural Language Processing* – A range of computational techniques for analyzing and representing naturally occurring texts at one or more levels of linguistic analysis for the purpose of achieving human-like language processing for the purpose of achieving human-like language processing for a range of tasks or applications [15].

*PDF* – Stands for ***Portable Document Format***; a file format that captures formatting information from a variety of desktop publishing applications, making it possible

to send formatted documents and have them appear on the recipient's monitor or printer as they were intended [16].

*PL2* – Stands for ***Poisson model with Laplace after-effect and normalization 2.*** This DRF document-weighting model can be used for tasks that require early precision [17].

*Proximity Search* – A search for words or phrases found near one another, but not following one another immediately even after noise words are disregarded [18].

*SNOMED CT* – Stands for ***Systematized Nomenclature of Medicine Clinical Terms***; an extensive clinical terminology and it is the most comprehensive clinical vocabulary available in any language. It is concept-oriented and has an advanced structure that meets accepted criteria for a well-formed, machine-readable terminology [19].

*Stemming* – The process for reducing inflected (or sometimes derived) words to their root form [20].

*Stopwords* – Words are natural language words which have very little meaning, such as "and", "the", "a", "an", and similar words [21].

*Structured Document* – An electronic document where some method of embedded coding, such as markup, is used to give the whole, and parts, of the document various structural meanings according to a schema [22].

*Structured Document Retrieval Principle* – A system should always retrieve the most specific part of a document answering the query [13].

*Tokenizing* – Stems from the root word *token* – a string of characters, categorized according to the rules as a symbol – and the process of forming tokens from an input stream of characters is called *tokenization/tokenizing* [23].

*TREC* – Stands for ***Text Retrieval Conference***[2]. The conference started in 1992 for the purpose to support research within the information retrieval community by providing the infrastructure necessary for large-scale evaluation of text retrieval methodologies.

*UMLS* – Stands for ***Unified Medical Language System***; is a set of files and software that brings together many health and biomedical vocabularies and standards to enable interoperability between computer systems [24].

[2] http://trec.nist.gov/

*WARC* – Stands for ***Web ARChive***; is a file format that specifies a method for combining multiple digital resources into an aggregate archival file together with related documents [25].

*XML* – Stands for ***Extensible Markup Language***. This file format was designed for web documents and allows designers to create customized commands, enabling the definition, transmission, validation, and interpretation of data between applications and between organizations [26].

# Chapter 1

# Introduction

## 1.1. Motivation

Healthcare is a central concern in every individual's life as it is inevitable that every human acquires some form of illness during his or her lifespan. As a result, human societies have evolved to include various healthcare institutions such as hospitals, walk-in clinics, medical laboratories, and general practitioner's offices. The goal for developing these healthcare institutions is to aid humans in preventing illness from occurring and curing them of any illness they did catch. As part of the effort to prevent future illnesses and be better able to treat patients, healthcare institutions have created and kept records of the patients that they have admitted for treatment Until recently medical records are mostly paper-based and they contain incomplete information on patients. Healthcare institutions may not share patient information between one another, or even be aware that a patient may be using other healthcare institutions. Thus, healthcare institutions

probably do not have all treatment and procedure records of their patients if they are using other healthcare institutions.

Healthcare institutions are migrating away from paper-based medical records and moving towards electronic-based records as an effort to facilitate the sharing of medical records and reports between them using electronic healthcare systems. It is clear that EMR system adoption is rising [27] and paper-based record keeping is declining, not only in Canada, but also globally. Canada alone has experienced an increase in EMR usage of about 7% between 2004 and 2006, totaling approximately 23% nationally. Other developed nations such as Australia, Germany, Netherlands, New Zealand, United Kingdom, and the United States have EMR adoption rates of roughly 90%, 90%, 98%, 98%, 99% and 28% respectively [28].

Furthermore, not only is the format of medical records changing within the healthcare industry, but the landscape of the Canadian healthcare industry is also changing. In Canada, the demand of healthcare is increasing and the supply of healthcare professionals is decreasing. It is predicted in a report by an independent healthcare think tank in Ontario named *The Change Foundation* that by 2041, roughly 22% of Canadians will be 65 years of age or older. In 2001, this same demographic has been around 13% [29]. As people age they generally need increased amounts of medical attention, clearly indicating the demand for healthcare will increase as the Canadian population ages. The Canadian Institute for Health Infomration released reports stating that in 2004, 30% of

physicians working in Canada are 55 years of age or older and 61% of physicians were 45 years of age or older [30]. The institute also says the average age of healthcare professionals is increasing and the retirement age is decreasing [31]. In addition, Canada is expecting a decrease in new medical professionals replacing the ones that are retiring [31] [32]. The Canadian Institute for Health Information is an independent corporation whose goal is to improve the Canadian healthcare system.

Unique opportunities emerge in light of these ongoing events in Canada. EMR systems have the potential to offer many benefits to healthcare institutions. Some of these benefits include: interoperability, quality of care, healthcare professionals efficiency and time management improvements, patient safety, patient privacy and confidentiality, patient-doctor relationship improvements, and decrease in the cost of healthcare [33]. In spite of these benefits, it is reasonable to speculate that a significant amount of experience and knowledge will be lost as healthcare professionals become older and their retirement age decreases, while having less medical professionals enter the industry than there is leaving it. Medical professionals can only gain this experience with years of experience and training in the industry. Additionally, it is reasonable to assume EMR systems will not counterbalance the loss of experienced medical professionals and the knowledge they have, even with the benefits EMR systems provide. Fortunately, it is possible to acquire knowledge contained in EMR's and use it to aid current healthcare professionals in the diagnosis and treatment of their patients.

Being able to gain knowledge from EMR's from past experienced healthcare professionals helps improve healthcare quality, the speed of patient visits, and cost of care, even with the decrease in medical professionals. To achieve the goals outlined above, it is necessary to first be able to identify medical concepts contained within any EMR to further leverage the records for further research. This thesis presents methods aim to help in performing accurate medical concept retrieval from EMRs and match the medical concepts with concepts in predefined topics. The goal is to identify the most relevant EMRs for any given topic and rank them in descending order.

## 1.2. Contributions

This thesis presents new methods that aim to extract medical concepts from EMRs. Being able to do so helps facilitate further research in leveraging EMRs to aid current healthcare professionals in diagnosing and treating their patients based on experiences and knowledge of other medical professionals. Presented below is the methods utilized in this thesis a description of each.

1. *Term-Based Matching* – Determines an EMR document relevancy by calculating a conceptual score, computed by matching terms in the EMR with those that overlap with a given topic.

2. *CUI-Based Matching* – Determines an EMR document relevancy by calculating a conceptual score, computed by matching CUI's in the EMR with those that overlap with a given topic.

Currently, there has never been a publically available dataset of EMRs. This is the first instance such a dataset has been made available to the public. TREC released this dataset of highly de-identified EMRs in 2011. There are instances where similar datasets are used in the private sector; however, those datasets are exclusive to select corporations for confidential research purposes. The research presented in this thesis is the first of its kind. The research aims to propose semantic matching models on EMR data with the dataset released by TREC.

## 1.3. Thesis Outline

The thesis is organized into six main chapters, and includes appendices along with a bibliography. The six chapters include:

- Chapter 1: Introduction
- Chapter 2: Background and Related Work
- Chapter 3: Information Retrieval Challenges and Proposed Solutions
- Chapter 4: Experimental Setting and Implementation
- Chapter 5: Results and Evaluation
- Chapter 6: Conclusions and Future Work

# Chapter 2

# Background and Related Work

Information retrieval is a field of study that helps with locating material of an unstructured nature. This field of study aims to satisfy an information need from large collections [13]. The ideology of computerized information retrieval has been around since 1945, when Vannevar Bush first popularized it in his article titled: *As We May Think* [34]. A manual form of information retrieval has existed prior to 1945, which was primary performed by reference librarians, paralegals, and other similar professional searchers. Subsequently, the first several information retrieval systems and techniques emerged in 1970 [13] and exploration has continued up to this very day. This exploration eventually formed the research conference known as TREC. The research conducted within this conference has significantly enhanced the methodologies in the field of information retrieval by creating document-scoring functions that ascribe scores to a

subset of unstructured documents retrieved by a searching system. These methodologies were developed early in the conferences history, and they are still being used today.

One contribution developed from TREC is the BM25 ranking equation. Arguably, this equation is one of the most important advancements in the field of information retrieval because years after its conception it is still being used [35] [36]. There are numerous information retrieval systems in existence that focus on the use of BM25. One of the most prestigious of information retrieval systems based on the BM25 ranking equation is OKAPI. OKAPI was developed in in the 1970s – 1980s by Stephen E. Robertson, Karen Sparck Jones, et.al at London's City University, and is widely used as a standard weighting equation in academia.

In relation to EMRs, the work of Rector from 1991 is one of the first to offer fundamental principles that underline the model of an EMR [37]. They establish premises such as an EMR data consists of healthcare professionals' presumptions based on their observations rather than the irrefutable condition of a patient. This allows other healthcare professionals to disagree with any data contained within an EMR. Furthermore, they present the premise that the records should contain descriptive information of what should be said or done, and not of what is correct to say or do.

It is important to understand the various types of ways EMRs have their information recorded before one is able to begin the process of retrieving information from them. There are two main methods of information entry for an EMR that was

established from the works of Hersh in 2009. The first method is using structured data entry systems, while the second method is using unstructured entry systems [38]. Hersh described the use of structured data entry systems allowing healthcare professionals to enter their data into pre-determined form fields. These fields can such contain data such as age, gender, name, or patient diagnosis. One may leverage the metadata contained in the fields later for further record examination. In contrast, unstructured data entry lacks any metadata information; however, it provides healthcare professionals the ability to enter free text. Allowing healthcare professionals to enter free text gives them the opportunity and flexibility to express themselves in reports that they otherwise would not have been able to do with structured data entry.

It is debatable which data entry method is superior because both formats provide their own advantages. A recent white paper written by Feldman in 2013 states that with the advancements in the techniques and technologies used in conceptual search and natural language processing applications, the ability to retrieve in-depth information, understanding, accuracy, and transparency becomes possible [39]. He states that traditional healthcare analysis focuses on structured data and does not utilize unstructured data. He claims this is due to the techniques and technologies being underdeveloped to perform such tasks. The paper mentions that health organizations will continue to struggle to improve predictive analytics to save lives, cutting costs, improving operational and clinical outcomes, preventing readmission, and dealing with the financial

implications unless they utilize the information contained in unstructured data. Furthermore, research has indicated that restricting healthcare professionals to enter data in a structured fashion causes a loss of freedom of expression [38], which undoubtedly will lead to important information being omitted in EMRs.

Standardization is a central aspect to nearly all fields of study and practice, and when analyzing EMRs this is no exception. The works of Waegemann from 2002 illustrates that there is a correlation between the quality of information retrieved form EMRs and the data capturing process to generate an EMR [40]. He suggests that standardizing all aspects of an EMR will facilitate an increase in data entry quality, and therefore directly improve the information retrieved from EMRs. Various types of medical controlled vocabularies have been developed in an effort to bring standardization to EMRs. Some of the medical controlled vocabularies are ICD-9, ICD-9-CM, ICD-10, ICD-10-CM, SNOMED CT, and UMLS. These vocabularies were developed for unique uses of categorizations in healthcare practice.

Currently in Canada, there is much work being performed in relation to EMRs and EMR systems. In September 2000, the Government of Canada invested $500,000,000.00 in an independent non-for-profit corporation named Canada Health Infoway to hasten the advancement of information systems technologies for healthcare [41]. In 2003, the Government of Canada invested an additional $600,000,000.00 to Canada Health Infoway followed by a $100,000,000.00 investment in 2004. These

investments add up to a total of $1,200,000,000.00 to improve the healthcare systems in Canada [41]. The Government of Canada is expecting to see significant improvements of healthcare quality in Canada that is similar to other first world nations who adopted EMR systems. For example, the United States of America is experiencing improvements their healthcare quality because EMRs enabled them to discover unexpected possibilities for healthcare developments [41].

## 2.1. Uniqueness of the Healthcare Domain

As mentioned in Section 1.1, there is a global movement in adopting EMR systems. As a result, some consequences arise with this adaptation. Healthcare systems will depend on electronic information and will have a greater need for information retrieval systems. There is also a correlation between the contributions made in the information retrieval field and the nature of datasets. Most information retrieval systems available today are primary used in research opposed to clinical uses [42]. In relation to biomedical datasets, there is an increased probability that extracted information from EMRs will be erroneous because of the lack of specialized information retrieval systems designed for medical documentation. What information is or isn't considered relevant is also highly dependent on the ones information need [43]. This forms a need develop specialized information retrieval systems to meet the perceptions of those seeking specific information from healthcare data. As an exemplification of this necessity, the genomics field has developed their own information retrieval tools that are used

conjunction with datasets geared towards their field. One tool they specifically developed was the *Basic Local Alignment Search Tool* that is designed to work with genomic datasets [44]. There is undoubtedly a need for unique information retrieval systems for the healthcare field as healthcare professionals use their own specialized terminology and symbolism. This notion is further supported by [45] where it states there is a need for determining relevant documents and matching them according to inference and meaning. Only with a specialized information retrieval system can a high precision be achieved with healthcare datasets.

## 2.2. Text REtrieval Conference – Measuring Performance

TREC is co-sponsored by the national Institute of Standards and Technology (NIST) and the U.S. Department of Defense. It was first started in 1992 as part of the TIPSTER Text Program. Its main objectives was to encourage research in information retrieval based on large test collections; increase communication among industry, academia, and government by creating an open forum for the exchange of research ideas; speed the transfer of technology from research labs into commercial products by demonstrating substantial improvements in retrieval methodologies on real-world problems; and increase the availability of appropriate evaluation techniques to be used by the industry and academia, including the development of new evaluation techniques more applicable to current systems. The conference provides the evaluation criteria to appraise any information retrieval. The work presented in this thesis is based on the effort

conducted for the Medical Records Track. The main goal of the Medical Records Track is to foster research by providing content-based access to free-text fields of electronic medical records. The golden standard is provided by TREC and it is the primary method used in determining the effectiveness of the approaches presented in this thesis.

## 2.3. Healthcare Research Today

Today, there are numerous information retrieval systems available to healthcare professionals. Using these systems is increasingly being encouraged in attempts to enhance patient care quality by providing better use of evidence-based medicine. The Canadian healthcare professional has a plethora of EMR vendors to choose from that may have the potential of providing or utilizing information retrieval functionality. Such vendors include Wolf Medical Systems[3], P & P Data Systems Inc.[4], Microquest Inc.[5], YMS[6], MD Physician Services[7], ABELMed Inc.[8], Applied Informatics for Health Society[9], Nightingale Informatix Corporation[10], Optimed Software Corporation[11],

[3] www.telushealth.com
[4] http://www.p-pdata.com/
[5] http://www.microquest.ca/
[6] http://www.ymsmd.com/
[7] http://www.md.cma.ca/ps-suite
[8] http://www.abelmed.com/
[9] http://www.aihs.ca/
[10] http://www.nightingalemd.ca/
[11] http://www.optimedsoftware.com/

Intrahealth Canada Ltd.[12], AlphaGlobal iT Inc.[13], Oscar Host[14], and Jonoke Software Development Inc[15].

### *2.3.1 Information Retrieval System Varieties*

There is at least one variety of a generic information system at the base of nearly every information retrieval system. These varieties include: Boolean information retrieval systems, ranking information retrieval systems, multimedia information retrieval systems, and distributed information retrieval systems [42].

In Boolean information retrieval systems, the dataset acts as mathematical expressions. To locate relevant materials within the dataset, it is necessary to generate queries consisting of key terms alongside operators. These operators include *and*, *or*, and *not*. Once this query has completed generating and is entered into the system, the system then attempts to locate the relevant materials by making substitutions that satisfy the query and the materials contained within the dataset [42].

In ranking information retrieval systems, the dataset is treated as objects defined by the values of properties related to the contained words within the dataset. A cumulative measure is leveraged to evaluate the similarity between various materials within the dataset [42]. As an exemplification, consider three documents; the first

[12] http://www.intrahealthcanada.com/
[13] http://www.alpha-it.com/
[14] http://oscarhost.ca/
[15] http://www.jonoke.com/

containing the word 'bronchitis' four times, the second containing the word 'bronchitis' five times, and the third containing no trace of the word 'bronchitis'. The first document would be considered more similar to the second document, opposed to the third document. The reason is because the cumulative appearances of the word 'bronchitis' are more similar between the first and second document, opposed to the third.

Multimedia information retrieval systems concern themselves with audio, video, and still imagery information. They use two main methods to analyze these materials: tagging method and content method. The tagging method is the process of correlating text to the multimedia being examined to allow existing information retrieval systems to leverage the text to identify and return the said multimedia as part of the results to a query. The content method involves matching the contents of the multimedia in terms of associations between the shapes, volumes, colours, and textures that establish the multimedia [42].

### *1.1.1 Current Healthcare Research in the Industry*

There is significant research being performed in the industry of healthcare that would increase the healthcare quality provided by medical professionals. Arguably, one of the uppermost profiled research is being conducted by International Business Machines Corp. (IBM) and their partnership with WellPoint Incorporated. WellPoint is a health benefit company in terms of medical membership in the United States of America

who have independent licensee with the Blue Cross and Blue Shield Association. Together, the two companies are taking IBM's Jeopardy champion computing system Watson and turning it into an information retrieval and knowledge discovery system for healthcare professionals. They hope it will be able to answer medical professionals' questions through the analysis of vast amounts of medical data [46].

In addition, Flybridge Capital Partners, Highland Capital Partners and Google Ventures invested $6,000,000.00 to collaborate with Predilytics [47] to deliver predictive models for clinical improvements. Predilytics is an information technology company that provides healthcare solutions leveraging machine-learning technology. These predictive models leveraging machine-learning technology used by Predilytics are able to examine biomedical text to identify medical concepts and discover relationships between these concepts. They are also able to detect new hidden patterns and new insights without human intervention, and develop new algorithms or modify existing models based on the discovered patterns and insights.

# Chapter 3

# Information Retrieval Challenges and Proposed Solutions

## 3.1. Challenges in Information Retrieval

The information retrieval field has made significant advancements since its conception; however, challenges continue to exist that need addressing as the demands and needs of the field steadily increase. The healthcare industry has unique challenges in relation to the capturing, managing, analyzing, and mining of healthcare data.

The dataset used for the research contained in this thesis contained XML EMRs. There are challenges in regards to retrieving information from XML documents. XML documents are structured in nature. With structured information retrieval, the user is interested in a specific XML elements and not the document in its entirety. For this

reason, it is important to keep in mind the structured document retrieval principle—a system should always retrieve the most specific part of a document answering a query. In addition, determining the XML element to index is of utmost importance, as indexing the entire document has a likely probability of increasing irrelevant documents being retrieved to a given search query. Indexing smaller passages in documents generally would result an increase of truly relevant documents because of the probability that a query will match a small passage of a document is higher opposed to matching an entire document [13].

The next challenge is with unstructured data in the EMRs. Unstructured data entries allow individuals to express themselves with natural human written language. In this context, natural human language written by healthcare professionals is complex and has various syntaxes. Due to this reason, it is necessary to perform Natural Language Processing on unstructured data fields in EMRs. Natural Language Processing is able to identify patterns contained within natural human language in EMRs, and teach computerized systems to recognize said patterns to extract meaningful elements such as names of people, places, drugs, diseases, symptoms, and the relationships between each of them [39]. In addition, the ability to perform stemming and treat words with the same stem as synonyms is commonplace for natural language processing today. It is able to perform these functions with the use of what is known as the lexicon.

A lexicon comes with its own set of challenges and the way one addresses these challenges affects the lexicons quality. One of the first challenges with lexicons is in

relation to knowledge acquisition. To create a well-formed lexicon it is necessary to ensure that one leverages various textual sources, interaction with human beings, and other sources so that one can extract lexical, grammatical, semantic, and pragmatic knowledge from them. The issue is how one can locate such sources, and what techniques ought to be used to process the data gained from them. Current knowledge acquisition techniques lack the speed, and are overly complex to use on a comprehensive scale or on large sources [48]. Interaction with multiple underlying systems that enable natural language processing systems flexibility is another challenge that needs addressing. There is a limitation on how one can leverage lexicons, and the forms of language they are able to communicate with is also limited [48]. A third challenge with a lexicon is with it gaining only partial knowledge from its sources because natural languages leverage multiple-sentences to convey ideologies. In addition, the language used in the unstructured fields of EMRs may contain fragmented language that will further limit the lexicons ability to use any of the knowledge it received from its sources fully. A lexicon would need sources with perfect input to produce perfect output, and they would also need perfect examining materials to give perfect results. Such sources are extremely unlikely to exist as the use of unusual, incomplete, and errorful language is more common [48].

In efforts to circumvent the challenges with developing a lexicon it is important to select high quality and standardized sources to develop high quality ontologies that are

used for examining and retrieving information accurately from the dataset. Fortunately, the medical field has such sources. Among the best known of these sources is the UMLS. Conversely, there over forty UMLS sources that are available and each contains their own unique vocabulary and classifications of medical language. Identifying which of these sources, or combination of sources, that would provide the ideal ontological representations of medical language in a lexicon is of vital importance. Using only one UMLS source to create ontologies may run the risk of having incomplete ontologies, and therefore increasing the risk of identifying medical concepts incorrectly or overlooking what otherwise would be considered a medical concept. Similarly, the same holds true with using too many UMLS sources because ontologies may become too broad and incorrectly identify medical concepts that should have never been a concept to begin with.

## 3.2. Solutions in Ranking Relevant Documents

Since its formation, the information retrieval field has developed many ranking functions whose aim is to accurately and reliably be able to rank a given document's relevance relating to any given topic. Each of the developed ranking functions share a common characteristic of assigning a weight based relevance score to documents. Performing normalization is important for the ability to produce effective and accurate relevance scores [49]. The majority of the widely used and relevance functions incorporate normalization as seen in the popular ranking model BM25. Attributable to

the wide popularity and success of BM25, this equation forms the baseline for the research conducted in this thesis. More details of the BM25 equation is in Section 3.3, and further information on how BM25 forms the baseline is in section 5.1.1.

There is also a wide range of free proprietary tools that help with ranking documents in various research fields. Some of these tools are specifically designed to function within specific domains, while others have been designed to be general all-purpose tools. One of the most popular freely available tools used by academic researchers is Terrier. Terrier aids in the ranking of documents by providing a wide array of ranking functions and has an automatic indexing process. This thesis leverages Terrier for its popularity. Information on the tool is found in Section 4.2.

## 3.3. BM25

As mentioned in Chapter 2, BM25 is arguably considered to be one of the most important advancements in the field of information retrieval. The function has proven itself by being used with a wide variety of content when ranking data was necessary, and it has always performed well in each of the fields of research. With its high tuneability and being well defined, BM25 is the go-to ranking function for most researchers. BM25 takes an ad-hoc approach to ranking documents, where the formula are tried because they seem to be plausible.

Chapters 3.3.1 to 3.3.6 introduce the evolution of the BM25 weighting equation. By illustrating the evolution of the equation, it is possible to have a clear understanding of how BM25 started from a basic weighting model, and evolved into sophisticated ranking model by adding various improvements such as the 2-Poisson model, term frequency improvements, document length improvements, and query term frequency improvements.

| | Relevant | Non-Relevant |
|---|---|---|
| Term "Hurricane" occurs | $P(\underline{x}\|R)$ | $P(\underline{x}\|\overline{R})$ |
| **Term "Hurricane" does not occur** | $P(\underline{0}\|R)$ | $P(\underline{0}\|\overline{R})$ |

**Table 1: Contingency Table to Calculate a Document's Relevance**

### *3.3.1 Basic Weighting Model*

From a statistics point of view, the basic weighting equation that BM25 is derived from is expressed in the following equation:

$$w(\underline{x}) = \log \frac{P(\underline{x}|R)P(\underline{0}|\overline{R})}{P(\underline{x}|\overline{R})P(\underline{0}|R)}$$

***Equation 1: Initial Weighting Model***

where $\underline{x}$ is a vector of information about the document, $\underline{0}$ is a reference vector representing a zero-weighted document, and where $R$ and $\overline{R}$ are representative of relevance and non-relevance respectively. For example, each component of $\underline{x}$ may

represent the presence or absence of a query term in the document or its document frequency, and $\underline{0}$ could be the "natural" zero vector representing all query terms absent. An exemplification of this is seen in Table 1 for a single term query, "hurricane". Single term queries can be understood as the simplest queries possible as they only have one term. One calculates a document's relevance using Equation 1.

If we assume the terms are independent from each other, even for the queries that contain multiple terms, Equation 1 can be used to calculate the relevance of a document to a specific query by decomposing $w$ into individual term weights. The equation can then be transformed to the following:

$$w = \log\frac{p(1-q)}{q(1-p)}$$

***Equation 2: Transformed Equation Based on Individual Terms***

where $p = p(term\ present|R)$ and $q = p(term\ present|\overline{R})$. With an appropriate estimation method, the equation can be transformed to become:

$$w^{(1)} = \log\frac{(r+0.5)/(R-r+0.5)}{(n-r+0.5)/(N-n-R+r+0.5)}$$

***Equation 3: BM1 as Used by S.E. Robertson in TREC-1***

where $N$ is the number of indexed documents, $n$ is the number of documents in $N$ containing the sought out term, $R$ is the number of known relevant documents, and $r$ is the number of documents in $R$ containing the sought for term. The value *0.5* is used to smooth out the results. If we do not smooth the results, the $w^{(1)}$ will come from Equation 2 by replacing $p$ with $r/R$, and $q$ with $\frac{n-r}{N-R}$ respectively. $w^{(1)}$ from Equation 3 is also known as BM1 and it is used by S.E. Robertson in TREC-1.

### *3.3.2 2-Position Model*

From Chapter 3.3.1, $w^{(1)}$ has the ability to model the presence and absence of terms; however, it cannot model the within-document term frequency. If one deals with within-document term frequency rather than the presence and absence of terms, then the equation is as follows:

$$w = \log \frac{p_{tf} q_0}{q_{tf} p_0}$$

***Equation 4: Within-Document Term Frequency Weighting Equation***

where $p_{tf} = p(tern\ oresent\ with\ frequency\ tf|R)$, $q$ is the corresponding probability for $\overline{R}$, and $p_0$ and $q_0$ are those for term absence.

Some work has been done in creating a technique to model within-document term frequencies by means of the mixture of traditional Poisson distributions. Hater originally

began work on 2-Position distribution [51]. Before discussing the 2-Poisson model, it is worth explaining the ideas that are necessary for the model to function.

One assumes that the occurrences of a term in a document have a random nature that reflects a real, but hidden distinction between documents that are about the concept represented by the term and those that are not. The documents that are about a given concept are described as being *elite* for that particular term. One may draw an inference about a given concept being elite from the term frequency, but this inference will actually be probabilistic. Additionally, relevance is related to a term being elite rather than to term frequency, which is assumed to be dependent only on a term being elite. The term-independence assumption is replaced by the assumption that the elite properties of different terms are independent of each other. It is useful to introduce this hidden elite variable in order to gain an understanding of the relationship between multiple term occurrences and their corresponding relevance.

The 2-Position model is a specific distributional assumption based on the elite variable hypothesis discussed above. The assumption is that the distribution of within-document frequencies is Poisson for the elite documents, and also for the non-elite documents but with different means. The 2-Position model assumes that a document length is constant.

For the 2-Position model, there are usually some estimation problems because the general estimation method for the Position parameters is not well defined, and because

the model is too complex by requiring a large number of different parameters for establishing an estimation. Successive work on mixed-Position models have been suggested. They provide alternative estimation methods that may be preferable what was exists [52]. Combining the 2-Position model with Equation 1, one can obtain the following weight equation for a term *t*:

$$w = \log \frac{(p' \lambda^{tf} e^{-\lambda} + (1-p')\mu^{tf} e^{-\mu})(q' e^{-\lambda} + (1-q')e^{-\mu})}{(q' \lambda^{tf} e^{-\lambda} + (1-q')\mu^{tf} e^{-\mu})(p' e^{-\lambda} + (1-p')e^{-\mu})}$$

***Equation 5: Combination of the 2-Position Model with the Initial Weighting Model***

where $\lambda$ and $\mu$ are the Position means for *tf* in the elite and non-elite sets for *t* respectively, $p' = p(document\ elite\ for\ t|R)$, and $q'$ is the corresponding probability for $\overline{R}$.

The estimation problem is apparent in Equation 5 in that there are four parameters for each term, which none is likely to have direct evidence because of the elite variable being a hidden variable. It is because of this problem that makes Equation 5 inflexible, which leads one having to go through the rough model approach.

### *3.3.3 Term Frequency Improvement*

In order to allow within-document term frequency *tf* to influence the weight, different functions are utilized. Once of the functions is effective and based on the

rational that even if we do not use the full Equation 5, one may use it to suggest the shape of an appropriate equation. Looking at Equation 5, one can see the following characteristics:

- it is zero for *tf = 0;*
- it increases monotonically with *tf*
- but to an asymptotic maximum;
- that approximates to the Robertson weight that would be given to a direct indicator of being elite

After rearranging Equation 5, we get the following formula:

$$w = \log \frac{(p' + (1-p')(\mu/\lambda)^{tf} e^{\lambda-\mu})(q' e^{\mu-\lambda} + (1-q'))}{(q' + (1-q')(\mu/\lambda)^{tf} e^{\lambda-\mu})(p' e^{\mu-\lambda} + (1-p'))}$$

***Equation 6: Rearranged Equation of the Combination of the 2-Position Equation with the Initial Weighting Model***

From Equation 6, $\mu$ is smaller than $\lambda$. As $tf \rightarrow \infty$, $(\mu/\lambda)^{tf}$ goes to zero, and $e^{\mu-\lambda}$ is small and as such, can be estimated as:

$$w = \log \frac{p'(1-q')}{q'(1-p')}$$

***Equation 7: The Estimation of the Rearranged Equation of the Combination of the 2-Position Equation with the Initial Weighting Model***

It is necessary to construct an equation that is *tf*-related and satisfies the characteristics outlined for Equation 5. Such an equation can be constructed by the following principles: The function $tf/(constant + tf)$ increases from 0 to an asymptotic maximum in approximately the right fashion. The constant determines the rate that the increase drops off. With a large constant, the function is linear for small *tf*, whereas with a small constant the effect of increasing *tf* rapidly decreases. This function has an asymptotic maximum so it needs to be multiplied by an appropriate weight similar to that of Equation 7. Since one cannot estimate Equation 7 directly, the alternative is using the ordinary Robertson weight $w^{(1)}$, based on the presence and absence of a term. With this, one obtains the following:

$$w = \frac{tf}{(k_1 + tf)} w^{(1)}$$

***Equation 8: Weight of a Term Based on the Presence and Absence of a Term using the S.E. Robertson Equation from TREC-1***

where $k_1$ is an unknown constant. The model does not convey anything about the kind of value that is to be expected for $k_1$. S.E. Robertson determines the value for $k_1$ by experiments with TREC datasets. They found values around 1.0 – 2.0 are correct values for TREC data. Additionally, they pointed out that the shape of Equation 8 is different from Equation 6 in one important way; Equation 6 is convex towards the upper left, whereas Equation 8 can be S-shaped with some combinations of parameters, which

increases slowly in the beginning, then rapidly increased in the center, and finally slowly again.

### *3.3.4 Document Length Improvement*

After the document term frequency is integrated into the weighting function, the document length becomes the next issue that needs addressing.

In real situations, a document can be short or long. Both these short and long documents may also have the same subject. At a minimum, there are three reasons why documents lengths vary in length. The first reason is that some documents may cover more material than other documents. For example, a long document may consist of a wide variety of unrelated short documents concatenated together. This is known as the scope hypothesis. The second reason why document length varies is that a long document may be similar to a short document in terms of the message it conveys; however, because the two documents have different authors, the individuals writing style of the authors makes one longer than the other. For example, one document covers a similar scope to a short document, but it uses more words to convey the same content. This is known as the verbosity hypothesis. The third reason is that real document collections have a combination of the aforementioned two reasons.

There is little progress in relation to the scope hypothesis, and the work on document length discussed here assumes the verbosity hypothesis. The verbosity

hypothesis implies that a document's properties, such as relevance and being elite, can be regarded as being independent from the document length. Being elite is given for a term, and the number of occurrences of a given term depends on the document length. From this perspective, one can incorporate this hypothesis by normalizing *tf* for a document length *d*. Assuming the value of $k_1$ is appropriate to documents that have an average length Δ, the weight of a term is then expressed as:

$$w = \frac{tf}{\left(\frac{k_1 * d}{\Delta} + tf\right)} w^{(1)}$$

***Equation 9: The Weight of a Term Based on Average Document Length***

### *3.3.5 Query Term Frequency Improvement*

There is natural symmetry between documents and queries, and this suggests that one could treat within-query term frequency *qtf* in a similar fashion to within-document term frequency. This suggests that by analogy with Equation 8, a weighting function for query terms can be as follows:

$$w = \frac{qtf}{k_3 + qtf} w^{(1)}$$

***Equation 10: Weight of a Term Based on the Presence and Absence of a Term using the S.E. Robertson Equation from TREC-1 for Query Term Frequencies***

where $k_3$ is an unknown constant. Experiments suggest a large value of $k_3$ is effective, making the following equation to be equivalent to Equation 10:

$$w = qtf * w^{(1)}$$

***Equation 11: Weight of a Term Based on the Presence and Absence of a Term using the S.E. Robertson Equation from TREC-1 for Query Term Frequencies with Large $k_3$ Value***

Equation 11 can be thought as the normalization of query term frequencies. The basic assumption is that the frequency of query terms should have a direct effect on the weighting function. The more frequently a term appears in a query, the more important that term should be. It matches with the human intuition for natural language whereby they emphasize points by repeating key terms.

### *3.3.6 Final BM25 Formula*

Once the above individual improvements are complete, they are integrated together to create the final weighting function. When there is no relevance information, $w^{(1)}$ approximates to the following equation:

$$w^{(1)} = \log\frac{N - n + 0.5}{n + 0.5}$$

***Equation 12: BM1 as Used by S.E. Robertson in TREC-1 Revised for No Relevance Information***

Based on this equation, two weighting equations become available for one to use:

$$w = \frac{tf}{k_1 + tf} * \log\frac{N - n + 0.5}{n + 0.5} * \frac{qtf}{k_3 + qtf}$$

***Equation 13: BM15 Weighting Equation***

$$w = \frac{tf}{\frac{k_1 * d}{\Delta} + tf} * \log\frac{N - n + 0.5}{n + 0.5} * \frac{qtf}{k_3 + qtf}$$

***Equation 14: BM11 Weighting Equation***

Equation 13 and Equation 14 can be combined into a single function known as BM25 that allows for numerous variations. The term frequency component is implemented as:

$$tf = \frac{tf^c}{K^c + tf^c}$$

***Equation 15: Term Frequency Component in BM25***

where $K = k_1\left((1 - b) + b\frac{d}{\Delta}\right)$. Therefore, if $c = 1$, and $b = 1$ gives the equation for BM11 and if $b = 0$ gives the equation for BM15. Different values of $b$ give a mix of the two equations. BM25 is referred to as $BM25(k_1, k_2, k_3, b)$. Therefore, the whole weight equation becomes the following:

$$w = \frac{tf^c}{K^c + tf} * \log\frac{N - n + 0.5}{n + 0.5} * \frac{qtf}{k_3 + qtf} \; and \; k_2 * nq\frac{\Delta - d}{\Delta + d}$$

***Equation 16: BM25 Weighting Equation***

There is an item $k_2 * nq\frac{\Delta-d}{\Delta+d}$ in Equation 16 that is called the correction factor. More details on the correction factor can be found in paper [53]. The $k_2$ value is usually set as *0* in experiments.

## 3.4. Semantic Matching Models

This thesis presents two semantic matching models that have been developed and used throughout the research in this thesis. Sectuin 3.6.1 and Section 3.6.2 describe the two semantic matching models; Term-Based Matching, and CUI-Based Matching respectively. These models determine the relevancy score of the extracted medical concepts from the EMR dataset and the extracted medical concepts from the topics.

### *3.4.1 Term-Based Matching*

Term-Based matching refers to leveraging the actual medical terminology from the EMRs and topics to form medical concepts. The concepts from the ERMs are then compared to the ones in the topics to discover whether there is a relationship between them, and their degree of relevancy between them. Equation 17 is used to establish document relevancy. Given the disease-based concept vector of topics $Q_{disease}$ = ($C_1$, $C_2$,

..., $C_m$) and the disease-based EMR concept vector $D_{disease}$ = ($C_1$, $C_2$, ..., $C_n$), the conceptual score of the EMR with those that overlap with the topic, and rank them according to the overlapping concept weights in the EMR as follows:

$$Score_{disease}(D) = \sum_{C_i \in Q_{disease}} \frac{1}{n} \sum_{C_j \in D_{disease} \,\&\, C_i^T \cap C_j^T \neq 0} w(C_j, D_{disease})$$

***Equation 17: Term-Based Matching Model***

$Score_{disease}(D)$ is the conceptual score of a given EMR that is obtained using the disease-based index. $Q_{disease}$ is the concept-based representation of a query concept given the disease index. $C_i$ is a concept that exists in query $Q_{disease}$. $C_i^T$ is the term-based representation of concept $C_i$. The variable $n$ is the total number of concepts in the query $Q_{disease}$. $w(C_i, Q_{disease})$ represents the weight of concept $C_i$ in the query $Q_{disease}$. $D_{disease}$ is the diseased-based concept vector of a EMR. $C_j$ is a concept in EMR $D_{disease}$. $C_j^T$ is the term-based representation of concept $C_j$. $w(C_j, D_{disease})$ represents the weight of concept $C_j$ in EMR $D_{disease}$ [54]. The final score of EMR is based on adding the weights of both the procedure-based and disease based EMR score, as indicated by Equation 19 [54]. Equation 17 also calculates procedure-based weights by replacing all instances of *disease* with *procedure*.

### *3.4.2 CUI-Based Matching*

CUI-Based matching refers to using a unique ID provided by the National Cancer Institute known as the Concept Unique Identifier that represents a medical concept. There is a unique ID for every almost every medical concept ranging from diseases to treatments. As such, the CUI identifies medical concepts that are discovered from the EMRs and topics. They are then used to determine whether there is a relationship between the EMRs and topics, and the degree of relevancy that the EMRs have with the topics. The following equation is used to establish document relevancy:

$$Score_{disease}(D) = \sum_{C_i \in Q_{disease}} w(C_i, Q_{disease}) * w(C_i, D_{disease})$$

***Equation 18: CUI-Based Matching Model***

$Score_{disease}(D)$ is the conceptual score of a given EMR that is obtained using the disease-based index. $Q_{disease}$ is the CUI-based representation of a query concept given the disease index. $C_i$ is a CUI that exists in query $Q_{disease}$. $w(C_i, Q_{disease})$ represents the weight of CUI $C_i$ in the query $Q_{disease}$. $D_{disease}$ is the diseased-based CUI vector of a EMR. $C_j$ is a CUI in EMR $D_{disease}$. $w(C_j, D_{disease})$ represents the weight of CUI $C_j$ in EMR $D_{disease}$ [54]. Equation 18 also calculates procedure-based weights by replacing all instances of *disease* to *procedure*. The final score of an EMR is based on adding the

weights of both the procedure-based and disease-based EMR score, as indicated by Equation 19.

$$Score_f(D) = Score_d(D) + Score_p(D)$$

***Equation 19: Final Score Model***

## 3.5. The Need for a New Solution

There is a need to have specialized methodologies for information retrieval in the healthcare domain because of its uniqueness and distinctiveness of the verbiage used in the field and with EMRs containing tactic knowledge, it is not always clear what the meaning of the information is that was left by healthcare professionals in EMRs. As explained in Section 2.1, there is a lack of specialized information retrieval systems and methodologies that are intended to examine medical documentation. There are numerous information retrieval systems and methodologies that are available for general-purpose use, and they have been successfully used in the past to improve various aspects of numerous fields of research. However, it is reasonable to assume that by developing specialized methodologies and information retrieval systems for a specific domain, the performance of those methodologies and systems would be better than the existing general-purpose methodologies and systems. This thesis aims to be a step towards achieving this.

# Chapter 4

# Experimental Setting and Implementation

## 4.1. The Datasets

The dataset that was used to conduct the work in this thesis is made available for research through the University of Pittsburgh BLULab NLP Repository for the TREC 2011 Medical Records track. The purpose of the research is to advance research pertaining to content-based access to free-text fields of EMRs [55]. The dataset contains 101,711 de-identified EMRs. Figure 1 illustrates a typical EMR in the dataset. As illustrated in Figure 1, each EMR in the dataset contains the following elements:

- *checksum* – A unique value which is used to represent the EMR.
- *subtype* – The subtype of the EMR.

- *type* – The type of the EMR .
- *chief_complaint* – The primary complaint that the patient has experienced.
- *admit_diagnosis* – The diagnosis of the patient upon being admitted to the healthcare institution.
- *discharge_diagnosis* – All the diagnosis that was given to the patient up to the point of being discharged from the healthcare institution.
- *year* – The year in which the EMR has been generated.
- *download_time* – The time that the EMR has been downloaded off the EMR system.
- *update_time* – The last time that the EMR has been updated.
- *deid* – The deidentfication of a EMR with explicit identifiers being removed, replaced or hidden to ensure data cannot be leveraged to identify a patient.
- *report_text* – The free text portion of the EMR that was written by a medical professional who examined the patient.

The EMR dataset contains various demographical information of the patient population. Table 1 demonstrates the gender distribution, Table 2 exhibits the age distribution, Table 3 displays the age distribution of the patients for each gender, and Table 4 shows other interesting information contained within the EMR dataset. Each

patient is identified in the EMR dataset through a value known as a Visit ID, and each of these Visit IDs are a part of one or more EMRs. In reality, it is possible for a patient to have more than one Visit ID, but there is no viable ways to identify what set of Visit IDs belong to any patient because the EMRs are extremely de-identified. As such, the assumption must exist that each Visit ID in the EMR dataset connects to a unique patient. Figure 2 further exemplifies information contained within each Visit ID.

In addition to the EMR dataset, participants were given 34 topics where the objective was to identify which EMRs are the most relevant to a given topic. Each of the 34 topics contained various pre-defined medical concepts such as diseases, conditions, treatments, and procedures. For example, topic 111 is looking to identify EMRs that meets the following criteria: "Patients with chronic back pain who receive an intraspinal pain-medicine pump". Thus, the deliverables expected for topic 111 would be a list of EMRs from the EMR dataset in order of decreasing relevancy of meeting the criteria for the topic.

```
- <report>
    <checksum>20051127OP-cQsnkGlmzZbN-848-71049104</checksum>
    <subtype>ORTHO OP</subtype>
    <type>OP</type>
    <chief_complaint>LFT LEG PAIN</chief_complaint>
    <admit_diagnosis>730.27</admit_diagnosis>
  - <discharge_diagnosis>
      250.81,707.14,403.91,428.0,711.06,276.7,424.1,416.0,730.27,250.51,362.01
    </discharge_diagnosis>
    <year>2007</year>
    <downlaod_time>2009-10-05</downlaod_time>
    <update_time/>
    <deid>v.6.22.08.0</deid>
  - <report_text>
      [Report de-identified (Safe-harbor compliant) by De-ID v.6.22.08.0] **INSTITU
      ASSISTANT(S): **NAME[RRR QQQ], M.D. ATTENDING PHYSICIAN: **NAM
      DEBRIDEMENT OF LEFT KNEE. ANESTHESIA: General. COMPLICATIONS
      The patient is a **AGE[in 60s]-year-old female with a history of end-stage rena
      length. I spoke to her and her daughter about the risks and benefits of surgical
      that irrigation and debridement of septic arthritis is indicated and we talked abo
      as the patient. She was taken to the operating room where she was placed su
      carefully placed high in the left thigh. The left leg was then prepped and drape
      inflated. A small approximately 5 cm parapatellar arthrotomy was performed sh
      fluid, the knee was pulse irrigated with 3 L of solution. After this, we reexamine
      accomplishing this, the arthrotomy was closed with 0 Vicryl in a watertight fash
      anesthesia. Earlier the tourniquet had been deflated prior to closure. There we
      **NAME[WWW XXX], M.D. MK/ga D: **DATE[Nov 27 2007] 18:23:27 T: **DAT
      ADMISSION DATE: **DATE[Nov 22 2007] SURGERY DATE: **DATE[Nov 27
    </report_text>
  </report>
```


*Figure 1: An Illustration of a Typical EMR in the Dataset*

| | |
|---|---|
| **Male** | **25.6%** |
| **Female** | **27.3%** |
| **Unknown Gender** | 47.1% |

*Table 2: Gender Distribution for Patients*

| Teen's | 1.1% | 20's | 3.9% |
|---|---|---|---|
| 30's | 3.7% | 40's | 5.8% |
| 50's | 7.2% | 60's | 7.5% |
| 70's | 7.6% | 80's | 7.3% |
| 90's | 1.4% | Unknown | 54.5% |

*Table 3: Age Distribution for Patients*

## 4.2. Leveraged Tools

To achieve the goal outlined in Section 1.2, it was necessary to use a wide variety of available tools to determine which one would yield the most accurate list of relevant EMRs from the dataset to a given topic. The tools that were employed to reach this goal were BioLabeler[16], OpenCalais[17], MetaMap[18], and Terrier[19].

| Male | | Female | | Unknown | |
|---|---|---|---|---|---|
| Teen's | 2.1% | Teen's | 2.0% | Teen's | 0.0% |
| 20's | 7.3% | 20's | 7.3% | 20's | 0.1% |
| 30's | 7.4% | 30's | 6.4% | 30's | 0.2% |
| 40's | 11.6% | 40's | 10.0% | 40's | 0.3% |
| 50's | 14.4% | 50's | 12.5% | 50's | 0.3% |
| 60's | 14.4% | 60's | 13.3% | 60's | 0.5% |
| 70's | 13.6% | 70's | 14.1% | 70's | 0.5% |
| 80's | 11.0% | 80's | 15.4% | 80's | 0.5% |
| 90's | 1.6% | 90's | 3.3% | 90's | 0.1% |
| Unknown | 16.6% | Unknown | 15.7% | Unknown | 97.5% |

*Table 4: Age Distribution for Patients for each Gender*

[16] http://www.biolabeler.com/bioLabeler/
[17] http://www.opencalais.com/
[18] http://metamap.nlm.nih.gov/
[19] http://www.terrier.org/

| Number of Unique Visits | 17199 |
|---|---|
| Number of Unique Admit Diagnosis | 1461 |

*Table 5: Other Information*

BioLabeler is a free web service designed to create concept associations to any given text. It performs stemming, stop word removal, and returns candidate National Cancer Institute (NCI) [20] concepts with corresponding confidence and normalized weights. BioLabeler performs this function by extracting UMLS concepts from biomedical texts such as scientific paper abstracts, experiments descriptions, or medical notes. BioLabeler can also be used to annotate BioMedical Literature or to index large documents. BioLabeler allows one to filter results by extracting specific medical concepts from specific UMLS sources and specific types of medical concepts, such as extracting only Diseases from the biomedical text.

OpenCalais is a free web service that creates semantic metadata from submitted content. It leverages natural language processing, machine learning, and other various methods to analyze documents to identify various entities contained within. The entities that were specifically leveraged when utilizing OpenCalais were MedicalCondition and Medical Treatment. Medical Condition extracts references to human and medical conditions like diseases, disorders, and syndromes, while MedicalTreatment extracts

[20] http://www.cancer.gov/

references to medical treatments like procedures, treatments and therapeutics provided to any medical condition.

MetaMap was developed by Dr. Alan (Lan) Aronson at the National Library of Medicine (NLM), and it is an exceedingly configurable software to map biomedical text to the UMLS ontologies to discover various medical concepts contained within the biomedical text. MetaMap accomplishes this task by taking a knowledge-intensive methodology based on natural language processing and computational-linguistic techniques. MetaMap is one of the foundations of the NLM's Medical Text Indexer, used for semiautomatic and fully automatic indexing of biomedical text at NLM.

Terrier was developed at the School of Computing Science at the University of Glasgow. It is a flexible, efficient, and effective open source search engine that is readily deployable on large-scale collections of documents. Terrier leverages state-of-the-art indexing and retrieval functionalities, and it provides an ideal platform for the rapid development and evaluation of large-scale retrieval applications. Some of the indexing strategies built into Terrier include multi-pass, single-pass and large-scale MapReduce. Some of the built-in retrieval approaches include Divergence from Randomness, BM25F, and Markov Random Fields.

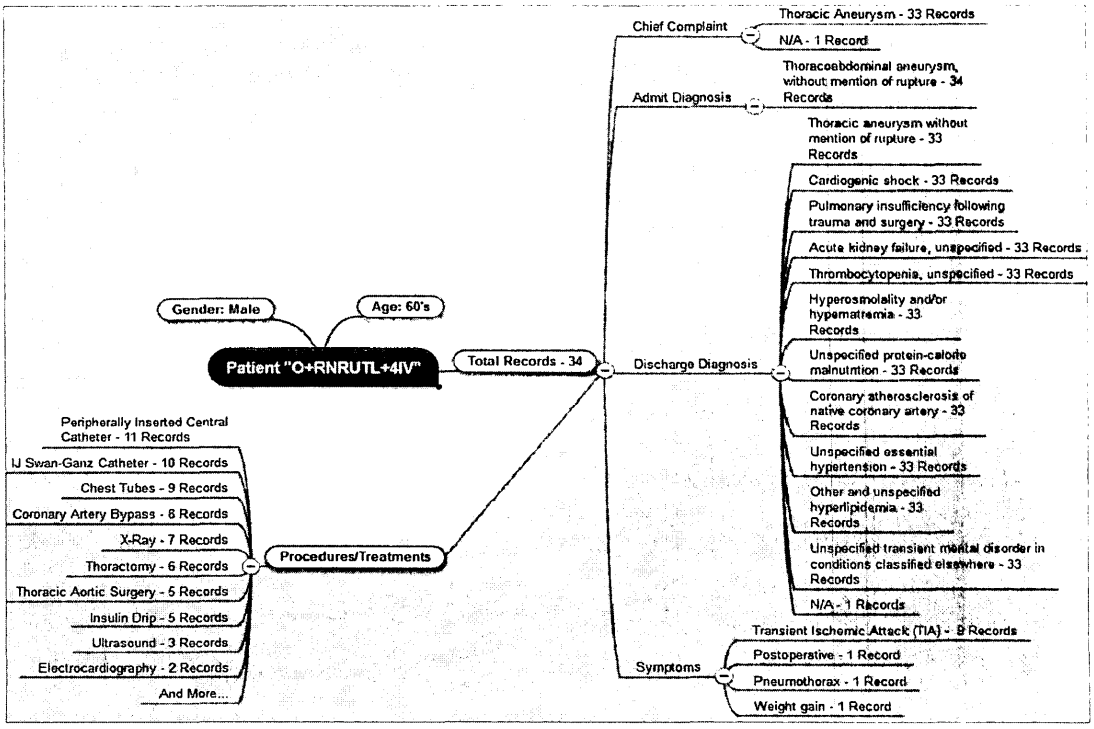


***Figure 2: Visual Representation of Information Contained Within a Visit ID***

## 4.3. Preprocessing the Dataset

To use the dataset with the tools presented in this thesis, it was necessary to preprocess the EMRs to get the most optimal results is possible. As mentioned earlier, BioLabeler is capable of extracting UMLS concepts from a multitude of biomedical texts; however, only those that are in plaintext are able to be processed by BioLabeler. The EMRs in the dataset are in an XML file format and this type of file is made up of metadata and various XML markup. It is necessary to convert the free-text portion of the EMRs in the dataset into plaintext before BioLabeler can use any data from the EMRs.

Failing to do this would result in BioLabeler failing to process any of the EMRs given to it. Thus, it is necessary to preprocess the dataset before it can be used.

The EMRs are in XML file format, and to efficiently leverage the contents of the records programmatically it was necessary to convert them to a relational database. This conversion would allow one to identify records with specific content with greater efficiency, opposed to accessing the records and processing the content within the XML file version multiple times. In addition, converting the dataset into a database also permits accessing the contents through a web-based portal. The ability to use the dataset through such a platform allows users to access the content through any conventional web browser, and opens the possibility to be able to work with the records without needed any technical knowledge if a system were to be created to manipulate the data. Furthermore, any new records that would be added to the database can be leveraged immediately for improving the prediction capabilities of the system. This is because as new records are added to the database, they will not need to be processed like an XML file would, as the records are already in a format that the system will be able to take advantage of when new queries are made.

## 4.4. Building Indexes

An index is a set of where each item in the index specifies one record of a dataset and it contains information about its address. Multiple indexes were generated to be able

to successfully leverage the various tools used in this thesis. Each of the tools generated their own indexes containing unique information. The generation of each index by the tools is dependent on the content submitted to the tool, and various information retrieval techniques are automatically leveraged to ensure that proper ontological concepts are found within the dataset.

### *4.4.1 BioLabeler Indexes*

BioLabeler creates indexes based on the biomedical text submitted to it. As explained earlier, Biolabeler is only able to use plaintext; therefore, it was not possible to submit each EMR in the dataset to BioLabler as each EMR is in XML format and contains information other than biomedical text. As a result, only the free text section of the EMR from the dataset is submitted to BioLabler, specifically the *report_text* section. BioLabler leverages UMLS sources to generate medical concept ontologies found within the biomedical text submitted to it. BioLabeler is flexible in the way it allows one to construct medical ontologies. The system allows one to select any combination of the forty-four available UMLS sources one wishes. These selections have a significant impact on how the system forms medical ontologies, and which concepts are identified within the biomedical text. In addition, BioLabeler also provides the flexibility in selecting any combination of one-hundred thirty-five semantic types that can be used in extracting medical concepts.

The total generated number of indexes consists of two sets of four indexes, where the indexes have limitations of being tied to ten specific semantic types to retrieve candidate concepts. Six of these extract disease concepts and the remaining four extract procedure concepts [54]. The six disease semantic types consist of Acquired Abnormality, Congenital Abnormality, Disease or Syndrome, Experimental Model of disease, Mental or Behavioral Dysfunction, and Neoplastic Process. The four procedure semantic types consist of Diagnostic Procedure, Health Care Activity, Laboratory Procedure, and Therapeutic or Preventive Procedure [54]. The first set of indexes generated leveraged all forty-four UMLS sources, the second set of indexes utilized the MedlinePlus Health Topics UMLS source, and the final set of indexes used the Medical Subject Headings UMLS source. The four indexes in each set of indexes consisted of: one for procedure medical concepts located in the EMR dataset, one for disease medical concepts located in the EMR dataset, one for procedure medical concepts located in the topics, and one for disease medical concepts in the topics [54]. Creating a custom Java application is necessary to interface with the BioLabeler system. The source code for the Java applications can be located in Appendix C.1.

Figure 3 exemplifies the output BioLabeler returns when submitting biomedical text. To receive the output, BioLabeler expects biomedical text; the desired UMLS sources; the semantic type's that need to be found; and an email address of the individual using the service. Upon submission of information to BioLabler, the system performs

stemming and stop word removal. It then returns its results of candidate concepts in descending order based on the score it calculated for each concept using the cosine similarity between the biomedical text and candidate concepts [56]. Contained in the said indexes is a subset of the information is found in Figure 3, specifically the National Cancer Institute Concept Unique Identifier (CUI), medical concept abbreviations, normalized weight, and weight [54].

### *4.4.2 OpenCalais Indexes*

OpenCalais creates indexes based on the biomedical text submitted to it. The system is able to process various documents such as plaintext, HTML, or XML; however, only the unstructured biomedical text from all EMR's in the dataset is sent as the focus of the research concentrates on extracting medical concepts from unstructured biomedical text. Although free to use, OpenCalais is a black box system and the specific logistics of how the system detects concept entities is not known. OpenCalais allows one to select the type of semantic metadata definitions and descriptions one wants to use. In total, there are one-hundred twenty-two semantic metadata types available for extraction from the documents submitted to OpenCalais.

One created index generated by OpenCalais used two specific semantic metadata definitions and descriptions. The first one was MedicalCondition, which extracts references to human medical conditions such as diseases, disorders, and syndromes. The

second one was MedicalTreatment, which extracts references to medical treatments such as procedures, treatments, and therapeutics provided to any medical condition.


```
{
  "concepts": [
    {
      "concepts": "Hospital-Physician Relations#Hospital Physician Relations#
      "cui": "C0242799",
      "normWeight": 0.2415234,
      "numTerms": 3,
      "sabs": "MSH#MSH#MSH#MSH#MSH",
      "termBitmap": "0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0
      "termList": "physician,hospit,relat",
      "totTerms": 3,
      "weight": 6.6004139999999998
    },
    {
      "concepts": "Palliative Care#Care, Palliative#Palliative Treatment#Pall
      "cui": "C0030231",
      "normWeight": 0.22696540000000001,
      "numTerms": 2,
      "sabs": "MSH#MSH#MSH#MSH#MSH",
      "termBitmap": "0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
      "termList": "palli,care",
      "totTerms": 2,
      "weight": 6.2025690000000004
    },
    {
      "concepts": "Continuity of Patient Care#Care Continuity, Patient#Patien
      "cui": "C0009853",
      "normWeight": 0.218976,
      "numTerms": 3,
```


***Figure 3: An Illustration of BioLabeler's Output of an Examination of Biomedical Text***

Figure 4 exemplifies the output OpenCalais returns when an EMR is submitted to it. To receive the output, OpenCalais expects a document along with the semantic metadata being sought for. Upon submission of the information to OpenCalais, the system uses natural language processing, machine learning, and other methods to analyze the submitted documents to find entities in it. Aside from this information, not much is

known about the exact workings of OpenCalais as the system is a black box. Contained in Figure 4 is a sample of the information in the indexes, specifically the MedicalCondition name or MedicalTreatment name, and the relevance score. It is necessary to create a custom Java application to interface with the MetaMap system. The source code for the Java application can be located in Appendix C.2.

```
<OpenCalaisSimple>
- <Description>
    <calaisRequestID>96cb5336-3554-fa82-134f-4c9bd7b18cb3</calaisRequestID>
    <id>http://id.opencalais.com/AM02TCiPNu9IKBbFmJ3KMA</id>
  - <about>
      http://d.opencalais.com/dochash-1/b154efdc-821c-3f85-ba58-9c2a6d03c466
    </about>
    <docTitle/>
    <docDate>2007-07-05</docDate>
  </Description>
- <CalaisSimpleOutputFormat>
    <MedicalCondition count="2" relevance="0.336">Right forearm cellulitis</MedicalCondition
    <MedicalCondition count="2" relevance="0.277">distal paresthesias</MedicalCondition>
    <MedicalCondition count="2" relevance="0.299">chills</MedicalCondition>
    <MedicalCondition count="2" relevance="0.148">erythema</MedicalCondition>
    <Organization count="2" relevance="0.354">Emergency Department</Organization>
    <Product count="2" relevance="0.257">Prograf</Product>
    <City count="1" relevance="0.053">Job</City>
```

***Figure 4: An Illustration of OpenCalais's Output of an Examination of an ERM***

### *4.4.3 MetaMap Indexes*

MetaMap creates indexes based on the biomedical text submitted to it. The system is able to accept ASCII only input, unformatted English free text, MEDLINE citations, input records delimited by blank lines, single-line delimited inputs, and single-line delimited inputs with identifiers. As the system has clear limitations in terms of the

type of inputs that it is able to accept, it was not possible to submit each EMR in the dataset to MetaMap as each EMR is in XML format and contains information other than what the system accepts. As a result, only the unstructured plaintext portion of the given EMR record from the dataset was sent to MetaMap. MetaMap uses a UMLS Metathesaurus to map biomedical text to candidate medical concepts, in addition to natural language processing methodologies along with computational linguistic techniques. MetaMap uses the 2011AA UMLS source that contains over 2.4 million concepts and 8.8 million unique concept names from one-hundred sixty source vocabularies that include MedlinePlus, Medical Subject Headings (MeSH), Medical Dictionary for Regulatory Activities (MedDRA), RxNorm, and Systematized Nomenclature of Medicine – Clinical Terms (SNOMED CT) [57]. Furthermore, the 2011AA UMLS source leverages the SPECIALIST Lexicon that is simply an English lexicon that includes biomedical terms as well as commonly occurring English Words.

Two indexes were created using MetaMap, the first being an index of all medical concepts located in the EMRs in the dataset, and the second being all medical concepts located in the topics. Creating a custom Java application is necessary to interface with the MetaMap system and it used the default system properties when generating the indexes.

### *4.4.4 Terrier Indexes*

Terrier is able to create indexes using a variety of sources. The information retrieval system accepts a variety of corpora formats including but not limited to various TREC formatted corpora; and various WARC corpora's that include HTML, Microsoft Word/Excel/PowerPoint, PDF, text documents, and XML. Terrier is able to process XML documents and because the EMR dataset is composed of XML records, Terrier is able to process the raw records without the need for preprocessing. Terrier has the ability to leverage three indexing methodologies when indexing documents given to it. These methodologies are classical two-pass indexing leveraging BasicIndexer, classical two-pass indexing leveraging BlockIndexer and, single-pass indexing.

The classical two-pass indexing leveraging BasicIndexer methodology for indexing iterates through the submitted dataset, tokenizing terms to add to its index. The system performs stemming and stop word removal on all tokens, and once the process is complete, the system generates three data structures. The three data structures are a DirectIndex, a DocumentIndex, and a Lexicon. The DirectIndex is a compressed file that contains all terms found in each examined document and it is used for automatic query expansion. The DocumentIndex is a fixed-length entry file where the storage of information about the examined documents occurs. This file contains information such as the number of indexed tokens (also known as document length), the identifier of a document, and the offset of its corresponding entry in the direct index. The Lexicon is

also a fixed-length entry file that stores information about the vocabulary of the examined dataset. When all three data structures are generated, Terrier creates an InvertedIndex data structure that contains information of the DirectIndex but with inverted values.

The classical two-pass indexing leveraging BlockIndexer methodology for indexing has the same functionality as the first with the exception that it leverages a larger DirectIndex and InvertedIndex for storing the positions that each token occurs at in each document. The benefit of using this methodology over the first is that this methodology allows a query to use term position information. By adding proximity search, it is possible to add restrictions when searching for multiple terms and ignore any matches whom do not abide to the restrictions. For example, setting proximity of 10 would only return matches of terms whom are within 10 words of each other.

The single-pass indexing methodology for indexing stores its data in computer memory opposed to data structures. In the event the computer begins to lack memory, this methodology will store data on the computer's hard disk. Once the dataset has been fully examined, all the data merges into one index file. It is faster to read and write to a computer's memory than to its storage, and so the computer memory is used to contain the data necessary to generate the index files at faster speeds.

When creating the index that was used to examine Terriers performance in comparison to the other three tools used in the research, each EMR in the dataset was submitted to Terrier in its XML format. As stated earlier in this section, Terrier is able to

accept XML documents and since the dataset consists of EMRs in XML format, and the methodology that was used for indexing the dataset was the first methodology mentioned in this section; classical two-pass indexing leveraging BasicIndexer.

## 4.5. Preparing Queries

Preparation of queries is necessary to leverage the tools used in this thesis. BioLabeler is an online service where the user manually inputs data into their web interface and it returns results. Although BioLabeler does provide a REST API, complications arise when attempting to use the official API with large biomedical text. Due to this circumstance, and because there are over 100,000 EMRs with large biomedical text, it is necessary to create a program that will simulate the manual process of plaintext submission to BioLabeler through the HTTP post protocol. In order to simulate the manual process, it is necessary to examine the information sent to BioLabeler along with what kind of information BioLabeler is returning to the browser. To accomplish this task, a tool that examines HTTP requests in a web browser is needed. Firebug for Mozilla Firefox was used to accomplish this task. Firebug is able to capture information being sent from and sent to Firefox through the BioLabeler website, and this information allows one to simulate the HTTP post request done by the BioLabeler website. Numerous samples of Java code that simulates the HTTP post request is in Appendix C.1. BioLabeler returns an index of identified medical concepts according to the previously mentioned criteria in Section 4.4.1. It is necessary to compare each of the

medical concepts that are found in the Topics provided by TREC to the medical concepts that are found in the EMR dataset. More information on the actual runs and queries are in Section 5.2.

OpenCalais provides an API for programming languages that will allow one to take advantage of their services. To use the API, it is necessary to create an application that will take advantage of the API that will automatically submit queries to the system. Using the API, the application submits the biomedical text to OpenCalais for processing and returns an index of the requested information. In this case, the requested information is the conditions and treatments contained within the submitted biomedical text. A comparison of the medical concepts that were found in the topics provided by TREC to the medical concepts that are found in the EMR dataset is then conducted. More information on the actual runs and queries are in Section 5.2.

MetaMap is a standalone application that needs to be installed on a computer in order to be used. To send queries to MetaMap, it is necessary to extract only the *report_text* section of the EMRs in the dataset as the restrictions of MetaMap prevent XML documents being examined by the system. Furthermore, it was necessary to create a Java application that will leverage the MetaMap Java packages that are installed on the computer with the MetaMap system. These packages allow one to interface with the MetaMap system and it is used to examine the materials sent to it. More information on the actual runs and queries are in Section 5.2.

# Chapter 5

# Results and Evaluation

## 5.1. Evaluation of Results

The capability to define the effectiveness of results is a challenging task, especially in the healthcare domain as leveraging EMRs to perform any knowledge discovery is a fresh field. TREC provides clear guidelines in regards to evaluating the results received from the techniques participants use in their work. TREC also provides all the necessary materials needed to meet their guidelines. Part of the materials given by TREC includes an evaluation script that is to be run in Linux. This script evaluates the performance of methodologies that are used to generate candidate concepts. Such an evaluation script is known as the golden standard. One must also establish a baseline using any established methodology that is used today in the field of information retrieval. Section 5.1.1 describes more details regarding the generated baseline. The goal of evaluating the generated results is to determine the impact of semantic indexing and the

conceptual matching models [54] that are presented in Section 4.6. A detailed explanation of the criteria used for evaluating the conceptual matching models and semantic indexing is further presented in section 5.1.2.

### *5.1.1 Baseline*

The methods and procedures used for retrieving conceptual representations of the EMRs in the dataset is of utmost importance to ensure achieving high performance and accuracy. To determine performance and accuracy it is necessary to establish a baseline. With the given dataset, it is possible to treat the problem domain as a classical data-mining problem where one of the most critical fields in the EMRs is the free text field *report_text*. As such, BM25 is used to generate the baseline as this ranking equation is one of the most widely used information retrieval equations, and has historically been used as a strong baseline in information retrieval [58]. The BM25 ranking equation has various variables that are adjustable, and the parameters were set as follows: *b* to *0.75*, *k1* to *1.2,* and *k3* to 8. These parameter values are the standard values used with the equation [59]. Figure 5 illustrates the established baseline using the BM25 ranking equation with the said variables. The illustrated results are in a scale of 0 – 1, where 0 indicates no documents from the results are relevant using the performance measure being looked at, and 1 indicates all documents from the results are relevant using the performance measure being looked at. To exemplify this further, a performance measure

resulting in a score of 0.1234 indicates that 12.34% of the documents in the results are relevant using the performance values being observed.

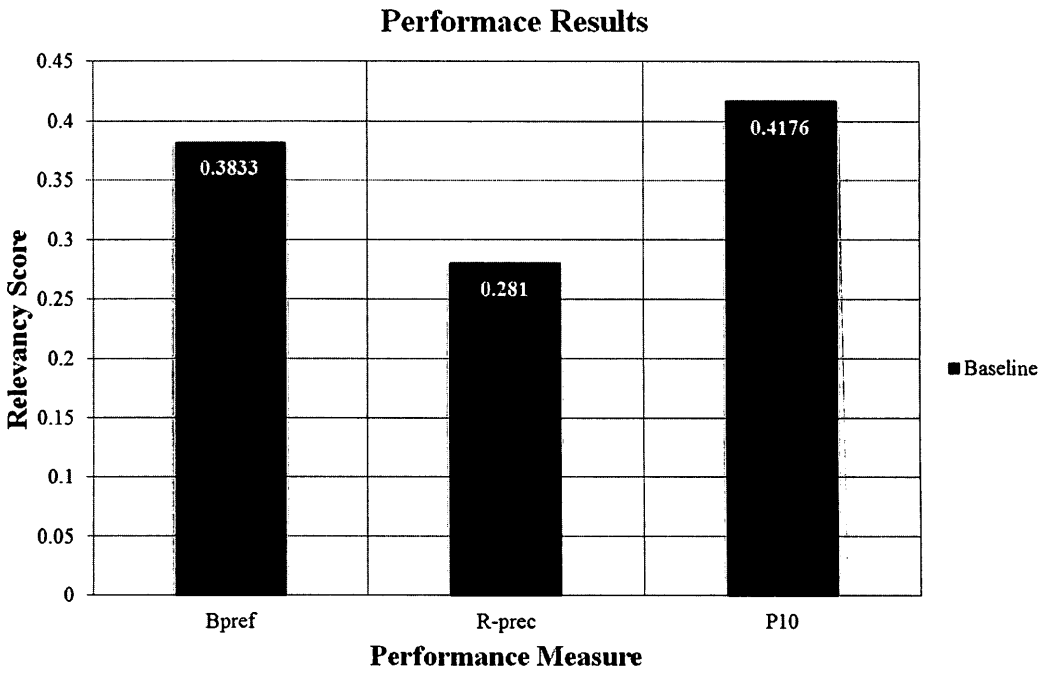


***Figure 5: The Performance Results of the Baseline***

### *5.1.2 Performance Criteria*

The materials provided by TREC generate a multitude of performance measures that outline the performance criteria to determine the effectiveness of the semantic matching models presented in this thesis. Figure 6 illustrates all potential performance measures; however, the official performance measures for evaluating the impact of semantic indexing and conceptual matching models as presented by TREC are Bpref, R-

prec and P10. These measures are generated by using the official evaluation package provided by TREC named *TREC_eval*. As the performed work relies on the materials provided by TREC, it is only logical to use the official performance measures as set by the conference.

The Bpref measure is an information retrieval metric function that is based on binary relevance [60]. The R-Prec measure is a function where measurements of document percision after R documents are retrieved occur, where R is the number of relevant documents for a given topic [61]. The P10 measure counts the number of relevant documents in the top 10 documents in the ranked list returned for a given topic [61].

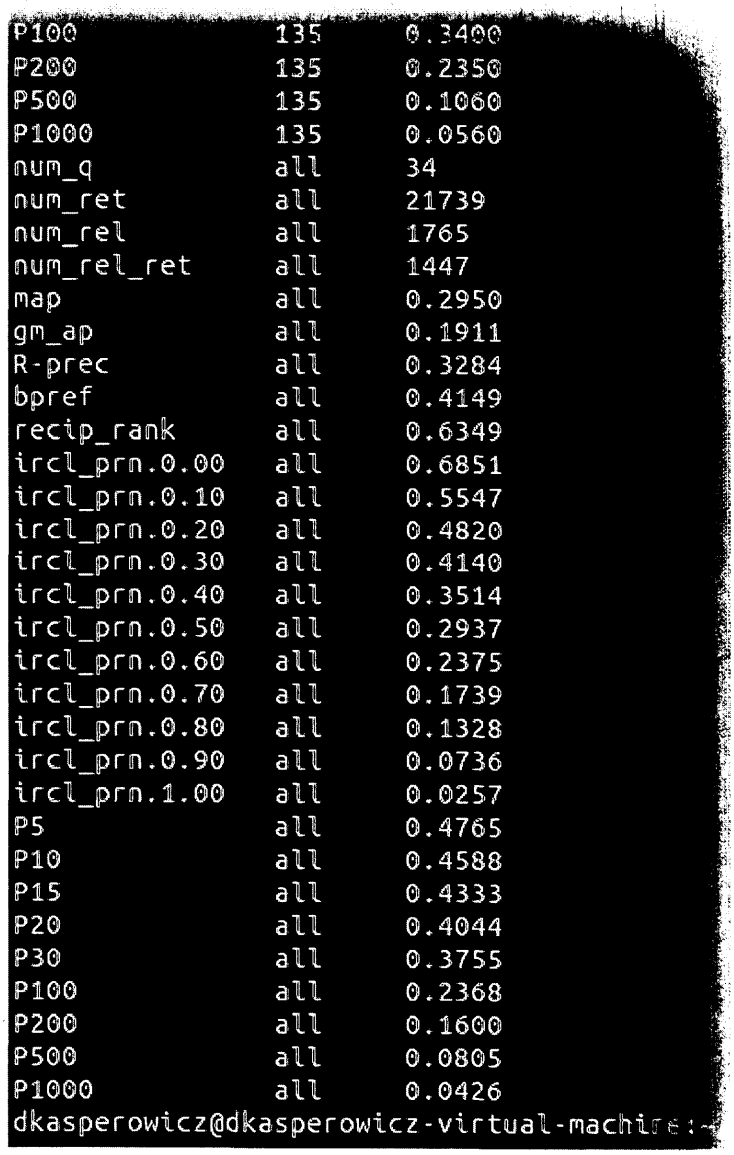


*Figure 6: Illustration of All Potential Measures and their Score's for a Sample Run*

```
115 Q0 BClmUqGjJ1DC 12088 251753 bioLabeler
115 Q0 8+8elzEpdtef 12089 251250 bioLabeler
115 Q0 8fJs96Tm7BX0 12090 247170 bioLabeler
115 Q0 7ioM2vkwfCjE 12091 247075 bioLabeler
115 Q0 wLqThLq8pozb 12092 246204 bioLabeler
115 Q0 o9H2m2P5jmqf 12093 244566 bioLabeler
115 Q0 MAxapVPi2YYU 12094 242514 bioLabeler
115 Q0 3dH7ynYeOsaT 12095 241995 bioLabeler
115 Q0 +al4GKiXiK5v 12096 239843 bioLabeler
115 Q0 eKUbBrgAiKRR 12097 239648 bioLabeler
116 Q0 pHuhEAqxIz6G 1 99855162 bioLabeler
116 Q0 eFns/XP0bZlv 2 99844655 bioLabeler
116 Q0 +0yUT/ag81Mm 3 99751359 bioLabeler
116 Q0 l2vLZcOiCoai 4 99726639 bioLabeler
116 Q0 zWUeBNzOiS/j 5 99676021 bioLabeler
116 Q0 hEtOMYtVJDou 6 99560226 bioLabeler
116 Q0 m2Ewz7MALrY+ 7 99474461 bioLabeler
116 Q0 4jgWgXnmD6od 8 99401916 bioLabeler
116 Q0 dvK5I+7diIKt 9 99350359 bioLabeler
116 Q0 QPHxkg5JCoUr 10 99289524 bioLabeler
```

***Figure 7: An Illustration of Results from BioLabeler and Terrier Using Matching Models from 4.6***

## 5.2. Results

Each tool described in Section 4.2 was used to generate multiple runs with various settings and configurations. Six runs are completed using the indexes generated by BioLabeler. The first run uses the term-based matching model presented in Section 4.6.1 to compute a relevance score for each EMR in the dataset with respect to a given topic [54]. The remaining five runs using BioLabeler employed the CUI-based matching model presented in Section 4.6.2 to compute a relevance score for each EMR in the dataset with respect to a given topic [54]. The six runs conducted using BioLabeler are further explained in Table 5.

Nine runs are completed using the indexes generated by Terrier. As Terrier is a well-established information retrieval system that uses classical and reputable methodologies, the system contains a series of models capable of computing their own relevance score. As such, the methodologies built into Terrier are used instead of the developed methodologies illustrated in Section 4.6 to compare the results between the models presented in section 4.6 and the established models used by Terrier. The nine runs conducted using Terrier are further described in Table 6.

| Run | Description |
|---|---|
| Bio-TB_DR | Term-based weighting model with dropped records |
| Bio-CUI_NDR | CUI-based weighting model without dropped records |
| Bio-CUI_NDR_TC | CUI-based weighting model without dropped records using the top concept |
| Bio-CUI_NDR_TC5 | CUI-based weighting model without dropped records using up to the 5 top concepts |
| Bio-CUI_NDR_TC_MSH | CUI-based weighting model without dropped records using the top concept with only the Medical Subject Headings UMLS source |
| Bio-CUI_NDR_TC5_MSH | CUI-based weighting model without dropped records using the top concept with only the Medical Subject Heading UMLS source |

***Table 6: Runs Conducted with BioLabeler***

Table 7 presents the evaluated results of each conducted run. Figure 7 is an exemplification of the results generated using BioLabeler and Terrier with the matching models presented in Section 4.6. Both OpenCalais and MetaMap were unable to produce

any medical concepts that matched topics and EMRs. Hence, it is not possible to determine or analyze how the matching models work with these two indexing tools.

## 5.3. Analysis and Discussion

The first two runs conducted in the research were Bio-TB_DR and Bio-CUI_NDR. The purpose for conducting these initial runs was to determine which of the two proposed methods would prove to have the superior performance when utilizing the BioLabeler system. In addition, a goal was to ascertain if by utilizing BioLabeler alongside the matching models if it would yield superior performance than an established information retrieval ranking function. Figure 8 illustrates the performance between the Baseline, Bio-TB_DR and Bio-CUI_NDR runs.

| Run | Description |
|---|---|
| Ter-PL2 | PL2 ranking model |
| Ter-BM25 | BM25 ranking model |
| Ter-BB2 | BB2 ranking model |
| Ter-DFI0 | DFI0 ranking model |
| Ter-DRF_BM25 | DRF_BM25 ranking model |
| Ter-DFRee | DFRee ranking model |
| Ter-DirLM | DirichletLM ranking model |
| Ter-DHL | DHL ranking model |
| Ter-DLH13 | DLH13 ranking model |

***Table 7: Runs Conducted with Terrier***

When examining run Bio-TB_DR and Bio-CUI_NDR, one instantly notices that Bio-CUI-NDR outperforms Bio-TB_DR. The potential factors that may contribute to the subpar performance of Bio-TB_DR are numerous. One such influence is related to the poor accuracy of concept extraction performed by BioLabeler, along with the limitations of term-based matching between concepts of topics and EMRs [54]. TREC also removed 845 EMRs from the official dataset that could have an impact on the performance of the run. As Bio-TB_DR was conducted prior to the announcement, the run included the EMRs that was dropped, opening the possibility these dropped records being determined as relevant. However, because the dropped records represent less than a percentage of the entire EMR dataset, the probability of these records significantly impacting the performance of the run in a negative way is low [54]. Bio-CUI_NDR suffers from similar negative factors as Bio-TB-DR, specifically the poor accuracy of concept extraction performed from BioLabeler. However, by switching to a CUI-based matching model from a term-based matching model, one can see a significant performance improvement. Both runs leverage all 44 UMLS sources that are available to BioLabeler along with the aforementioned procedure and disease semantic types stated in Section 4.4.1.

It is now established that the CUI-based matching model outperforms the term-based matching model. However, Figure 9 illustrates that the CUI-based matching model is still outperformed by the baseline. Additional research with the CUI-based matching

model is necessary to attempt to improve the performance of the model to a poitn it would be capable of outperforming the baseline. The next runs performed to improve the performance of the CUI-based matching model limited the number of concepts used for characterizing a given EMR and topic to the top most concept (Bio-CUI_NDR_TC) and the top five most concepts (Bio_CUI_NDR_TC5). Figure 9 and Figure 10 illustrate the Bio-CUI_NDR_TC and Bio-CUI_NDR_TC5 performance in comparison with Bio-CUI_NDR and the Baseline.

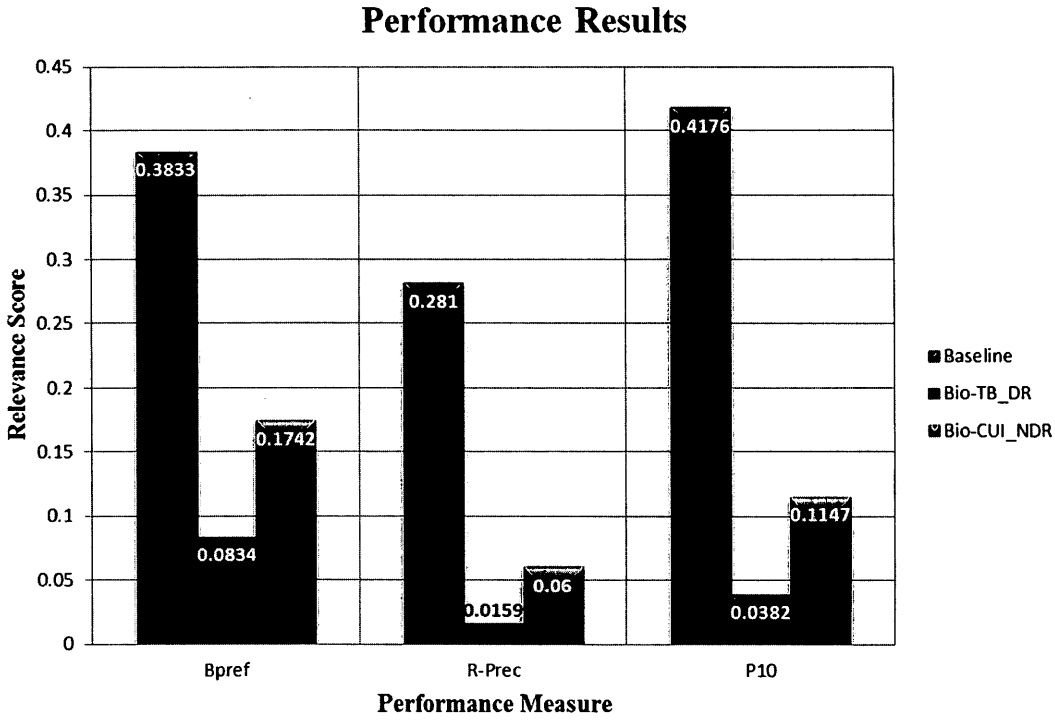


***Figure 8: Performance Comparison of the Baseline, Bio-TB_DR, and Bio-CUI_NDR Runs***

By examining Figure 9 and Figure 10, one can clearly see that by restricting the number of concepts that are used for formulating ontological impressions, a performance enhancement of the CUI-based matching model is expreinced. Figure 9 clearly demonstrates this fact by showing the R-Prec performance measure improving by 0.0252, and the P10 measure increasing by 0.0823 when limiting ontological impressions to the top most concept. Figure 10 likewise establishes that by limiting ontological impressions to the top five most concepts, there is a performance improvement of the official measures, as is clearly seen by the Bpref performance criteria increasing by 0.0267, the R-Prec performance criteria increasing by 0.0369, and the P10 performance measure increasing by 0.0853.

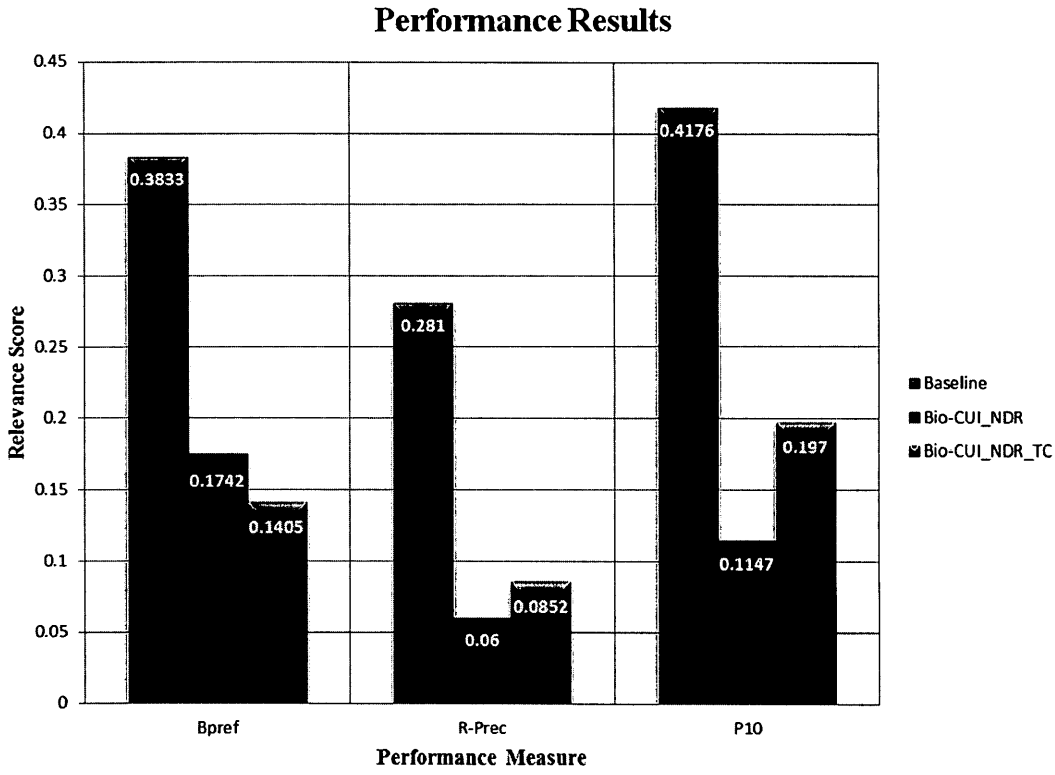


***Figure 9: Performance Comparison of the Baseline, Bio-CUI_NDR, and Bio-CUI_NDR_TC Runs***

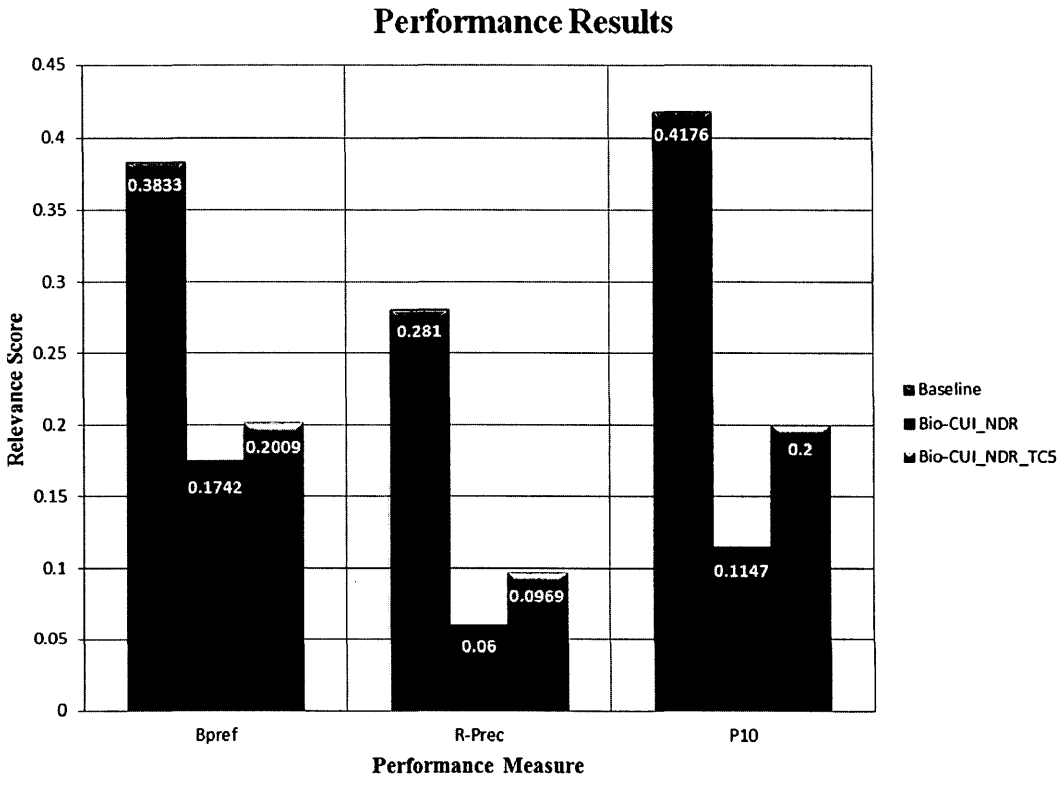


***Figure 10: Performance Comparison of the Baseline, Bio-CUI_NDR, and Bio-CUI_NDR_TC5 Runs***

Although an improvement occurs from the Bio-CUI_NDR run to the Bio-CUI_NDR_TC and Bio-CUI_NDR_TC5 runs, there remains a significant performance gap between the improved runs and the Baseline. The Baseline outperforms Bio-CUI_NDR_TC by 0.2428 for the Bpref performance measure, 0.1958 for the R-Prec performance measure, and 0.2206 for the P10 performance measure, as Figure 9 illustrates. Moreover, the Baseline outperforms Bio-CUI_NDR_TC5 by 0.1824 for the

Bpref performance measure, 0.1841 for the R-Prec performance measure, and 0.2176 for the P10 performance measure, as exemplified in Figure 10. Further attempts to improve the CUI-based matching model performance are conducted by adding an additional restriction. Instead of using all the UMLS sources to generate concepts, only the MSH UMLS source is now used to extract concepts from the EMRs and topics. These new runs continue to compare the performance difference between restricting to the top most concept and the top five concepts. As can be seen in Figure 11, there is no clear indication that by limiting the use to the top five concepts will outperform the top most concept. Although, the top five concepts does outperform the top most concept by 0.0604 for the Bpref performance measure, 0.0117 for the R-Prec performance measure, and 0.0030 for the P10 performance measure, the difference is too minor to come to concrete conclusions.

Figure 12 illustrates the performance results between the Baseline, Bio-CUI_NDR_TC, and Bio-CUI_NDR_TC_MSH runs. The Bio-CUI_NDR_TC_MSH is the run that restricts the use of UMLS sources to only the MSH UMLS source. A swift analysis of Figure 12 discloses that the run Bio-CUI_NDR_TC_MSH outperforms the run Bio-CUI_NDR_TC and it does so by 0.1123 for the Bpref performance criteria, 0.0873 for the R-Prec performance criteria and 0.0589 for the P10 performance criteria. Yet, run Bio-CUI_NDR_TC_MSH is outperformed by the Baseline in the Bpref performance measure by 0.1305, in the R-Prec performance measure by 0.1085, and in

the P10 performance measure by 0.1617. These results suggest that by restricting the UMLS sources used for generating concepts, the results will become more accurate.

| Run | Bpref | R-Prec | P10 |
|---|---|---|---|
| Baseline | 0.3833 | 0.2810 | 0.4176 |
| Bio-TB_DR | 0.0834 | 0.0159 | 0.0382 |
| Bio-CUI_NDR | 0.1742 | 0.0600 | 0.1147 |
| Bio-CUI_NDR_TC | 0.1405 | 0.0852 | 0.1970 |
| Bio-CUI_NDR_TC5 | 0.2009 | 0.0969 | 0.2000 |
| Bio-CUI_NDR_TC_MSH | 0.2528 | 0.1725 | 0.2559 |
| Bio-CUI_NDR_TC5_MSH | 0.2817 | 0.0639 | 0.2143 |
| Ter-PL2 | 0.4185 | 0.3264 | 0.4882 |
| Ter-BM25 | 0.4238 | 0.3291 | 0.4735 |
| Ter-BB2 | 0.4142 | 0.3185 | 0.4647 |
| Ter-DFI0 | 0.3918 | 0.3084 | 0.4882 |
| Ter-DRF_BM25 | 0.4241 | 0.3292 | 0.4735 |
| Ter-DFRee | 0.4050 | 0.3148 | 0.4647 |
| Ter-DirLM | 0.4136 | 0.3273 | 0.4588 |
| Ter-DHL | 0.4222 | 0.3269 | 0.4706 |
| Ter-DLH13 | 0.4149 | 0.3284 | 0.4588 |

***Table 8: Results of All Successful Runs***

The final run is given the same condition where only the MSH UMLS source was leveraged and used the top five concepts. Figure 13 elucidates the performance results between the Baseline, Bio-CUI-NDR_TC5, and Bio-CUI_NDR_TC5_MSH runs. The

Bio-CUI_NDR_TC5_MSH run restricts the use of UMLS sources to only the MSH UMLS source. Figure 13 demonstrates a similar performance improvement as seen in Figure 12. One can observe that the Bio-CUI_NDR_TC5_MSH run outperforms the Bio-CUI_NDR_TC5 run by 0.0808 for the Bpref performance measure, and by 0.0143 for the P10.

Having completed the planned runs using BioLabeler, Terrier is then used to compare how a leading traditional information retrieval methodologies fair against BioLabeler and the baseline. Figure 14 reveals the performance results between the baseline and nine traditional information retrieval methodologies implemented by Terrier. As illustrated, every traditional methodology used with Terrier resulted in their performance being superior to the baseline. Each run performed by Terrier resulted in improved performance by 0.0217 – 0.0408 for the Bpref performance measure, improved performance by 0.0274 – 0.0482 for the R-Prec performance measure, and an improved performance by 0.0412 – 0.0706 for the P10 performance measure.

It is evident that the BioLabler runs are outclassed by those performed by Terrier as is indicated by the difference in performance from leveraging the two tools. The worst-case performance measures resulted by Terrier outperforms all the best measures resulted from BioLabeler by 0.1101 for the Bpref performance measure, 0.1359 for the R-Prec performance measure, and 0.2029 for the P10 performance measure.

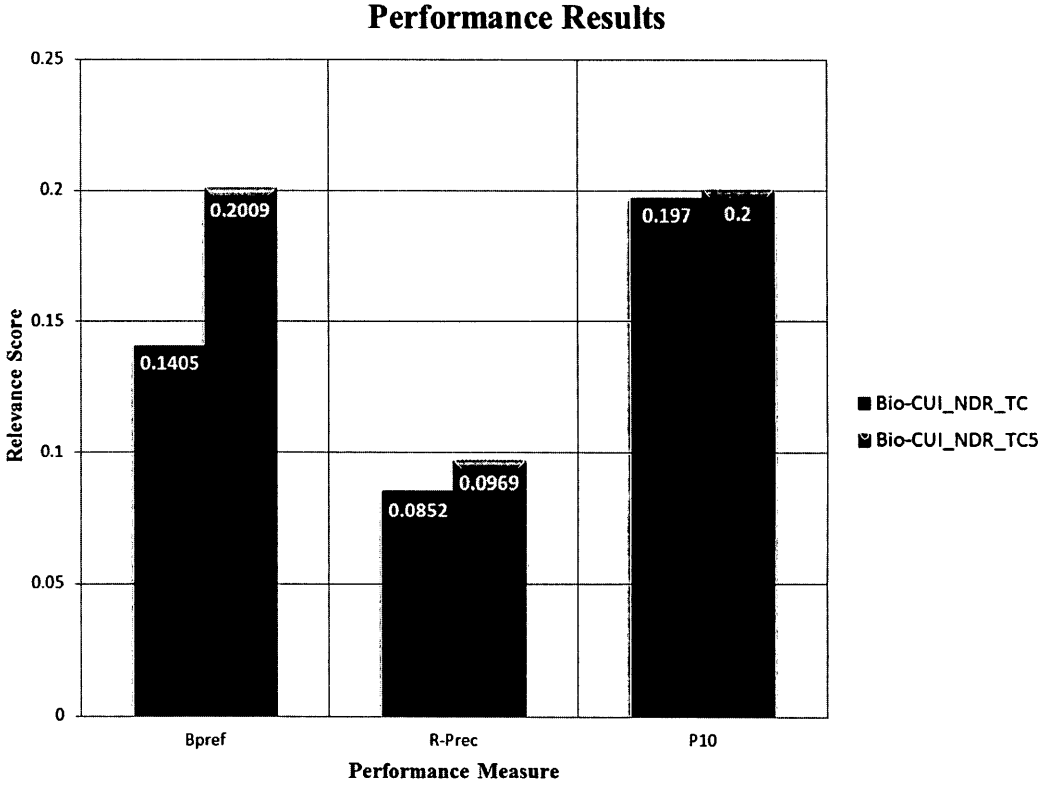


*Figure 11: Performance Comparison of Bio-CUI_NDR_TC and Bio-CUI_NDR_TC5 Runs*

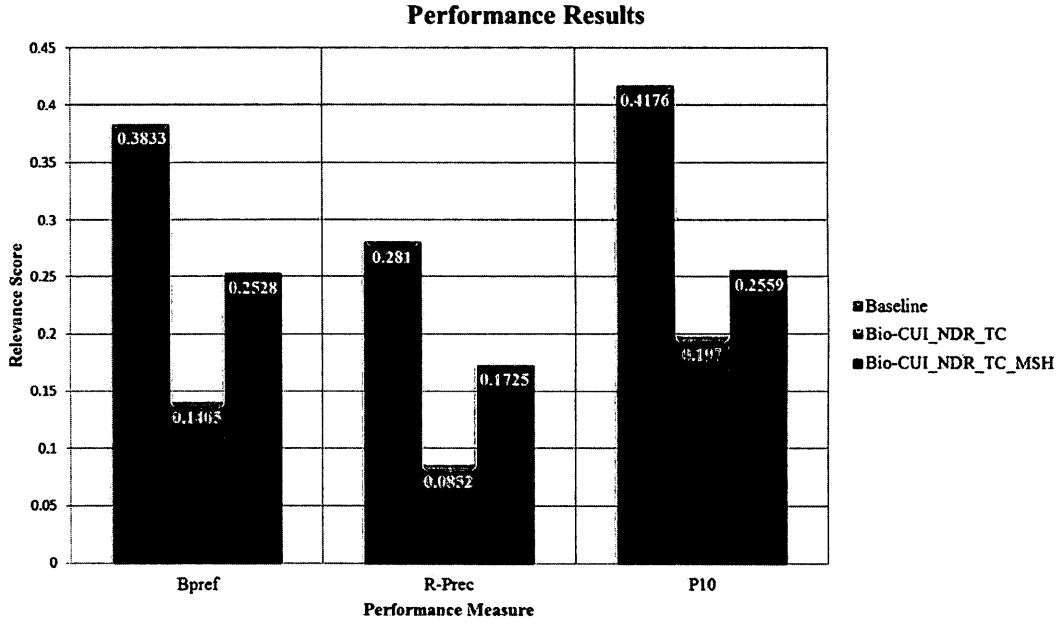


*Figure 12: Performance Comparison of the Baseline, Bio-CUI_NDR_TC, and Bio-CUI_NDR_TC_MSH Runs*

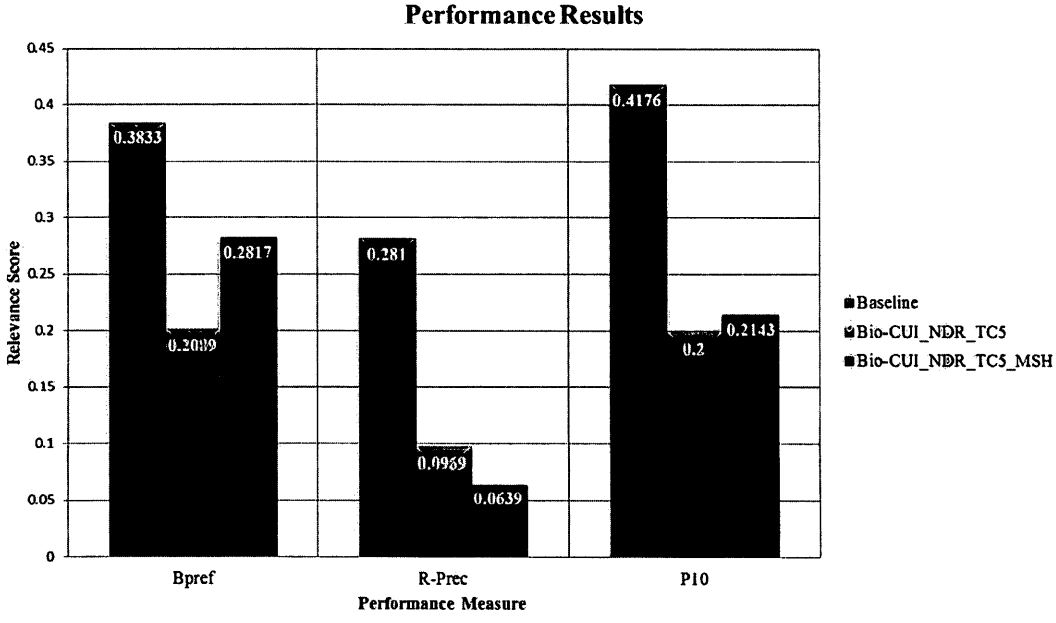


***Figure 13: Performance Comparison of the Baseline, Bio-CUI_NDR_TC5 and Bio-CUI_NDR_TC5_MSH Runs***

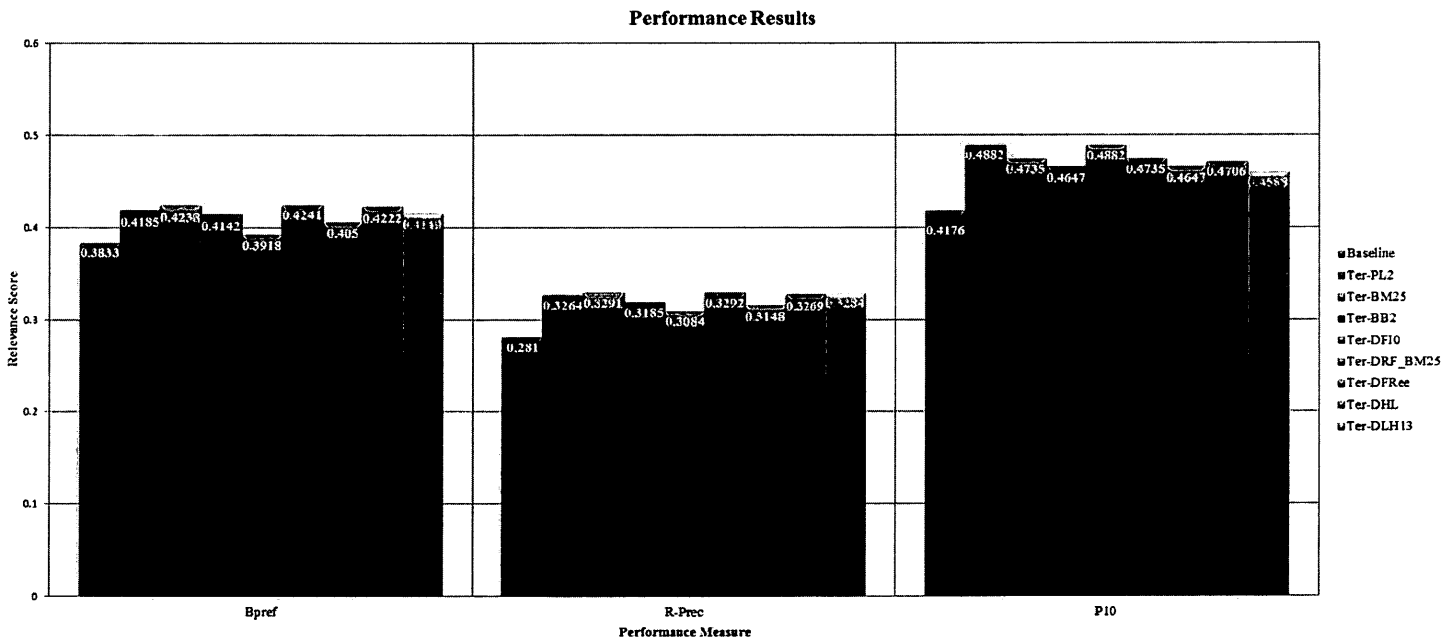


*Figure 14: Performance Comparison of the Baseline, and Various Terrier Weighting Methodologies*

# Chapter 6

# Conclusions and Future Work

## 6.1. Conclusions

Through a series of efforts, this thesis leveraged various tools and techniques in attempts to perform accurate and comprehensive biomedical concept extraction from EMRs. This thesis has leveraged numerous specially designed tools for EMR information retrieval, and presented two semantic matching models used in conjunction with those tools. In addition, this thesis utilizes traditional tools and methodologies to compare and contrast their accuracy to the specially designed tools made for EMR information retrieval.

This thesis proposes two semantic matching models to rank extracted UMLS concepts according to relevance. Existing specialized medical text mining tools are used to create UMLS concepts. The experiments showed that the proposed semantic matching

models underperform to traditional weighting algorithms such as BM25. Furthermore, when comparing the performance measures between traditional information retrieval systems to those who are specifically designed for medical text information retrieval, it is clear that the traditional system outperforms the specialized ones.

## 6.2. Theoretical Contributions

In the theoretical space, this thesis experimented with the TREC dataset using systems that are designed for medical text, along with traditional information retrieval systems. In addition, two new semantic matching models are used with specialized information retrieval systems that are designed to work with medical text. The purpose for creating new methodologies alongside these tools is to create a methodology that would accurately extract medical concepts from EMRs better than the traditional methodologies.

This thesis presents variations of experiments with the proposed semantic matching models to demonstrate that it is possible to receive higher performance over traditional methodologies. This thesis compares the results from leveraging the newly developed methodologies against traditional methodologies. The first methodology uses the weighting model BM25 opposed to the proposed semantic matching models. The second methodology leverages Terrier to perform indexing and uses numerous developed existing weighting models proven to have had good performance.

When comparing the performance measures, it is demonstrated that the use of traditional weighting models yield superior performance over the semantic models presented in this thesis. Furthermore, the use of traditional and established information retrieval systems further improve upon the performance when joined with traditional weighting models.

These results lead to two main theoretical conclusions. The first conclusion is that there is a strong need for further development of information systems that are specifically designed for medical text information retrieval, as traditional information retrieval systems are capable of outperforming the existing specialized tools. The reason for this lack of performance by existing specialized tools is because this particular field of research is new, and the systems that are in place have had a significantly shorter timeframe to be developed and refined. The second conclusion is that further research in developing a methodology for medical concept matching is necessary. Although the proposed semantic matching methodologies are show capable of having increasing performance values when properly configured, the traditional weighting models still outperform it. Similar to the reason of the first conclusion, the traditional weighting models have had more time to be developed and refined.

## 6.3. Impact to the Healthcare Industry and Information Retrieval

It is more conceivable than ever before to positively affect the healthcare industry with EMRs emerging on a global scale. They allow the discovery of new infomration within the healthcare domain that may potentially lead to new medical breakthroughs that would of previously not have been made. EMRs may contain information on diseases and treatments, along with relationships between them that may have previously not been known. This information may potentially lead to increasing the likelihood of saving an individual's life.

The work in this thesis attempts to exploit EMRs so that research can be done to the first pivotal phases in identifying medical concepts contained within the records or, as a minimum, encourage further research to enhance the ability to obtain medical information and improve the overall health of humankind.

## 6.4. Future Work

There are a number of important factors that could significantly influence the performance of medical concept retrieval. The runs conducted as part of this thesis use specific parameters and settings in conjunction with the proposed methods. Specifically, the UMLS sources that are being used with the proposed methods and the number of concepts that are being sought for are the primary parameters that play a role in concept extraction. The thesis presents a small subset of all potential possibilities chosen based

on the advice of both active medical professionals and philosophical doctors who have vested interest in the healthcare field. The performance can conceivably be improved in the future by automatically adjusting these settings and parameter values to what would yield the best performance. This is accomplished by running each possibility in parallel and selecting the run that contained the highest performance values. Selecting the most optimal settings and parameters is vital when it comes to healthcare, as any improvement in accuracy could potentially translate to an increase in curing illnesses and preventing death.

Further improvements appear to be possible in relation to the matching models presented in this thesis. As demonstrated, the traditional weighting models outperform the proposed methods. There could potentially be multiple factors that contribute to this occurrence. The first potential factor is in relation to the tools used for indexing the dataset. Future research would entail developing a tool designed specifically for medical concept extraction from biomedical text, or leveraging alternative existing tools for medical concept extraction that are of higher quality than those used in this thesis to generate the index of medical concepts for the dataset. The second potential element resides in the matching models themselves. As deliberated earlier, the traditional methods outperform the matching models presented in this thesis. Future research would necessitate advanced models that would potentially be more suitable for healthcare centered data and tailored towards the tools that are leveraged alongside them.

# Bibliography

[1] Google Inc., "Baseline - Google Search," 2012. [Online]. Available: https://www.google.ca/search?q=definition+of+baseline.

[2] S. E. Robertson, S. Walker, S. Jones, M. Hancock-Beaulieu and M. Gatford, *Proceedings of the Third Text REtrieval Conference (TREC 1994),* November 1994.

[3] Wikimedia Foundation Inc., "Concept Search," 19 May 2012. [Online]. Available: http://en.wikipedia.org/wiki/Concept_Search.

[4] B. Liu, Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data, 1st ed., M. J. Carey, S. Ceri and et al., Eds., New York: Springer, 2007, p. 6.

[5] S. P. Harter, " A probabilistic approach to automatic keyword indexing," 1974.

[6] The Daily Record, "Glossary," [Online]. Available: http://www.thedailyrecord.info/glossary.htm#D. [Accessed 18 October 2012].

[7] T. D. Gunter and N. P. Terry, "The Emergence of National Electronic Health Record Architectures in the United States and Australia: Models, Costs, and Questions," *Journal of Medical Internet Research,* vol. 7, no. 1, 14 March 2005.

[8] I. Iakovidis, "Towards Semantic Interoperability for Electronic Health Records: Domain Knowledge Governance for openEHR Archetypes," *Methods of Information in Medicine,* vol. 46, no. 3, pp. 332-343, 2007.

[9] QuinStreet Inc., "What is HTML?," [Online]. Available: http://www.webopedia.com/TERM/H/HTML.html.

[10] Centres for Disease Control and Prevention, "ICD - Classification of Diseases, Functioning, and Disability," 5 December 2011. [Online]. Available: http://www.cdc.gov/nchs/icd.htm.

[11] Apple Inc., "Glossary," 6 12 2005. [Online]. Available: http://developer.apple.com/library/mac/#documentation/userexperience/conceptual/SearchKitConcepts/searchKit_glossary/searchKit_glossary.html.

[12] json.org, "Introducing JSON," [Online]. Available: http://www.json.org/.

[13] C. D. Manning, P. Raghavan and H. Schutze, An Introduction to Information Retrieval, 1st ed., Cambridge: Cambridge University Press, 2008.

[14] U.S. National Library of Medicine, "Medical Subject Headings® - Overview," 9 September 2011. [Online]. Available: http://www.nlm.nih.gov/mesh/overview.html.

[15] E. D. Liddy, Natural Language Processing, 2nd ed., NY. Marcel Decker, Inc., 2001.

[16] QuinStreet Inc., "What is PDF?," [Online]. Available: http://www.webopedia.com/TERM/P/PDF.html.

[17] B. He and I. Ounis, "Term Frequency Normalisation Tuning for BM25 and DFR Model," in *Proceedings of the 27th European Conference on Information Retrieval (ECIR'05)*, 2005.

[18] Microsoft Co., "Glossary," 2012. [Online]. Available: http://technet.microsoft.com/en-us/library/cc966484.aspx.

[19] U.S. National Library of Medicine, "FAQs: SNOMED CT® in the UMLS®," 22 May 2012. [Online]. Available: http://www.nlm.nih.gov/research/umls/Snomed/snomed_faq.html.

[20] Wikimedia Foundation Inc., "Stemming," 20 August 2012. [Online]. Available: http://en.wikipedia.org/wiki/Stemming.

[21] QuinStreet Inc., "What is Stop Words?," [Online]. Available: http://www.webopedia.com/TERM/S/stop_words.html.

[22] Wikimedia Foundation Inc., "Structured Document," 21 September 2001. [Online]. Available: http://en.wikipedia.org/wiki/Structured_document.

[23] Wikimedia Foundation Inc., "Lexical analysis," 15 August 2012. [Online]. Available: http://en.wikipedia.org/wiki/Token_(parser)#Token.

[24] U.S. National Library of Medicine - National Institutes of Health, "UMLS Quick Start Guide," 6 December 2011. [Online]. Available: http://www.nlm.nih.gov/research/umls/quickstart.html.

[25] The Library of Congress, "WARC, Web ARChive file format," 31 August 2009. [Online]. Available: http://www.digitalpreservation.gov/formats/fdd/fdd000236.shtml.

[26] QuinStreet Inc., "What is XML?," 2012. [Online]. Available: http://www.webopedia.com/TERM/X/XML.html. [Accessed 10 May 2012].

[27] K. D. McInnes, D. Saltman and M. R. Kidd, "General practitioners' use of computers for prescribing and electronic health records: results from a national survey," *The Medical Journal of Australia,* vol. 185, no. 2, pp. 88-91, 17 July 2006.

[28] A. K. Jha, D. Doolan, D. Grandt, T. Scott and D. W. Bates, "The use of health information technology in seven nations," *International Journal of Medical*

*Informatics,* vol. 77, no. 12, pp. 848-854, December 2008.

[29] D. A. Ludwick and J. Doucette, "Adopting electronic medical records in primary care: Lessons learned from health information systems," *International Journal of Medical Informatics,* vol. 78, no. 1, pp. 22-31, 1 1 2009.

[30] Canadian Institute for Healthcare Information, Analytical Bulletin: 2004 National Physician Survey Response Rates and Comparability of Physician Demographic Distributions Physician Demographic Distributions, Ottawa, Ontario, 2005, p. 7.

[31] Canadian Institute for Healthcare Information, "Canada's Health Care Providers," Ottawa, 2001.

[32] Canadian Institute for Health Information, "Health Care in Canada," Ottawa, 2001.

[33] M. Thakkar and D. C. Davis, "Risks, Barriers, and Benefits of EHR Systems: A Comparative Study Based on Size of Hospital," *Perspectives in Health Information Management,* vol. 3, no. 5, pp. 1-19, 2006.

[34] A. Singhal, "Modern Information Retrieval: A Brief Overview," *Bulletin of the IEEE Computer Soceity Technical Committee on Data Engineering,* vol. 24, no. 4, pp. 35-43, 2001.

[35] S. E. Robertson and K. Sparck Jones, "Relevance Weighting of Search Terms,"

*Journal of the American Soceity for Information Science and Technology,* vol. 27, no. 3, pp. 129-146, May/June 1976.

[36] S. E. Robertson, S. Walker, S. Jones, M. M. Hancock-Beaulieu and M. Gatford, "Okapi at TREC-3," 1996.

[37] A. L. Rector, W. A. Nolan and S. Kay, "Foundations for an Electronic Medical Record," *Methods of Information in Medicine,* vol. 30, no. 3, pp. 179-186, 1991.

[38] W. Hersh, Information Retrieval: A Health and Biomedical Perspective, 3rd ed., K. J. Hannah and M. J. Ball, Eds., New York: Springer, 2009.

[39] S. Feldman, C. Burghard, J. Hanover and D. Schubmehl, "Unlocking the Power of Unstructured Data," *IDC Health Insights,* pp. 1-10, June 2012.

[40] C. P. Waegemann, C. Tessier, A. Barbash, B. H. Blumenfeld, J. Borden, R. M. Brinson Jr, T. Cooper, P. L. Elkin, J. M. Fitzmaurice, S. Helbig, K. M. Hunter, B. Hurley, B. Jackson, J. M. Maisel, D. Mohr, K. Rockel, J. H. Schneider, T. Sullivan and J. Weber, "Healthcare Documentation: A Report on Information Capture and Report Generation," 2002.

[41] Canada Health Infoway, "EHR: 2015 - Advancing Canada's Next Generation of Healthcare," The Globe and Mail.

[42] M. Gardner, "Information Retrieval for Patient Care," *British Medical Journal,* 29 March 1997.

[43] L. Schamber, M. Eisenberg and M. S. Nilan, "A Re-Examination of Relevance: Towards a Dynamic, Situational Definition," *Information Processing and Management: An International Journal,* vol. 26, no. 6, pp. 755-776, 1990.

[44] S. F. Altschul, W. Gish, W. Miller, E. W. Myers and D. J. Lipman, "Basic Local Alignment Search Tool," *Journal of Molecular Biology,* vol. 215, no. 3, pp. 403-410, 1990.

[45] C. J. van Rijsbergen, "Towards an Information Logic," in *Proceedings of the 12th annual international ACM SIGIR Conference on Research and Development in Information Retrieval*, New York, 1986.

[46] International Business Machines Corp., "WellPoint and IBM Announce Agreement to Put Watson to Work in Health Care," New York, 2011.

[47] D. Harris, "Machine Learning and Health Care Mean $6 Million for Predilytics," 5 September 2012. [Online]. Available: http://gigaom.com/data/machine-learning-and-health-care-mean-6m-for-predilytics/.

[48] M. Bates and R. M. Weischedel, Challenges in Natural Language Processing,

Cambridge University Press, 2006, p. 312.

[49] A. Singhal, C. Buckley and M. Mitra, "Pivoted Document Length Normalization," in *SIGIR '96 Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, New York, 1996.

[50] X. Huang, D. Sotoudeh-Hosseini, H. Rohian and X. An, "York University at TREC 2007: Genomics Track," in *The Sixteenth Text REtrieval Conference (TREC 2007) Proceedings*, 2007.

[51] S. P. Harter, "A Probabilistic Approach to Automatic Keyword Indexing Part II," *Journal of the American Society for Infomration Science,* vol. 26, no. 5, pp. 280-289, 1975.

[52] M. E. Ruiz, "Experiments on Genomics Ad Hoc Retrieval," in *Proceedings of the 14th Text Retrieval Conference*, 2005.

[53] S. E. Robertson and S. Walker, "Some Simple Effective Approximations to the 2-Position Method for Probabalistic Weighted Retrieval," in *17th Annual International ACM SIGIR Conference on Research and Development in Infomration Retrieval (SIGIR'94)*, 1994.

[54] D. Kasperowicz and J. Huang, "Semantic Matching Models for Medical Information

Retrieval: A Case Study," in *Proceedings of the 2012 Advances in Health Informatiocs Conference (AHIC 2012)*, Toronto, 2012.

[55] M. Daoud, D. Kasperowicz, J. Miao and J. Huang, "York University at TREC 2011: MEdical REords Track," in *Proceedings of the 20th TREC 2011: Medical Records Track*, 2011.

[56] H. Shatkay and R. Feldman, "Mining the Biomedical Literature in the Genomic Era: An Overview," *journal of Computional Biiology,* vol. 10, no. 6, pp. 821-855, 2003.

[57] U.S. National Library of Medicine, "UMLS 2011AA Release Available," 5 May 2011. [Online]. Available: http://www.nlm.nih.gov/pubs/techbull/mj11/mj11_umls_2011aa_release.html.

[58] K. M. Svore and C. J. C. Burges, "A Machine Learning Approach for Improved BM25 Retrieval," Redmond, 2009.

[59] S. E. Robertson and S. Walker, "Okapi/Keenbow at TREC-8," in *The Eighth Text REtrieval Conference (TREC–8)*, 2000.

[60] T. Sakai, "Alternatives to Bpref," in *Proceedings of the 30th Annual International AMC SIGIR Conference on Research and Development in Information Retrieval*, New York, 2007.

[61] C. Buckley and E. M. Voorhees, "Retrieval Evaluation with Incomplete Information," in *Proceedings of the 27th Annual International AMC SIGIR Conference on Research and Development in Information Retrieval*, 2004.

[62] M. Zhong and X. Huang, "Concept-Based Biomedical Text Retrieval," in *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2006.

[63] W. Hersh and R. T. Bhupatiraju, "TREC Genomics Track Overview," in *Proceedings of the Twelfth Text Retrieval Conference*, 2003.

[64] W. Hersh, A. M. Cphen, P. Roberts and H. K. Rekapalli, "TREC 2006 Genomics Track Overview," in *Proceedings of the 15th Text REtrieval Conference*, 2006.

[65] C. Friedman, G. Hripcsak, W. DunMouchel, S. B. Johnson and P. D. Clayton, "Natural Language Processing in an Operational Clinical Information Syste,," *Natural Language Engineering,* vol. 1, no. 1, pp. 83-108, 1995.

[66] W. Zhou, C. Yu, N. Smalheiser, V. Torvik and J. Hong, "Knowledge-Intensive Conceptual Retrieval and Passage Extraction of Biomedical Literature," in *PRoceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information retreival*, New York, 2007.

[67] M. Luís, J. L. Marina and A. Pascual-Montano, "BioLabeler and Moara in the First Round of the CALBC challenge," in *Proceedings of the First CALBC Workshop*, Hinxton, Cambridgeshire, 2010.

[68] B. Liu, Web Data Mining: Explloring Hyperlinks, Contents, and Usage Data, 1st Edition ed., New York: Springer, 2007, pp. 187-191.

[69] X. Huang, M. Zhong and L. Si, "York University at TREC 2005: Genomics Track," in *Proceedings of the Fourtheenth Text REtrieval Conference*, 2005.

[70] M. M. Beaulieu, M. Gatford, X. Huang, S. E. Robertson, S. Walker and P. Williams, "Okapi at TREC-5," in *Proceedings of the Fifth Text REtrieval Conference (TREC-5)*, 1996.

[71] J. Zhao, X. Huang and T. Hu, "A Bayesian-based Personalized Recommendation Model for Health Care," *BMC Genomics Journal,* p. 18, 21 June 2013.

[72] X. Yin, X. J. Huang, Z. Li and X. Zhou, "A Survival Modeling Approach to Biomedical Search Result Diversification Using Wikipedia," *IEEE Transactions on Knowledge and Data Engineering (TKDE),* vol. 25, no. 6, 2013.

[73] Q. Hu and X. J. Huang, "Enhancing Genomics Infomration Retrieval Through Dimensional Analysis," *Journal of Bioinfomratics and Computational Biology*

*(JBCB),* vol. 11, no. 3, p. 14, 2013.

[74] X. An and X. J. Huang, "Boosting Novelty for Biomedical Information Information Retrieval Through Probabilistic Latent Semantic Analysis," in *Proceedings of the 36th Annual International ACM SIGIR Conference on Research and Development in Infomration Retrieval (SIGIR'13)*, Dublin, Ireland, 2013.

[75] A. Babashzadeh, X. J. Huang and M. Daoud, "Exploiting Semantics for Improving Clinical Information Retrieval," in *Proceedings of the 36th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'13)*, Dublin, Ireland, 2013.

[76] Q. Hu, X. Huang and T. Hu, "Modeling and Mining Term Association for Improving Biomedical Information Retrieval Performance," *Bioinformatics Journal,* p. 18, 11 June 2012.

[77] Z. Ye, X. J. Huang and J. Miao, "A Hybrid Model for Adhoc Information Retrieval," in *Proceedings of the 35th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'12)*, Portland, Oregon, 2012.

[78] J. Miao, X. J. Huang and Z. Ye, "Proximity-based Rocchio's Model for Pseudo Relevance Feedback," in *Proceedings of the 35th Annual International ACM SIGIR*

*Conference on Research and Development in Information Retrieval (SIGIR'12)*, Portland, Oregon, 2012.

[79] J. Zhao, X. J. Huang and B. He, "CRTER: Using Cross Terms to Enhance Probabilistic Infomration Retrieval," in *Proceedings of the 34th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'11)*, Beijing, China, 2011.

[80] Y. Chen, X. Yin, Z. Li, T. Hu and X. Huang, "A LDA-based Approach to Promoting Ranking Diversity for Genomics Information Retrieval," *BMC Genomics Journal,* p. 17, 21 June 2012.

[81] X. Yin, X. Huang and Z. Li, "Mining and Modeling Linkage Information from Citation Context for Improving Biomedical Literature Retrieval," *Infomration Processing & Management: An International Journal (IPM),* vol. 47, no. 1, pp. 53-67, 2011.

[82] Q. Hu, X. Huang and J. Miao, "A Robust Approach to Optimixing Multi-Source Information for Enhancing Genomics Retrieval Performance," *BMC Bioinformatics Journal,* p. 18, 27 July 2011.

[83] X. Yin, Z. Li, X. Huang and X. Hu, "Promoting Ranking Diversity for Genomics Search with a Relevance-Novelty Combined Model," *BMC Bioinformatics Journal,*

p. 16, 27 July 2011.

[84] Q. Hu and X. Huang, "Passage Extraction and Result Combination for Genomics Information Retrieval," *Journal of Intelligent Information Systems (JIIS),* vol. 34, no. 3, pp. 249-274, 2010.

[85] X. Huang, A. An and Q. Hu, "Medical Search and Classification Tools for Recommendation," in *Proceedings of the 33rd Annual International Conference on Research and Development in Informational Retrieval*, Geneva, Sqitzerland, 2010.

[86] Q. Hu and X. Huang, "Genomics Information Retrieval Using a Bayesian Model for Learning and Re-ranking," in *Proceedings of the 2010 ACM International Conference on Bioinformatics and Computational Biology (BCB)*, New York, USA, 2010.

[87] X. Yin and X. Huang, "Promoting Ranking Diversity for Biomedical Information Retrieval Using Wikipedia," in *Proceedings of the 32nd European Conference on Information Retrieval (ECIR2010)*, Milton Keynes, UK, 2010.

[88] X. Yin, X. Huang and Z. Li, "Towards a Better Ranking for Biomedical Information Retrieval Using Context," in *Proceedings of the 2009 IEEE International Conference on Bioinformatics & Biomedicine*, Washington D.C., USA, 2009.

[89] X. Yin, X. Huang and Z. Li, "BioCLink: A Probabilistic Approach for Improving Genomics Search with Citation Links," in *Proceedings of the 2009 IEEE International Conference on Bioinformatics & Biomedicine*, Washington D.C., USA, 2009.

[90] X. Yin, H. Xiangji, Q. Hu and Z. Li, "Boosting Biomedical Information Retrieval Performance through Citation Graph: An Empirical Study," in *PAKDD*, 2009.

[91] X. Huang and Q. Hu, "A Bayesian Learning Approach to Promoting Diversity in Ranking for Biomedical Information Retrieval," in *Proceedings of the 32nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Boston, USA, 2009.

# Appendix A

# TREC Topics

This thesis used the topics given by TREC 2011 Medical Records track. Below is the exhaustive list of all topics. TREC dropped topic number 130 from the official golden standard, and as such was not used in this study.

| | |
|---|---|
| 101 | Patients with hearing loss |
| 102 | Patients with complicated GERD who receive endoscopy |
| 103 | Hospitalized patients treated for methicillin-resistant Staphylococcus aureus (MRSA) endocarditis |
| 104 | Patients diagnosed with localized prostate cancer and treated with robotic surgery |
| 105 | Patients with dementia |
| 106 | Patients who had positron emission tomography (PET), magnetic resonance imaging (MRI), or computed tomography (CT) for staging or monitoring of cancer |
| 107 | Patients with ductal carcinoma in situ (DCIS) |
| 108 | Patients treated for vascular claudication surgically |
| 109 | Women with osteopenia |

110 Patients being discharged from the hospital on hemodialysis

111 Patients with chronic back pain who receive an intraspinal pain-medicine pump

112 Female patients with breast cancer with mastectomies during admission

113 Adult patients who received colonoscopies during admission which revealed adenocarcinoma

114 Adult patients discharged home with palliative care / home hospice

115 Adult patients who are admitted with an asthma exacerbation

116 Patients who received methotrexate for cancer treatment while in the hospital

117 Patients with Post-traumatic Stress Disorder

118 Adults who received a coronary stent during an admission

119 Adult patients who presented to the emergency room with with anion gap acidosis secondary to insulin dependent diabetes

120 Patients admitted for treatment of CHF exacerbation

121 Patients with CAD who presented to the Emergency Department with Acute Coronary Syndrome and were given Plavix

122 Patients who received total parenteral nutrition while in the hospital

123 Diabetic patients who received diabetic education in the hospital

124 Patients who present to the hospital with episodes of acute loss of vision secondary to glaucoma

125 Patients co-infected with Hepatitis C and HIV

126 Patients admitted with a diagnosis of multiple sclerosis

127 Patients admitted with morbid obesity and secondary diseases of diabetes and or hypertension

128 Patients admitted for hip or knee surgery who were treated with anti-coagulant medications post-op

129 Patients admitted with chest pain and assessed with CT angiography

130 Children admitted with cerebral palsy who received physical therapy

131 Patients who underwent minimally invasive abdominal surgery

132 Patients admitted for surgery of the cervical spine for fusion or discectomy

133 Patients admitted for care who take herbal products for osteoarthritis

134 Patients admitted with chronic seizure disorder to control seizure activity

135 Cancer patients with liver metastasis treated in the hospital who underwent a procedure

# Appendix B

# MySQL Tables

MySQL tables were leveraged in parts of this thesis in order to store some information that was needed to conduct the research contained within. Below are the SQL needed to create all the tables in a MySQL database.

## B.1 biolabeler_medlineplus_procedureanddisease Table

```
delimiter $$
CREATE TABLE `biolabeler_medlineplus_procedureanddisease`
(
  `BioLabelerID` int(10) unsigned NOT NULL AUTO_INCREMENT,
  `recordID` int(10) unsigned NOT NULL,
  `cui` varchar(255) NOT NULL,
  `normalizedWeight` varchar(255) NOT NULL,
  PRIMARY KEY (`BioLabelerID`)
) ENGINE=InnoDB AUTO_INCREMENT=5552669 DEFAULT CHARSET=latin1$$
```

## B.2 biolabeler_medlineplus_procedureanddisease_topics Table

```
delimiter $$
CREATE TABLE `biolabeler_medlineplus_procedureanddisease_topics`
(
 `BioLabelerID` int(10) unsigned NOT NULL AUTO_INCREMENT,
 `topicID` int(10) unsigned NOT NULL,
 `cui` varchar(255) NOT NULL,
 `normalizedWeight` varchar(255) NOT NULL,
 PRIMARY KEY (`BioLabelerID`)
) ENGINE=InnoDB AUTO_INCREMENT=817 DEFAULT CHARSET=latin1$$
```

## B.3 biolabeler_msh_procedureanddisease Table

```
delimiter $$
CREATE TABLE `biolabeler_medlineplus_procedureanddisease_topics`
 (
 `BioLabelerID` int(10) unsigned NOT NULL AUTO_INCREMENT,
 `topicID` int(10) unsigned NOT NULL,
 `cui` varchar(255) NOT NULL,
 `normalizedWeight` varchar(255) NOT NULL,
 PRIMARY KEY (`BioLabelerID`)
) ENGINE=InnoDB AUTO_INCREMENT=817 DEFAULT CHARSET=latin1$$
```

## B.4 biolabeler_msh_procedureanddisease_topics Table

```
delimiter $$
CREATE TABLE `biolabeler_msh_procedureanddisease_topics`
(
 `BioLabelerID` int(10) unsigned NOT NULL AUTO_INCREMENT,
 `topicID` int(10) unsigned NOT NULL,
 `cui` varchar(255) NOT NULL,
 `normalizedWeight` varchar(255) NOT NULL,
 PRIMARY KEY (`BioLabelerID`)
) ENGINE=InnoDB AUTO_INCREMENT=3142 DEFAULT CHARSET=latin1$$
```

# Appendix C

# Programming Code

The following code was used as part of the experimentation process throughout this thesis. Only the main files are included, and minor code changes are required in order to receive the data needed for each run. In addition, minute changes to the code have been made to protect sensitive information, such as usernames and passwords.

## C.1 BioLabeler Code

### *C.1.1 MSH_ProcedureAndDisease_Records*

```
package bioLabeler;

import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.io.OutputStreamWriter;
import java.net.URL;
import java.net.URLConnection;
import java.net.URLEncoder;
import java.sql.Connection;
```

```
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import org.json.JSONArray;
import org.json.JSONObject;
import com.mysql.jdbc.Statement;
import ca.dawidk.inputs.DateAndTime;
import ca.dawidk.sql.MySQL;

public class MSH_ProcedureAndDisease_Records
{
 public static void main(String[] args)
 {
  String bioLabelerAddress =
"http://www.biolabeler.com/bioLabeler/bioLabelerMain/save";
  String parameters[][] = new String[14][2];
  String jsonCode = null;
  Connection connection = MySQL.connectToDatabase("USERNAME", "PASSWORD",
"jdbc:mysql://DATABASE LOCATION");
  ResultSet resultSet = getSelectResultsFromDatabase(connection, "SELECT
recordNumber, report_text FROM records;");

  try
  {
   while(resultSet.next())
   {
    parameters[0][0] = "MSH";          parameters[0][1] = "on";
    parameters[1][0] = "T019";         parameters[1][1] = "on";
    parameters[2][0] = "T020";         parameters[2][1] = "on";
    parameters[3][0] = "T047";         parameters[3][1] = "on";
    parameters[4][0] = "T048";         parameters[4][1] = "on";
    parameters[5][0] = "T050";         parameters[5][1] = "on";
    parameters[6][0] = "T058";         parameters[6][1] = "on";
    parameters[7][0] = "T059";         parameters[7][1] = "on";
    parameters[8][0] = "T060";         parameters[8][1] = "on";
    parameters[9][0] = "T061";         parameters[9][1] = "on";
    parameters[10][0] = "T191";        parameters[10][1] = "on";
    parameters[11][0] = "abstractText"; parameters[11][1] =
resultSet.getString("report_text");
    parameters[12][0] = "eMail";       parameters[12][1] = "dawidk@yorku.ca";
    parameters[13][0] = "maxConcepts"; parameters[13][1] = "1000";
```

```
   jsonCode = sendPostRequest(bioLabelerAddress, parameters);

   if(!jsonCode.equals("false"))
   {
    JSONObject json = new JSONObject(jsonCode.toString());
    JSONArray jsonA = new JSONArray(json.get("concepts").toString());

    for(int i = 0; i < jsonA.length(); i++)
    {
     JSONObject jsonObject = jsonA.getJSONObject(i);
     PreparedStatement insertStatement = connection.prepareStatement((new
StringBuilder("INSERT INTO biolabeler_msh_procedureanddisease (recordID, cui,
normalizedWeight) VALUES
(")).append(resultSet.getString("recordNumber")).append(",
\"").append(jsonObject.get("cui")).append("\",
\"").append(jsonObject.get("normWeight")).append("\")").toString());
      insertStatement.executeUpdate();
      insertStatement.close();
    }
   }
  }

  resultSet.close();
  connection.commit();
  connection.close();
 }
 catch(Exception e)
 {
  e.printStackTrace();
  e.getMessage();
  e.getStackTrace();
  System.exit(0);
 }
}

 private static ResultSet getSelectResultsFromDatabase(Connection connection, String
selectQuery)
 {
  ResultSet resultSet = null;

  try
```

```
{
 Statement statement = (Statement)connection.createStatement();
 statement.executeQuery(selectQuery);
 resultSet = statement.getResultSet();
}
catch(Exception e)
{
 e.printStackTrace();
 e.getMessage();
 e.getStackTrace();
 System.exit(0);
}

return resultSet;
}

public static String sendPostRequest(String url, String parameters[][])
{
 String data = "";
 StringBuffer answer = new StringBuffer();

 for(int i = 0; i < parameters.length; i++)
 {
  try
  {
   data = (new
StringBuilder(String.valueOf(data))).append(parameters[i][0]).append("=").append(URL
Encoder.encode(parameters[i][1], "UTF-8")).append("&").toString();
  }
  catch(Exception e)
  {
   e.printStackTrace();
  }
 }

 data = data.substring(0, data.length() - 1);

 try
 {
  URL address = new URL(url);
  URLConnection conn = address.openConnection();
```

```
  conn.setDoOutput(true);
  OutputStreamWriter writer = new OutputStreamWriter(conn.getOutputStream());
  writer.write(data);
  writer.flush();
  BufferedReader reader = new BufferedReader(new
InputStreamReader(conn.getInputStream()));
  String line;

  while((line = reader.readLine()) != null)
  {
   answer.append(line);
  }

  writer.close();
  reader.close();
 }
 catch(Exception e)
 {
  e.printStackTrace();
  answer.append("false");
 }

 return answer.toString();
 }
}
```

### *C.1.2 MSH_ProcedureAndDisease_Topics*

```
package bioLabeler;

import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.io.OutputStreamWriter;
import java.net.URL;
import java.net.URLConnection;
import java.net.URLEncoder;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
```

```
import java.util.Date;
import org.json.JSONArray;
import org.json.JSONObject;
import ca.dawidk.inputs.DateAndTime;
import ca.dawidk.sql.MySQL;
import com.mysql.jdbc.Statement;

public class MSH_ProcedureAndDisease_Topics
{
 public static void main(String[] args)
 {
  String bioLabelerAddress =
"http://www.biolabeler.com/bioLabeler/bioLabelerMain/save";
  String parameters[][] = new String[14][2];
  String jsonCode = null;
  Connection connection = MySQL.connectToDatabase("USERNAME", "PASSWORD",
"jdbc:mysql://DATABASE LOCATION");
  ResultSet resultSet = getSelectResultsFromDatabase(connection, "SELECT topicID,
text FROM topics;");

  try
  {
   while(resultSet.next())
   {
    parameters[0][0] = "MSH";          parameters[0][1] = "on";
    parameters[1][0] = "T019";         parameters[1][1] = "on";
    parameters[2][0] = "T020";         parameters[2][1] = "on";
    parameters[3][0] = "T047";         parameters[3][1] = "on";
    parameters[4][0] = "T048";         parameters[4][1] = "on";
    parameters[5][0] = "T050";         parameters[5][1] = "on";
    parameters[6][0] = "T058";         parameters[6][1] = "on";
    parameters[7][0] = "T059";         parameters[7][1] = "on";
    parameters[8][0] = "T060";         parameters[8][1] = "on";
    parameters[9][0] = "T061";         parameters[9][1] = "on";
    parameters[10][0] = "T191";        parameters[10][1] = "on";
    parameters[11][0] = "abstractText"; parameters[11][1] = resultSet.getString("text");
    parameters[12][0] = "eMail";       parameters[12][1] = "dawidk@yorku.ca";
    parameters[13][0] = "maxConcepts"; parameters[13][1] = "100";

    jsonCode = sendPostRequest(bioLabelerAddress, parameters);
```

```
   if(!jsonCode.equals("false"))
   {
    JSONObject json = new JSONObject(jsonCode.toString());
    JSONArray jsonA = new JSONArray(json.get("concepts").toString());

    for(int i = 0; i < jsonA.length(); i++)
    {
     JSONObject jsonObject = jsonA.getJSONObject(i);
     PreparedStatement insertStatement = connection.prepareStatement((new
StringBuilder("INSERT INTO BioLabeler_MSH_ProcedureAndDisease_Topics
(topicID, cui, normalizedWeight) VALUES
(")).append(resultSet.getString("topicID")).append(",
\"").append(jsonObject.get("cui")).append("\",
\"").append(jsonObject.get("normWeight")).append("\")").toString());
     insertStatement.executeUpdate();
     insertStatement.close();
    }
   }
  }

  resultSet.close();
  connection.commit();
  connection.close();
 }
 catch(Exception e)
 {
  e.printStackTrace();
  e.getMessage();
  e.getStackTrace();
  System.exit(0);
 }
}

 private static ResultSet getSelectResultsFromDatabase(Connection connection, String
selectQuery)
 {
  ResultSet resultSet = null;

  try
  {
   Statement statement = (Statement)connection.createStatement();
```

```
 statement.executeQuery(selectQuery);
 resultSet = statement.getResultSet();
 }
 catch(Exception e)
 {
 e.printStackTrace();
 e.getMessage();
 e.getStackTrace();
 System.exit(0);
 }

 return resultSet;
}

 public static String sendPostRequest(String url, String parameters[][])
 {
 String data = "";
 StringBuffer answer = new StringBuffer();

 for(int i = 0; i < parameters.length; i++)
 {
  try
  {
   data = (new
StringBuilder(String.valueOf(data))).append(parameters[i][0]).append("=").append(URL
Encoder.encode(parameters[i][1], "UTF-8")).append("&").toString();
  }
  catch(Exception e)
  {
   e.printStackTrace();
   System.err.println(data);
  }
 }

 data = data.substring(0, data.length() - 1);

 try
 {
 URL address = new URL(url);
 URLConnection conn = address.openConnection();
 conn.setDoOutput(true);
```

```
  OutputStreamWriter writer = new OutputStreamWriter(conn.getOutputStream());
  writer.write(data);
  writer.flush();
  BufferedReader reader = new BufferedReader(new
InputStreamReader(conn.getInputStream()));
  String line;

  while((line = reader.readLine()) != null)
  {
   answer.append(line);
  }

  writer.close();
  reader.close();
 }
 catch(Exception ex)
 {
  ex.printStackTrace();
  answer.append("false");
 }

 return answer.toString();
 }
}
```

## C.2 OpenCalais Code

### *C.2.1 XMLParser_AddToDatabase.java*

```
package openCalais;

import java.io.File;
import java.io.IOException;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.SQLException;
import java.util.List;
import java.util.regex.Matcher;
```

```
import java.util.regex.Pattern;
import org.jdom.Document;
import org.jdom.Element;
import org.jdom.JDOMException;
import org.jdom.input.SAXBuilder;
import ca.dawidk.sql.MySQL;

public class XMLParser_AddToDatabase
{
 public static void main(String args[])
 {
  SAXBuilder saxBuilder = new SAXBuilder();
  File directory = new File("OUTPUT DATA FILE PATH");
  File[] xmlFiles = directory.listFiles();
  Connection connection = MySQL.connectToDatabase("USERNAME", "PASSWORD",
"jdbc:mysql://DATABASE LOCATION");

  for(int a = 0 ; a < xmlFiles.length ; a++)
  {
   try
   {
    Pattern pattern = Pattern.compile("\\d+");
    Matcher matcher = pattern.matcher(xmlFiles[a].getName());

    if(matcher.find())
    {
     Document document = (Document) saxBuilder.build(xmlFiles[a]);
     Element rootNode = document.getRootElement();
     List list = rootNode.getChildren("CalaisSimpleOutputFormat");

     for(int i = 0 ; i < list.size() ; i++)
     {
      Element node = (Element) list.get(i);
      List medicalTreatmentChildren = node.getChildren("MedicalTreatment");
      List medicalConditionChildren = node.getChildren("MedicalCondition");

      for(int j = 0 ; j < medicalTreatmentChildren.size() ; j++)
      {
       Element childrenNode = (Element) medicalTreatmentChildren.get(j);
       PreparedStatement insertStatement = connection.prepareStatement("INSERT INTO
openCalais_conditionsAndTreatments (recordNumber, isConditionOrTreatment,
```

```
conditionAndTreatmentName, relevance) VALUES (" + matcher.group() + ", \"T\", \"" +
childrenNode.getText().replace("\n", "").replace("\"", "").replace("  ", " ").trim() + "\", " +
childrenNode.getAttributeValue("relevance").trim() + ");");
     insertStatement.executeUpdate();
     insertStatement.close();
    }

    for(int j = 0 ; j < medicalConditionChildren.size() ; j++)
    {
     Element childrenNode = (Element) medicalConditionChildren.get(j);
     PreparedStatement insertStatement = connection.prepareStatement("INSERT INTO
openCalais_conditionsAndTreatments (recordNumber, isConditionOrTreatment,
conditionAndTreatmentName, relevance) VALUES (" + matcher.group() + ", \"C\", \"" +
childrenNode.getText().replace("\n", "").replace("\"", "").replace("  ", " ").trim() + "\", " +
childrenNode.getAttributeValue("relevance").trim() + ");");
     insertStatement.executeUpdate();
     insertStatement.close();
    }
   }
  }
  catch(IOException e)
  {
   System.err.println(e.getMessage());
   e.printStackTrace();

   try
   {
    connection.rollback();
    connection.close();
   }
   catch(SQLException f)
   {
    System.err.println(f.getMessage());
    f.printStackTrace();
    System.exit(0);
   }

   System.exit(0);
  }
  catch(JDOMException e)
```

```
{
 System.err.println(e.getMessage());
 e.printStackTrace();

 try
 {
  connection.rollback();
  connection.close();
 }
 catch(SQLException f)
 {
  System.err.println(f.getMessage());
  f.printStackTrace();
  System.exit(0);
 }

 System.exit(0);
}
catch (SQLException e)
{
 System.err.println(e.getMessage());
 e.printStackTrace();

 try
 {
  connection.rollback();
  connection.close();
 }
 catch(SQLException f)
 {
  System.err.println(f.getMessage());
  f.printStackTrace();
  System.exit(0);
 }

 System.exit(0);
}
}

try
{
```

```
 connection.commit();
 connection.close();
 }
 catch(SQLException e)
 {
  System.err.println(e.getMessage());
  e.printStackTrace();
  System.exit(0);
 }
}
}
```

### *C.2.2 HTTPClientPost.java*

```
package openCalais;

import org.apache.commons.httpclient.HttpClient;
import org.apache.commons.httpclient.HttpStatus;
import org.apache.commons.httpclient.methods.FileRequestEntity;
import org.apache.commons.httpclient.methods.PostMethod;
import java.io.*;

public class HttpClientPost
{
 private static final String CALAIS_URL = "http://api.opencalais.com/tag/rs/enrich";
 private File input;
 private File output;
 private HttpClient client;

 public static void main(String[] args)
 {
  HttpClientPost httpClientPost = new HttpClientPost();
  httpClientPost.input = new File("TOPIC FILE LOCATION");
  httpClientPost.output = new File("OUTPUT LOCATION");
  httpClientPost.client = new HttpClient();
  httpClientPost.client.getParams().setParameter("http.useragent", "Calais Rest Client");
  httpClientPost.run();
 }
```

```
private PostMethod createPostMethod()
{
 PostMethod method = new PostMethod(CALAIS_URL);
 method.setRequestHeader("x-calais-licenseID", "zfzr7kfd6gft26sfn4evhb3n");
 method.setRequestHeader("Content-Type", "text/raw; charset=UTF-8");
 method.setRequestHeader("Accept", "Text/Simple");

 return method;
}

private void run()
{
 try
 {
  if(input.isFile())
  {
   postFile(input, createPostMethod());
  }
  else if(input.isDirectory())
  {
   for (File file : input.listFiles())
   {
    if (file.isFile())
    {
     postFile(file, createPostMethod());
    }
   }
  }
 }
 catch (Exception e)
 {
  e.printStackTrace();
 }
}

private void doRequest(File file, PostMethod method)
{
 try
 {
  int returnCode = client.executeMethod(method);
```

```
 if (returnCode == HttpStatus.SC_NOT_IMPLEMENTED)
 {
  System.err.println("The Post method is not implemented by this URI");
  method.getResponseBodyAsString();
 }
 else if (returnCode == HttpStatus.SC_OK)
 {
  saveResponse(file, method);
 }
 else
 {
  System.err.println("File post failed: " + file);
  System.err.println("Got code: " + returnCode);
  System.err.println("response: "+method.getResponseBodyAsString());
 }
}
catch (Exception e)
{
 e.printStackTrace();
}
finally
{
 method.releaseConnection();
}
}

private void saveResponse(File file, PostMethod method) throws IOException
{
 PrintWriter writer = null;

 try
 {
  BufferedReader reader = new BufferedReader(new
InputStreamReader(method.getResponseBodyAsStream(), "UTF-8"));
  File out = new File(output, file.getName() + ".xml");
  writer = new PrintWriter(new BufferedWriter(new FileWriter(out)));
  String line;

  while ((line = reader.readLine()) != null)
  {
   writer.println(line);
```

```
 }
 }
 catch (IOException e)
 {
  e.printStackTrace();
 }
 finally
 {
  if (writer != null) try {writer.close();} catch (Exception ignored) {}
 }
}

private void postFile(File file, PostMethod method) throws IOException
{
 method.setRequestEntity(new FileRequestEntity(file, null));
 doRequest(file, method);
}
}
```

## C.3 MetaMap Code

### *C.3.1 GenerateTopicConcepts.java*

```
package metamap;

import gov.nih.nlm.nls.metamap.Ev;
import gov.nih.nlm.nls.metamap.MetaMapApi;
import gov.nih.nlm.nls.metamap.MetaMapApiImpl;
import gov.nih.nlm.nls.metamap.PCM;
import gov.nih.nlm.nls.metamap.Result;
import gov.nih.nlm.nls.metamap.Utterance;
import java.io.BufferedReader;
import java.io.File;
import java.io.FileReader;
import java.io.IOException;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.SQLException;
```

```
import java.util.LinkedList;
import java.util.List;
import java.util.ArrayList;
import ca.dawidk.inputs.DateAndTime;
import ca.dawidk.sql.MySQL;

public class GenerateTopicConcepts
{
 public static void main(String[] args)
 {
  String line = null;
  Connection connection = MySQL.connectToDatabase("USERNAME", "PASSWORD",
"jdbc:mysql://DATABASE LOCATION");

  try
  {
   BufferedReader bufferedReader = new BufferedReader(new FileReader(new
File("LOCATION OF TOPIC FILES")));

   while((line = bufferedReader.readLine()) != null)
   {
    String[] tokens = line.split("~");
    LinkedList<String> conceptsWithScore = getMetaMapConcepts(tokens[1]);
    String[] conceptsAndScores = new String[conceptsWithScore.size()];
    conceptsWithScore.toArray(conceptsAndScores);

    for(int i = 0 ; i < conceptsAndScores.length ; i++)
    {
     String[] tokens2 = conceptsAndScores[i].split("~");
     PreparedStatement insertStatement = connection.prepareStatement("INSERT INTO
metamap_topics (topicID, cui, score) VALUES (" + tokens[0] + ", \"" + tokens2[1] + "\",
" + tokens2[0] + ");");
     insertStatement.executeUpdate();
     insertStatement.close();
    }
   }

   connection.commit();
   connection.close();
  }
  catch(IOException e)
```

```
{
 e.printStackTrace();
}
catch(SQLException e)
{
 e.printStackTrace();
}
}

private static LinkedList<String> getMetaMapConcepts(String terms)
{
 LinkedList<String> conceptsWithScore = new LinkedList<String>();

 try
 {
  MetaMapApi api = new MetaMapApiImpl();
  List<String> theOptions = new ArrayList<String>();
  theOptions.add("-y");  // turn on Word Sense Disambiguation

  if(theOptions.size() > 0)
  {
   api.setOptions(theOptions);
  }

  List<Result> resultList = api.processCitationsFromString(terms);
  Result result = resultList.get(0);

  for(Utterance utterance: result.getUtteranceList())
  {
   for(PCM pcm: utterance.getPCMList())
   {
    for(Ev ev: pcm.getCandidateList())
    {
     String insert = ev.getScore() + "~" + ev.getConceptId();
     conceptsWithScore.add(insert.substring(1));
    }
   }
  }
 }
 catch(Exception e)
 {
```

```
 e.printStackTrace();
 }

 return conceptsWithScore;
}
}
```

### *C.3.2 GenerateRecordConcepts.java*

```
package metamap;

import gov.nih.nlm.nls.metamap.Ev;
import gov.nih.nlm.nls.metamap.MetaMapApi;
import gov.nih.nlm.nls.metamap.MetaMapApiImpl;
import gov.nih.nlm.nls.metamap.PCM;
import gov.nih.nlm.nls.metamap.Result;
import gov.nih.nlm.nls.metamap.Utterance;
import java.io.IOException;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.LinkedList;
import java.util.List;
import com.mysql.jdbc.Statement;
import ca.dawidk.inputs.DateAndTime;
import ca.dawidk.sql.MySQL;

public class GenerateRecordConcepts
{
 public static void main(String[] args)
 {
  Connection connection = MySQL.connectToDatabase("USERNAME", "PASSWORD",
"jdbc:mysql://DATABASE LOCATION");
  ResultSet resultSet = getSelectResultsFromDatabase(connection, "SELECT
recordNumber, report_text FROM records WHERE recordNumber = 10");

  try
```

```
 {
  while(resultSet.next())
  {
   LinkedList<String> conceptsWithScore =
getMetaMapConcepts(resultSet.getString("report_text").replace("\n\r", " "));
   String[] conceptsAndScores = new String[conceptsWithScore.size()];
   conceptsWithScore.toArray(conceptsAndScores);

   for(int i = 0 ; i < conceptsAndScores.length ; i++)
   {
    String[] tokens = conceptsAndScores[i].split("~");
    PreparedStatement insertStatement = connection.prepareStatement("INSERT INTO
metamap_records (recordID, cui, score) VALUES (" +
resultSet.getString("recordNumber") + ", \"" + tokens[1] + "\", " + tokens[0] + ");");
    insertStatement.executeUpdate();
    insertStatement.close();
   }

   connection.commit();
  }

  resultSet.close();
  connection.close();
 }
 catch(SQLException e)
 {
  e.printStackTrace();
 }
 catch(InterruptedException e)
 {
  e.printStackTrace();
  e.getMessage();
  e.getStackTrace();
 }
 catch(IOException e)
 {
  e.printStackTrace();
 }
}
```

```
 private static LinkedList<String> getMetaMapConcepts(String terms) throws
InterruptedException, IOException
 {
 LinkedList<String> conceptsWithScore = new LinkedList<String>();

 try
 {
 MetaMapApi api = new MetaMapApiImpl();
 List<String> theOptions = new ArrayList<String>();
 theOptions.add("-y");  // turn on Word Sense Disambiguation

 if(theOptions.size() > 0)
 {
  api.setOptions(theOptions);
 }

 List<Result> resultList = api.processCitationsFromString(terms);
 Result result = resultList.get(0);

 for(Utterance utterance: result.getUtteranceList())
 {
  for(PCM pcm: utterance.getPCMList())
  {
   for(Ev ev: pcm.getCandidateList())
   {
    String insert = ev.getScore() + "~" + ev.getConceptId();
    conceptsWithScore.add(insert.substring(1));
   }
  }
 }

 api.disconnect();
}
catch(Exception e)
{
 e.printStackTrace();

 Runtime.getRuntime().exec("cmd.exe /k start METAMAP FILE
PATH/bin/mmserver11v2.bat");
 Thread.sleep(10000);
}
```

```
 return conceptsWithScore;
}

 private static ResultSet getSelectResultsFromDatabase(Connection connection, String
selectQuery)
 {
 ResultSet resultSet = null;

 try
 {
  Statement statement = (Statement)connection.createStatement();
  statement.executeQuery(selectQuery);
  resultSet = statement.getResultSet();
 }
 catch(Exception e)
 {
  e.printStackTrace();
  System.exit(0);
 }

 return resultSet;
 }
}
```