5-2010

CREATING A BIOMEDICAL ONTOLOGY INDEXED SEARCH ENGINE TO IMPROVE THE SEMANTIC RELEVANCE OF RETREIVED MEDICAL TEXT

William Taylor II
Clemson University, wptaylo@clemson.edu

CREATING A BIOMEDICAL ONTOLOGY INDEXED SEARCH ENGINE TO IMPROVE THE SEMANTIC RELEVANCE OF RETREIVED MEDICAL TEXT

A Dissertation
Presented to
The Graduate School of
Clemson University

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy
School of Computing

By
William Phillip Taylor II
May 2010

Accepted by:
Dr. Zijun Wang, Committee Chair
Dr. John D. McGregor
Dr. Pradip K. Srimani
Dr. Feng Luo

ABSTRACT

Medical Subject Headings (MeSH) is a controlled vocabulary used by the National Library of Medicine to index medical articles, abstracts, and journals contained within the MEDLINE database. Although MeSH imposes uniformity and consistency in the indexing process, it has been proven that using MeSH indices only result in a small increase in precision over free-text indexing. Moreover, studies have shown that the use of controlled vocabularies in the indexing process is not an effective method to increase semantic relevance in information retrieval.

To address the need for semantic relevance, we present an ontology-based information retrieval system for the MEDLINE collection that result in a 37.5% increase in precision when compared to free-text indexing systems. The presented system focuses on the ontology to: provide an alternative to text-representation for medical articles, finding relationships among co-occurring terms in abstracts, and to index terms that appear in text as well as discovered relationships. The presented system is then compared to existing MeSH and Free-Text information retrieval systems.

This dissertation provides a proof-of-concept for an online retrieval system capable of providing increased semantic relevance when searching through medical abstracts in MEDLINE.

ACKNOWLEDGEMENTS

It is a pleasure to thank those who made this thesis possible: Dr. Wang, Dr. Grossman, Dr. S M. Hedetniemi, and Lea Benson. Each of you has provided a significant impact in this work whether directly or indirectly and all or your efforts are much appreciated. I am indebted to many of my colleagues who supported me Tiras Eaddy, Rondey Smalls, Tyiesha Arrington, and Shelby Darnel either through offering different perspectives on possible avenues in this research or through friendly causerie uplifting my spirits and boosting morale. Lastly, thank you to my parents William & Sonia Taylor for your never-ending encouragement, support, and love.

TABLE OF CONTENTS

Page

FIGURES

TABLES

\

EQUATIONS

Chapter 1

Introduction

To provide health professionals access to information necessary for research, health care, and education the National Library of Medicine has been a leading contributor in indexing biomedical literature. What began as a printed index to articles, the Index Medicus, has grown to a database containing in excess of 6 million records representing articles in biomedicine known as MEDLINE. As articles in biomedicine became available on the web it is natural to assume the medical community would have a need to search through collections of articles for their own purposes. Needs of this nature have given rise to information retrieval systems whose purpose is to answer a user's 'questions' or query by responding with appropriate relevant documents that 'answer' the user's 'question'. The most prevalent information retrieval system is a Free-Text system. These systems rely on matching keywords in the user's query to keywords found in the desired document – this is not the best solution.

As meanings of words and understanding of concepts differ in different communities, different users might use the same word for different concepts (polysemy) or use different words for the same concept (synonymy). Thus, matching only keywords may not accurately retrieve semantically similar medical documents. To address these issues the National Library of Medicine has instituted the use of controlled vocabularies in an effort to curtail the effects of these phenomena. This work discounts the effectiveness of controlled-vocabularies in information retrieval and offers an entirely new method for retrieval systems - each being explored in this work.

This thesis explores the creation of an information retrieval system designed specifically for the retrieval of medical abstracts from the MEDLINE database. This work begins by reviewing the

leading and most popular medical search engine available, the National Library of Medicine's (NLM) Pubmed retrieval system. This system will become the baseline from which the presented information retrieval system will be compared. In reviewing the NLM approach to retrieval discussion begins with the MEDLINE database. Next, a review on indexing strategies as performed on medical abstracts contained in MEDLINE using the Medical Subject Headings (MeSH) controlled vocabulary is presented. Finally, focus is devoted to presenting the inner-workings of Pubmed and the method by which PubMed uses MeSH headings to retrieve medical abstracts from the MEDLINE database and to resolve variations in term-use from searchers.

1.1 MEDLINE

MEDLINE is the National Library of Medicine's premier database that houses medical journals and articles in the life sciences with a concentration in biomedicine. The database includes topics such as microbiology, nutrition, pharmacy, and environmental health and covers categories as anatomy, organisms, disease, psychiatry, psychology and the physical sciences. It also leverages a controlled vocabulary, meaning that there is a specific set of terms used to describe each stored article; describing each article is generally known as indexing. A distinctive feature of MEDLINE is that the records are indexed with the MeSH vocabulary to facilitate retrieval by regular users, researchers, students, and doctors. Users who are familiar with the MeSH vocabulary are typically better searchers then those users who are unfamiliar with the specialized vocabulary.

1.2 Medical Subject Headings (MeSH)

MeSH is the U.S. National Library of Medicine's controlled vocabulary used for indexing articles for MEDLINE. MeSH terminology provides a consistent way to retrieve information that may use different terminology for the same concepts and imposes uniformity and consistency in the indexing of biomedical literature. Its vocabulary is arranged in a hierarchical categorized manner called the MeSH tree structure, pictured in *Figure 1*.

A. Anatomy
B. Organisms
C. Diseases
D. Chemical and Drugs
E. Analytical, Diagnostic and Therapeutic Techniques and Equipment
F. Psychiatry and Psychology
G. Biological Sciences
H. Natural Sciences
I. Anthropology, Education, Sociology and Social Phenomena
J. Technology, Industry, Agriculture
K. Humanities
L. Information Science
M. Named Groups
N. Health Care
V. Publication Characteristics
Z. Geographic Locations

Figure 1: MeSH Tree Structure

This structure contains sixteen branches with each representing a distinctive subject area. The further one travels through the MeSH tree structure the more specific the terms in that area become; terms may also co-occur in multiple places within the MeSH hierarchy. In use, the NLM human indexers examine articles to be added to MEDLINE and assign the most specific MeSH heading(s) that appropriately describe the content of the article. The indexer will assign as many MeSH headings as appropriate to cover the topics of the article, generally five to fifteen headings per article. In the event that there is no suitable MeSH heading for a concept, the indexer will use the closest MeSH heading available. Once articles have been indexed with MeSH headings the articles then become available to searchers.

1.3 NLM Pubmed Information Retrieval System

Users search through MEDLINE through Pubmed. Pubmed is a web-based information retrieval system developed by the National Center for Biotechnology Information (NCBI) to provide access to citations from biomedical literature. The weaknesses of the Pubmed information retrieval system are made manifest when indexing medical articles and resolving users search queries to

indexes. In an effort to build an information retrieval system based on semantic retrieval, Pubmed has heavily utilized the MeSH vocabulary in its indexing and user-querying components. The MeSH vocabulary contains 27,000 unique descriptors derived from medical concepts; the NLM Meta-Thesaurus is comprised of 870,000 medical concepts. The primary disadvantage of the Pubmed system is that it indexes the six million medical abstracts housed in MEDLINE with less than 3.1% of the available medical vocabulary. This disadvantage is made evident when searching through Pubmed and retrieving results that are semantically close to the information requested, but not sufficiently narrow resulting in very low precision & recall and requiring multiple searches by users.

In an information retrieval system there are primarily five components [38]: parsing linguistics, indexers, indexing-store or 'Indexes', query-parser, and ranking, see *Figure 2*.

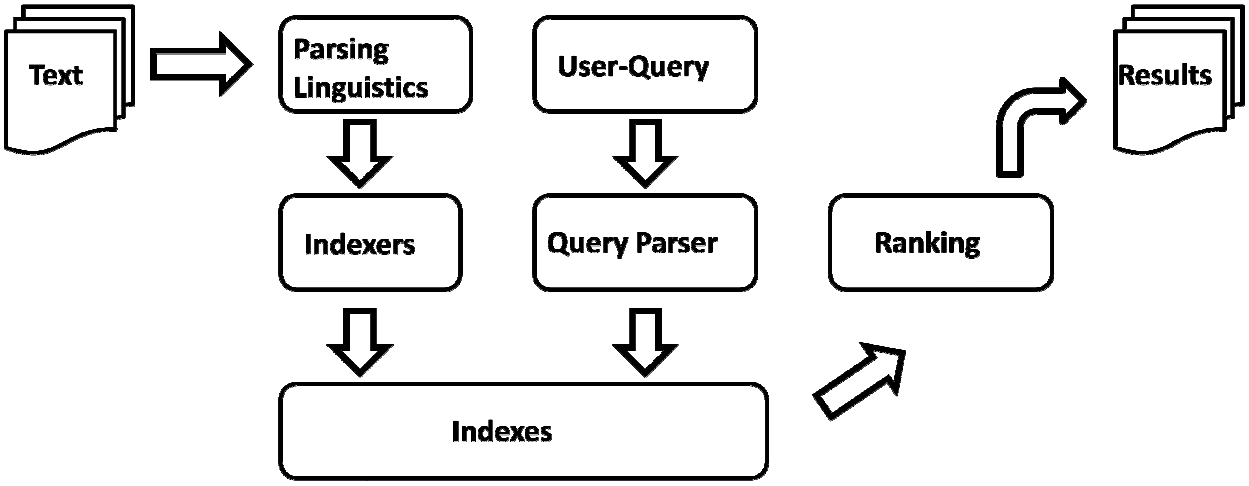


Figure 2: Information Retrieval System Components

The parsing linguistics component is commonly used for the removal of terms that appear in a user defined stop-list. This component is a preprocessing step for text that will become part of the collection. The indexer component is used to create an index for each term in the text to identify the documents the term makes an appearance, while the indexing-store or 'Indexes' is typically a database

storing the index. The query-parser component performs logic to map a user's query to an index to retrieve text from the database and the ranking component is used to rank the documents returned by a query to present the most relevant documents to the user in ascending order.

To respond to a user's search request with relevant search results, Pubmed's information retrieval system has been designed with the use of a niche medical vocabulary, MeSH, in the indexer and query-parser components. The Pubmed information retrieval system is pictured in *Figure 3* and each component is discussed in further detail.

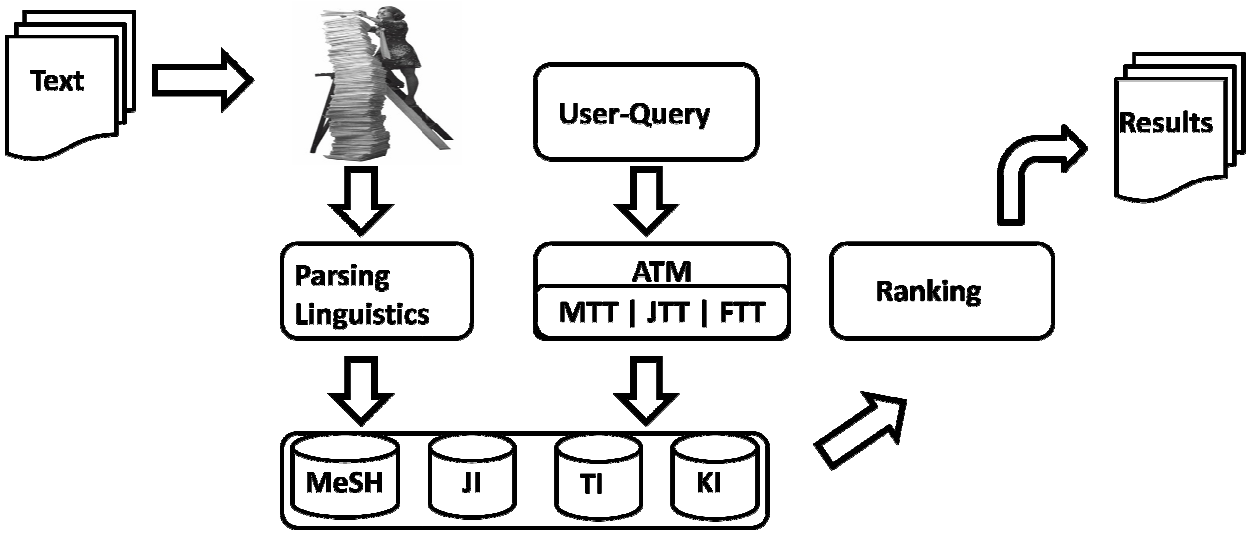


Figure 3: Pubmed Information Retrieval System

To avoid a misclassification of abstracts by automated methods, Pubmed replaces the traditional indexing component with human indexers. It is the duty of the human indexers, employed by the NLM, to read and index new medical texts appropriately with the MeSH vocabulary. This vocabulary is a subset of the NLM Meta-Thesaurus and as of this writing is comprised of approximately 27,000 unique medical concepts called MeSH descriptors. Each medical article contained within MEDLINE has been indexed with one or multiple MeSH descriptors identifying its context. Retrieving articles then becomes a process of mapping a user's query to MeSH descriptors.

Once human indexers have assigned MeSH descriptors to medical articles, the articles are sent through the parsing linguistics component. Pubmed's parsing linguistics component is consistent with modern implementations in information retrieval systems. This component seeks to remove terms from abstracts that add no additional semantic value, e.g. terms that have proven not to be good discriminators - these terms include common nouns, pronouns, adjectives, and many others; any remaining terms are used as keyword indices.

After the parsing linguistic phase, the MeSH and keyword indices for the medical article are stored in a central location, usually a database or data warehouse. This is the index-store component. The indexing-store component contains a listing of identifiers (usually the vocabulary of the text-collection) and the text in which they appear. Retrieval is then dependent on users using the text-collection vocabulary to express their information needs. PubMed has four such index-stores for journals, titles, authors, keywords, and MeSH. Only the keywords and MeSH descriptors have been discussed as the other indices make significantly less contributions when retrieving medical articles.

In Pubmed, the query-parsing component is where the logic behind mapping a user's query to abstract indices resides. Traditional query-parsers inside modern search engines (yahoo, google) perform little alterations to a user's search terms besides term-stemming before retrieval. Because MEDLINE abstracts are indexed with MeSH descriptors, Pubmed has developed Automatic Term Mapping (ATM) [12,40]. ATM serves to resolve a search-query to MeSH descriptor(s) equivalent. Given a user's search-query ATM attempts to resolve the search-query by referring to three tables: MeSH Translation Table (MTT), Journal Translation Table (JTT), and Full Author Translation Table (FTT). When searching for medical abstracts if ATM can resolve the query-terms based on MTT then the MeSH descriptors found are sent along with query terms in the retrieval process. In the event that query terms cannot be resolved by MTT then ATM uses the search-query terms as purely keyword based indexes in the retrieval process. The journal and full author translation tables are not discussed

as they are not reliant on the content of medical abstracts.

An advantage to this design is that when a search-query cannot be mapped to a MeSH descriptor, the retrieval system is reduced to a basic keyword indexed search, however, still retaining the problems of synonymy and polysemy [12,40]. It is due to this safeguard that Pubmed, under worst-case circumstances, performs no worse than a free-text search engine [58]. Pubmed's use of the controlled vocabulary is a step in the right direction as [58] has proven that indexing using the MeSH vocabulary does prove to perform better than using Free-Text / keyword-based indices alone; however because the MeSH descriptors are a broad category of terms, retrieving text semantically close to the initial user's query usually results in Pubmed using keyword-indices to resolve the query terms because specific concepts in the user query may be not captured in MeSH. In the next section existing research is explored for information retrieval for the MEDLINE database.

1.4 Existing Work in Information Retrieval for MEDLINE

There has been much debate over the effectiveness of document-indexing strategies which utilize indexes created from free-text to that of those created with controlled-vocabularies, e.g. MeSH. Many Information Retrieval (IR) Systems deployed in industry and academia are based on some component of free-text indexing. Free-Text indexing is based on keyword-indexed backend data-sources where users submit descriptive queries. The keywords used in the query are then used unmodified to retrieve relevant documents. The primary cause in migrating from a free-text system is the limited expressive-power of keywords; especially in collection-sets containing uncommon vocabulary. The terms in a free-text query map directly to documents containing only those terms. The disadvantage of this approach is found in its failure to resolve the synonymy of query terms and its failure in identifying the relationships that may exist among the terms used in the query itself.

Research has moved towards resolving the former disadvantages with the advent and subsequent adoption of incorporating controlled-vocabularies to aid in diminishing the severity of the concept-

resolution dilemma. The main focus of this dissertation is concerned with the document indexing of medical literature. The most widely used controlled vocabulary used to index medical literature is the National Library of Medicine's (NLM) Meta-thesaurus. Researchers have used the Meta-thesaurus in a myriad of ways, primarily using it to manually maintain an indexing process, to construct a semi-automatic indexing process, or to create a fully-automatic indexing process – all to index medical literature. Each will be discussed further below.

The first-approach gives rise to the NLM Medical Subject Headings (MeSH) vocabulary. The MeSH vocabulary is a subset of medical concepts contained within the NLM Meta-thesaurus and used to map concepts found in a user's query to MeSH indices to retrieve documents containing identical MeSH indexes. Srinivasan [58], has found that combining Free-Text and MeSH in document indexing leads to a 8.2% increase in 11-pt average precision when compared to Free-Text indexing alone using the SMART retrieval engine [40]. Srinivasan, [58] has also found that using MeSH alone (not paired with Free-Indexing) performs 7% worse than Free-Text indexing alone. The downside of using MeSH is that its maintenance is a completely manual process. MeSH behaves as a simple mapping between user-query and MeSH concept, but does not take into account how a query term is used in context to other terms in the same query.

In creating a fully-automatic approach to indexing text researchers have shifted to refining concepts found in controlled-vocabularies to concepts found directly in text and then indexing by the refined concepts [3,33,37,43,45,62]. Their efforts culminate into the development of MetaMap, CHARTLINE, and SAPPHIRE, among others. Still other approaches have mapped Free-Text directly to MeSH descriptors [1,2,5]. Each method mentioned thus far suffers from the same challenges of overcoming poor recall levels. In fact MetaMap developed by NLM was thoroughly tested alongside the use of NLM employed indexers and reported only approximately 32% satisfactory rate among human-indexers in assigning abstracts to appropriate MeSH descriptors when using MetaMap.

Semi-automatic methods [4,60] such as the Medical Text Indexer (MTI) suffer the same failures as the fully automatic methods but also have the disadvantage of requiring a human component in creating the index.

Examining indexing vocabularies in context with MEDLINE medical database, researchers have observed that in most cases controlled-vocabulary based indexing does not perform better than free-text indexing [16,17,19,60,65] at best it has been shown that performance results are nearly identical. Despite the many advancements in this area researchers have centered on utilizing free-text or controlled-vocabularies in document indexing - relatively little emphasis has been given to indexing strategies that simultaneously exploit the information in free-text, controlled vocabularies, and lexical databases.

In this thesis, an information retrieval system with the capability to retrieve semantically relevant medical text is presented - human indexers and controlled vocabularies that represent only a fraction of the medical vocabulary needed to index text are removed and automated methods are developed for their replacement. Instead, efforts are directed at discovering the relationships of terms in medical text and indexing those terms and found relationships.

1.5 Thesis Outline

The outline of this dissertation follows *Figure 2*. Chapters 2 -4 represent the parsing linguistic components of the presented information retrieval system. Chapter 2 begins discussion with the difficulties in representing text, introducing the ontology as an alternative method to represent text, and concludes with the discussion on using ontology to create a Concept-Forest. Chapter 3 is devoted to validating the Concept-Forest as a text-representation alternative by using document clustering. In validating the Concept-Forest approach a survey is taken of popular keyword-based algorithms and the algorithms are compared against two similarity measurements developed specifically for Concept-Forests to prove that the Concept-Forest has merit. Chapter 4 highlights the disadvantages of the

Concept-Forest in its inability to represent medical concepts found in medical abstracts. This chapter serves to remove this disadvantage by temporarily merging two data-sources, the NLM Meta-Thesaurus and WordNet, to aid in discovering medical relationships from medical abstracts. Chapter 5 concludes the presentation on the parsing-linguistics component for the presented information retrieval system. The remaining components of the presented information retrieval system, indexing, query-parsers, and ranking are reviewed in Chapter 5.

Chapter 6 offers a comparative study of the presented ontology indexed information retrieval system with that of a MeSH indexed system similar to Pubmed. Details of the MeSH indexed system is discussed in detail. This chapter also serves to divulge how the retrieval system will be compared and metrics used in evaluation. The remaining two chapters detail an online architecture and implementation of the ontology indexed information retrieval system (Chapter 7). In Chapter 8, final thoughts on this work, investigating shortcomings and future directions are presented.

Chapter 2

Difficulties of Text Representation

This chapter reviews the parsing linguistics component of the presented information retrieval system. In information retrieval this component is traditionally devoted to text preprocessing tasks, of which duties include converting text to a common case and removing terms that appear in a user-defined stop-list. The approach taken in the presented information retrieval system follows traditional retrieval methods while also introducing techniques to determine synonymous and polysemy relationships between co-occurring terms in text. Resolving synonymous and polysemy relationships between co-occurring terms in text is an important area of study as it is able to widen or narrow the disparities between texts. This chapter presents background on the challenges of representing text, the idea of the ontology to surmount the arisen challenges, the WordNet ontology, and an alternative form of text-representation using the WordNet ontology, the Concept-Forest.

2.1 Background

Currently, keywords-based techniques are commonly used in various information retrieval and text mining applications. Among these keywords-based methods, Vector Space Model [69] and Latent Semantic Indexing [31] are the most widely adopted.

Using VSM, a text document is represented by a vector of the frequencies of terms appearing in this document. The similarity between two text documents is measured as the cosine similarity between their term frequency vectors; however, a major drawback of the keywords-based VSM approach is its inability of handling the polysemy and synonymy phenomenon of the natural language. As meanings of words and understanding of concepts differ in different communities, different users might use the same word for different concepts (polysemy) or use different words for the same concept (synonymy).

Thus, matching only keywords may not accurately reveal the semantic similarity among text documents or between search criteria and text documents due to the heterogeneity and independency of data sources and data repositories. For example, the keyword “java” can represent three different concepts: coffee, an island, or a programming language, while keywords “dog” and “canine” may represent the same concept in different documents.

LSI tries to overcome the limitation of VSM by using statistically derived conceptual indices to represent text documents and queries. LSI assumes that there is an underlying latent structure in word usage that is partially obscured by variability of word choice and tries to address the polysemy and synonymy problems through modeling the co-occurrence of keywords in documents. Though earlier studies contend that LSI may implicitly reveal concepts through the co-occurrence of keywords, we found that the co-occurrence of keywords in documents may not necessarily mean the contextuality, especially in multi-disciplinary research papers such as biomedical research papers. This is exactly why using LSI-based tools to extract terms from shorter documents, such as medical abstracts, is a questionable practice. These short documents may not provide enough co-occurrence information for the LSI-based semantic similarity measurement.

Still other retrieval mining applications have used controlled vocabularies with VSM and LSI to represent text. Although controlled vocabularies provide uniformity and consistency in representing text, controlled vocabularies are rarely all-inclusive and have the additional drawback of requiring constant updates. This means that the synonym and polysemy issue will always present a problem for controlled vocabularies unless the vocabulary is over a very specific domain.

2.2 Ontology

Instead of representing shorter length documents by their keywords which has proven to be insufficient in resolving the synonymy and polysemy of natural language, a better method for representing text is by using the co-occurrence of terms that appear in the text. Using the notion of the ontology, term relationships can be discovered in documents utilizing co-occurring terms;

synonyms can be discovered for terms and the severity of polysemy can be reduced.

Term relationships in text are discovered through a lexical aid called ontology. Ontology is defined as a set of representational primitives, in which these primitives contain attributes/properties and may contain relationships with other primitives. The entire set of primitives is known as the universe of discourse and is all information that can be represented by the ontology. For the purposes of this thesis, representational primitives are terms. These terms hold attributes/properties which are in the form of lexical information such as part-of-speech, synonyms, hypernyms, meronyms, and hyponyms. The relationships between terms can be synonymous, polysemous, or by hypernymy.

The advantages of using the ontology are its ability in resolving synonymy, polysemy, and hypernym problems. In resolving synonyms between terms in two differing text, using the ontology affords the opportunity to add synonyms for each term appearing in text. Then, when comparing the texts for similarity based on a similarity algorithm, similarity between the two synonymous terms can now be captured and contribute to the similarity score.

In lessening the polysemy issue, using the hypernym structure of the ontology in tandem with using co-occurring terms in text facilitates the use of word-sense disambiguation. The correct sense of a term can be identified by using the ontology and co-occurring terms in text. In terms of similarity measurements resolving polysemy issues lead to a more accurate similarity score. Using the ontology no longer require that two documents being compared for similarity, use the same vocabulary as synonyms and polysemes of the two texts can be now be discovered. An example ontology includes WordNet, and SWEET.

2.3 WordNet Ontology

Using the WordNet Ontology helps to achieve our goal by providing a rich lexicon of common English terms and in-depth lexical information per term. The lexical information provided is synonyms, polysemes, hypernyms, meronyms, hyponyms, and many others. WordNet is a lexical

database of English words, in which nouns, verbs, adjectives and adverbs are grouped into cognitive-sets, synsets, with each synset representing an individual concept. Each synset contains a group of synonymous words with different senses of a word being in different synsets. Most synsets are connected to other synsets through semantic relations, such as hypernym, hyponym, etc. These synsets are interlinked by means of conceptual and lexical relations. The dominant semantic relationship in WordNet is the hypernym relationship, the "IS_A" relationship. Most nouns and verbs are organized into hierarchies, defined by hypernym relationships. For example, *Figure 4* depicts the hypernym hierarchy for the first sense of the word "car".

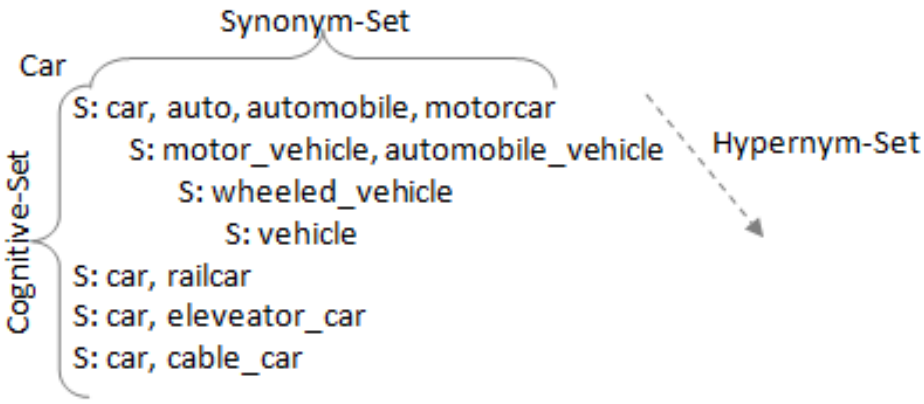


Figure 4: Sample Hypernym Structure

Through the use of the WordNet ontology we gain the capability to discover many differing relationships between terms in text. This ontology serves to aid us in resolving the synonymy problem by providing us a list of synonyms for common English terms. The synonyms that WordNet provides can then be used in determining the similarity of two texts based on the concepts that the texts convey and not on the appearance of identical vocabulary. With the polysemy issue, this ontology provides the presented work with a listing of differing ways or senses in which a term can be used. Then using co-occurring terms in text and/or synonyms for the term it can be determined the correct 'sense' that the term is being used in the text. Discovering the correct sense of a term in usage, affords the opportunity to 'tag' or identify the sense by a unique identifier. Then, when comparing

two texts for similarity it can be determined if two texts are using the same term, while also determining if the two texts are using the same term in the same way.

2.4 Existing Work – Ontology in Use

Many studies have used ontology to assist in information retrieval and text document processing. These ontology-based approaches can be divided into two categories. One category of ontology-based methods [8,13,27,32,34,36,55,59] apply machine learning methods, such as clustering analysis and fuzzy logic, to construct ontology's from text documents and then use the resulting ontology to assist in information retrieval, text categorization, or text document processing [22,43,55,61,67]. The performance of such a system is completely dependent upon the black-box machine-learning implementation that does not lend itself to facile modification. During the corpus analysis, terms rarely appearing in the document corpus are often ignored because of their low frequencies of occurrence. However, terms that occur infrequently in text may be considered unique and hold high value for information retrieval according to information theory. Ignoring infrequent terms in the constructed ontology may affect the performance in information retrieval experiments.

The second group of ontology-based methods utilizes an existing ontology, such as WordNet [14], to assist information retrieval. These methods use three different approaches to take advantage of the existing ontological knowledge. The first approach [24,29,30,36,48,51] involves using WordNet to find synonyms or hypernyms of terms to improve the performance of information retrieval and text document processing. However, this approach may introduce irrelevant data by including semantic information that is not present in the document text. To illustrate, given a document about "beef" and a document about "pork", a hypernym-based method may use "meat" to replace "beef" and "pork" because the two terms have a common parent hypernym relationship "meat". This approach over-simplifies or over-generalizes the concept, making it impossible to distinguish documents containing "beef" from documents containing "pork". A second issue that arises with this approach is that it does

not perform word sense disambiguation. Word-sense disambiguation is the process of resolving the problem of polysemy for term concepts. In their approach, all synonyms or hypernyms related to a keyword are used to replace the keyword. These weaknesses often lead to disappointing performances [22,26]. The second approach focuses on word sense disambiguation [11,35,68,73] to address the synonymy and polysemy problem in natural language processing. This approach tries to determine an exact sense for a term, often resulting in a misclassification of terms. This approach fails to measure the impact of the semantic similarities and relationships among different terms in the same text document to disambiguate word senses.

To address the problems in the first two approaches, the third approach applies various techniques [23,25,52,54] to discover the semantic similarities and relationships of terms and uses them to enhance the keywords-based information retrieval and text document processing methods of the Vector Space Model (VSM). However, the techniques used to discover the term relationships and similarities have their weaknesses. Sedding [52] used a naive, syntax-based disambiguation approach by assigning each word a part-of-speech (PoS) tag and by enriching the “bag-of-words” data representation, which were often used for document clustering by extracting synonyms and hypernyms from WordNet. Unfortunately, this study found that including synonyms and hypernyms, disambiguated only by PoS tags, does not improve the effectiveness of text document clustering. The authors attributed this underperformance to noise introduced by incorrect senses retrieved from WordNet and concluded that disambiguation by PoS alone is insufficient to reveal background knowledge in information retrieval.

While the studies presented suggest exploiting term relationships and similarities using WordNet may not improve the performance of information retrieval, other studies using different methods imply that it is possible to use supplemental information to improve the performance of the keywords-based VSM. Huang [23] used a guided self-organization map (SOM), a result of merging statistical methods, competitive neural models, and semantic relationships obtained from WordNet, to improve

the performance of the traditional VSM. However, certain human involvement is required to build the guided SOM. Jing [25] calculates a mutual information matrix for all terms in the documents based on information obtained from WordNet and uses the mutual information to enhance the keywords-based VSM method. However automatically computing term mutual information (TMI) is sometimes problematic and may lead to incorrect conclusions about the quality of the learned mutual similarity [49]. Even though using SOM and TMI can improve the performance of the keywords-based VSM, their performance in comparison to Latent Semantic Indexing (LSI), the leading keywords-based method, has not been investigated.

2.5 Proposed Concept-Forest

To address the issues in existing ontology-based methods, we propose an ontology-assisted method to capture the semantic relationships of terms in text; the method proposed recognizes conceptually and semantically related terms. This new method constructs a concept forest (CF) from a text document, based on the co-occurrence of terms and their semantic relationships found in WordNet. The CF will be used to represent the semantic relationships between terms in the same text document. A unique feature of the leveraged CF-based method is that we derive the concept forest based only on analyzing the co-occurrences and relationships of terms within a single document.

A Concept-Forest is used as a word-sense disambiguation and synonymy resolution tool which allows the narrowing of concepts a word is associated with – resolves polysemy problem - and broadening a terms representation by including synonyms – resolves differing vocabulary for same concept problem. The Concept-Forest also has the capability of synonym and ancestral derivative resolutions via its hypernym relationship. The Concept-Forest is needed to determine the true similarity between texts. Once terms synonymous and polysemous relationships have been addressed for all terms in text documents then gauging the similarity of the texts becomes a straight forward process. Text similarity becomes dependent upon if a term or any of its synonymous concepts appear

in the text, if so then the two texts share some degree of similarity and the same can be said of the polysemy relationship.

WordNet serves as the universe of discourse of all information that can be resolved using the Concept-Forest. In practice the Concept-Forest is a network of nodes where edges are representative of resolved concept-clashes and nodes represent terms and their associated synonym-sets; a concept-clash is a term which has multiple disparate meanings, e.g. a problem of polysemy. Together the nodes (with associated synonym-sets) and edges (represented by unique concept identifiers) encompass a Concept-Forest.

2.5.1 Formal Definition of Concept-Forests

An concept forest is defined as a Directed Acyclic Graph (DAG): CF = [*T, E, R*], where $T = \{t_1, t_2, \ldots, t_n\}$ is a set of stemmed terms and $E = \{e_1, e_2, \ldots, e_m\}$ is a set of edges connecting terms with relationships defined in $R = \{r_1, r_2, \ldots, r_k\}$. Specifically, an edge e_i is defined as a triplet $[t_{i1}, t_{i2}, r_j]$ where $t_{i1}, t_{i2} \in T$ and $r_j \in R$. In addition, two terms can be linked by only one relationship, that is,

$$\forall l \neq k, [t_i, t_j, r_k] \in E \Rightarrow [t_i, t_j, r_l] \notin E .$$

Given two concept forests $CF_1 = [T_1, E_1, R_1]$ and $CF_2 = [T_2, E_2, R_2]$, determining the semantic similarity of the two concept-forests must consider the similarities of their associated term sets, edge sets, and relationship sets.

2.5.2 Building the Concept-Forest

The algorithm to construct a concept-forest can be defined as follows: Given a text document, we initially extract all keywords and their occurrence frequencies from the document. Next, keywords are removed from consideration that are included in the stop-list or that are pronouns, common verbs, common nouns, and adjectives; these terms were found to add little or no value in determining the document's semantic content by previous studies [10,50]. Then, a WordNet morphology interface (function morphstr()) is used to stem remaining keywords, i.e., to map inflected (or sometimes derived)

words to their stem, base or root form. For instance, “cared”, “cares”, “caring”, “careless”, and “carelessness” are all mapped to the root word “care”. After term stemming, each term’s cognitive-set is retrieved from WordNet. Next, to determine the proper synset to include in the CF we use term co-occurrence information in the document and the semantic relationships of senses defined in WordNet to draw distinctions among terms.

Our procedure checks every pair of stemmed words obtained from the text document to determine whether there are semantic relationships between their senses defined in WordNet. Given two terms (t_1 and t_2) obtained from the same text document, if their respective synsets s_1 and s_2 have a relationship, the synsetID of s_1 or s_2 is used (s_1 and s_2 represent the same concept) to represent the concepts of t_1 and t_2 and other senses of t_1 and t_2 will be discarded. Meanwhile, a relationship link is formed between the terms of t_1 and t_2 with the synsetID of s_1 and s_2.

If a keyword contains multiple senses and no relationships are found in the text for said keyword, then the original stemmed keyword will be used in the Concept-Forest. The keyword will serves as its own root due to insufficient information to disambiguate its concepts [9,21,23,34,33,55,56,67,,73]. Finally if a keyword has many senses and one or more senses were mapped to relationships in another term then only the senses that were mapped are kept – remaining senses are disregarded deemed irrelevant and adding no increase in the discovered knowledge. This process completes when all pairs of stemmed words are investigated.

To illustrate, given a document containing words “disease”, “sickness”, “influenza”, “drug” and “medicine”, we can construct a concept tree for terms “disease”, “sickness” and “influenza” using a relationship link based on the hypernym relationship among these terms**Error! Reference source not found.**. Similarly, a concept tree can be built for terms “drug” and “medicine”. These two concept trees form a concept forest depicted in Figure 5 for the document.

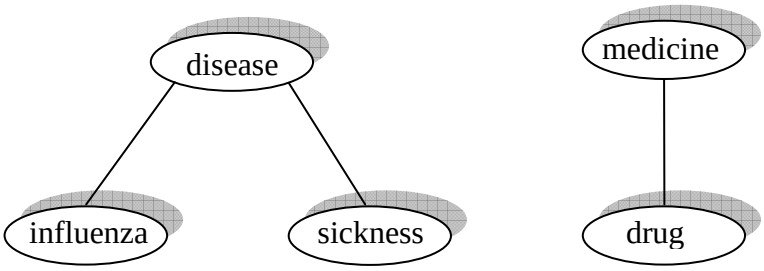


Figure 5: Sample Concept Forest from Using WordNet Only

We note that the terms and not their related synsetIDs are shown in the concept forest for demonstration purposes only. In actual concept forests, the synsetIDs are used to represent the links between concepts. The pseudo-code can be found below:

Pseudo-code: Creating Concept-Forest

```
Begin
        Remove terms in D that appear in stop-list
        Remove duplicate terms from D
        For-Each term t in D
                root := createRoot(t)
                retrieveCognitiveSets(root)
                Add root to C
        End For-Each

        For-Each root r1 in C
                For-Each root r2 in C
                        If r1 != r2
                                For-Each synset s in r1
                                        For-Each hypernym h in s
                                                If r2 in h
                                                        id := s->id
                                                        createLink(r1,r2)
                                                        nameEdge(id,r1,r2)
                                                End-If
                                        End For-Each
                                End For-Each
                        End-If
                End For-Each
        End For-Each
End
```

Variables	*functions*	*Definitions and/or Descriptions*
D		document being processed
C		set of all nodes within concept-forest
s		synonym-set
h		hierarchy of concepts related to the synonym-set, based on derivative IS-A relationships
root		Node which houses cognitive-set (synonym-set and hypernym-set) data for each term
	createRoot	Creates a node
	createLink	Creates a link between two nodes
	nameEdge	Names the edge between two linked nodes by the relationship found (synset-id)
	retrieveCognitiveSets	Takes the term from the node passed as a parameter and retrieves the NOUN and VERB cognitive-sets for the term and assigns it to the node

Figure 6: Concept-Tree Algorithm, Variables, & Definitions

2.5.3 Semantic Content Purification

A concept forest constructed by the procedure outlined in Section 2.4.2 may contain terms or synsetIDs that are not closely related to the main topics of the text document, and these terms or synsetIDs may sometimes introduce noise to information retrieval and text document processing. To address this issue, we use the frequencies of terms occurring in the text document to calculate a semantic content rate (SCR) for a concept tree in the concept forest. We note that a concept tree may contain only a single stemmed word.

Each stemmed word obtained from a text document has an associated word frequency value corresponding to the number of occurrences found in the text. When a stemmed word is mapped to a particular synsetID during concept forest construction, the associated word frequency value is transferred to the synsetID. If several stemmed words are mapped to the same synsetID, the word frequency value of this synsetID is the sum of the word frequency values of these associated words. We further define the semantic content weight for a concept tree as the sum of the word frequency values of all its associated synsetIDs. For a single-node tree, its semantic content weight is the word

frequency value of this single node.

Assuming the semantic content weights of concept trees in a concept forest are $w_1, w_2, \ldots, w_n$, respectively, the semantic content rate of concept tree i is defined as:

$$SCR_i = w_i \Big/ \sum_{j=1}^{n} w_j$$

Equation 1: Subject Content Rate Formula

The SCR values of a concept forest indicate the semantic organization of the associated text document. A concept forest obtained from a clearly and concisely written single-topic abstract may contain a concept tree having an SCR value greater than 85%, while the concept forest obtained from a long multiple-topic text document may contain several concept trees with smaller SCR values. To purify the semantic content of a concept forest, we use a threshold (i.e., 5%) to filter out concept trees with low SCR values. Any concept tree whose SCR value falls below this threshold will be removed from the final purified concept forest. When using the SCR terms that serve as their own root are not removed from the Concept-Forest as to protect the possibility of unique terms that may not share relationships with other terms in text. The SCR is not used in the research but is presented as a way to reduce the terms in text by considering the significance of their relationships with co-occurring terms in text.

In the upcoming chapter, we continue in the parsing linguistics phase of the presented information retrieval system. Chapter 2 is devoted to verifying claims that the presented method for text-representation, the Concept-Forest, sufficiently and accurately removes the problems of synonymy and polysemy that Vector Space Model and Latent Semantic Indexing do not resolve. Verification begins by presenting methods to determine ontology similarity and performing a comparative study with popular methods Vector Space Model, Latent Semantic Indexing and Part of Speech tagging with N*

Grams. Document clustering experiments are used to provide evidence that the Concept-Forest representation of documents performs better than keywords based representations.

Chapter 3

Validation by Document Clustering

Chapter three continues work devoted to the parsing linguistics component in the presented information retrieval system. The purpose of this chapter is to determine if sufficient semantic information is present in the proposed form of text-representation, the Concept-Forest, to rival leading text categorization measurements that utilize only keywords. To meet this purpose this chapter explores the uses of the Vector Space Model (VSM), Latent Semantic Indexing (LSI), and language modeling and explores two new similarity measurement algorithms to measure the similarity between Concept-Forests. This chapter concludes with performance comparison of keywords-based and Concept-Forest approaches.

3.1 Existing Work – Similarity Measurements

Aside from VSM and LSI, spoken in-depthly in the previous chapter, there is yet another similarity metric heavily used in text categorization, Part-of-Speech Tagging with N*Grams. Part-of-Speech tagging with N*Grams is a text categorization method based on language models. A language model is a function that puts a probability measure over the strings drawn from some vocabulary [38]. Because the language model of a document represents the probabilities of each term occurring in the model, to gain the probability of a word sequence, the probability of each term in the word sequence is multiplied by the probability that the term appears in the model.

In text-categorization, a training-set is set-aside and a language model is created for each text in the training-set. To classify new documents, the document is classified based on the probability of the sentences in the chosen document to appear in a predetermined set of categories that exist in the

training-set. Whichever language model returns the highest score for the sentences in the document, then the document is assigned the same category of the language model by which it relates; there are also many spin-offs in determining classification.

The N*Gram model is type of language model in which the probability of occurrence of a term is dependent upon the prior occurrence of N-1 other symbols [38]. N*Gram is typically constructed from statistics obtained from a large corpus of text using the co-occurrences of words in the corpus to determine word sequence probabilities. These models rely on the likelihood of sequences of words, such as word pairs (in the case of bigrams) or word triples (in the case of trigrams) and is are traditionally used with an a-priori likelihood P(W) of a given word sequence W. The N*Grams method suffers from the same disadvantages of the VSM and LSI approaches. N*Grams performs at its best for longer length text, this is so because longer length text contains more terms than shorter length text and longer text introduces a wider set of different vocabulary contained within the text. For shorter length text there is little information regarding the probability of terms in text to generate a language model that is representative of the text. The N*Grams approach also needs training information and classifies new documents based on its training data. This approach is disadvantageous when the training-set data is significantly small in length of text or when the training-set is small in the number of documents it contains. Lastly, because this approach is based on the likelihood that a term or string of terms may appear in text, the issue of synonymy is still a relevant one. N*Grams computes the likelihood of P(W) of a given word sequence W, but not the probability of W where the terms contained within W have synonymous relationships.

Still other research in this area has been devoted to determining similarity of user annotated ontology such as OWL, DAML-OIL, and SHOE ontology [5]. This type of ontology takes the form of user defined xml like structures including tags with attributes that reflect the content of the text. Other researchers have taken a semiotic view to similarity [53] in which edit-distance and term-

matching are supreme or using ontology-vectors (much like VSM) [53]; this approach does not view the edge-relationship which may determine the sense in the term's use. Euzenta [7] presents a study for similarity but it does not give a definition to how their ontology is constructed; however it does present a survey of existing similarity measurements that are being applied throughout the research community to tackle this problem. Some of the techniques explored in [7] will be the starting point in the creation of the presented ontology-based similarity measurements.

3.2 Proposed Similarity Measurements – Concept Forest Distance

Unlike existing approaches [9,21,23,34,53,55,56,67,73], which use all terms and all retrieved WordNet synsets of the words to represent the semantic content of a document, in our study we take two unique approaches in representing the concept-space of a document. The first approach uses the node-edge pairings of keywords and edges in the concept-forest to generate the resulting ontology. The second approach captures only the edge relationships in the ontology. By treating keywords differently according to their synset properties and the semantic relationships resolved among synsets of keywords, it can be determined the quantity of information that should be added to the ontology.

This dissertation will apply two similarity metrics, with each aiming to take advantage of different elements within the Concept-Forest. Given that the ontology is essentially a network of nodes - a graph - we will present two similarity measurements that are based on the relationships of edges and node-to-edges. Using a concept forest to represent the semantic content of a text document, the semantic similarity of two text documents can be determined by comparing their associated edges, nodes, or node-edge pairings. Each similarity measurement is explained in further detail.

Given two documents D_1 and D_2, and their concept forests $CF_1 = [T_1, E_1, R_1]$ and $CF_2 = [T_2, E_2, R_2]$ respectively (see 2.5.1 Formal Definition of Concept-Forests), the semantic similarity between the two Concept-Forests is measured by the similarity of the shared edges (Edge-Based Distance) or measured by the commonality between edges and nodes (Node-Edge Distance). The Node-Edge pairing should

prove to outperform the Edge-based method as edges do not provide adequate information to draw accurate similarity. Because edge-based parings rely solely on edges this measurement seeks to determine if the relationship between two terms exists but the not the terms. Each is defined further in the following subsections.

3.2.1 Edge-Based Distance

In the Edge-Based distance measure we seek to discover the effectiveness of determining similarity based solely on the relationships between terms across texts. Given a concept forest we define an edge-based similarity measure that attempts to measure these relationships. A relationship is measured by determining if there exists an overlap in edges between two concept-forests. We can then define the similarity as:

$$Sim(D_1, D_2) = \frac{|E_1 \cap E_2|}{|E_1 \cup E_2|}$$

Equation 2: Edge-Based Distance Formula

In this instance only the relationship between terms is compared in the concept-forest; in the event a concept-forest only contains a root-node, this root term serves as its own relationship/edge – a loop on itself.

3.2.2 Node-Edge Based Distance

The significance of the Node-Edge similarity measurement is that it is more strict then the Edge-Based measurement. This is so because in the Edge-Based measurement the algorithm only determines if the relationship is present. Moreover in the Node-Edge algorithm it is our agenda to determine if the term (node) and the sense of which it is used is present (edge). In matter of operation the Node-Edge distance formula is computed using set-relations for non-binary data; the common terms and edges are summed and divided by the total occurrences of nodes and edges in the Concept-

Forest. Node-Edge based distance is defined concretely as follows:

$$Sim(D_1, D_2) = \frac{|T_1 \cap T_2| + |E_1 \cap E_2|}{|T_1 \cup T_2| + |E_1 \cup E_2|}$$

Equation 3: Node-Edge Based Distance Formula

In [7] the Jaccard Coefficient proved to be one of the better similarity metrics to use when comparing ontology's; the Jaccard Coefficient is the intersection of two sets over its union similar to Equation 2.

3.3 Document Clustering Dataset

The document corpus is derived from the Reuters-21578 Text Categorization Collection of the University of California – Irvine Knowledge Discovery in Databases (UCI - KDD) archive. The Reuters-21578 dataset is a collection of documents that appeared on Reuter's newswire in 1987. These documents were assembled and indexed with categories. This dataset consists of approximately 21,500 files covering 132 (possibly overlapping) categories with the file size per article ranging from 12 to 1200 words.

3.4 Test Corpus

The test-collection was chosen purposely to mimic the nature of the shorter lengths of medical abstracts. Therefore, we select four sets of text document corpuses. The set of test corpuses consists of smaller sized documents containing less than 400 words (listed in Table 1) to replicate the shorter length texts of medical abstracts. The test-collection is comprised of multiple documents across multiple categories to increase the level of difficulty in text-categorization.

Table 1: Test Corpuses

Corpus	Corpus Characteristics
C-S-1	50 Documents, 5 categories, 10 documents in each category.
C-S-2	100 Documents, 10 categories 10 documents in each category.
C-S-3	300 Documents, 6 categories 50 documents in each category.
C-S-4	500 Documents, 5 categories 100 documents in each category.

3.5 Document Clustering Evaluation Method

To prove our claim that the Concept-Forest approach with the Node-Edge and Edge similarity measurements are more accurate in computing the similarity between text by resolving synonymous and polysemy relationships, a performance study based on text-categorization is presented comparing our approaches with VSM, LSI, and N*Grams.

To evaluate the effectiveness of the proposed similarity measurements we convert a set of documents into the Concept-Forest equivalent. We then cluster the resulting Concept-Forests based on their semantic similarity values calculated by *Equation 2* and *Equation 3*. The clustered results are then compared to the results of document clustering based on the keywords implementations of VSM, LSI, and Part-of-Speech tagging using N*Grams using the same datasets and the results are reviewed.

Hierarchical Agglomerative Clustering (HAC)

In performance evaluation HAC is a specific technique for unsupervised document organization used to automatically group sets of similar documents into a list of categories. Hierarchical clustering algorithms are either top-down or bottom-up. Bottom-up algorithms treat each document as a singleton cluster in the beginning and then successively merging pairs of clusters until all clusters have been merged into a single cluster that contains all documents.

Given a document corpus with N documents, each document initially belongs to its own

individual cluster. The initial similarity threshold is set to be 1 indicating that only documents that are completely similar are merged into the same cluster. The similarity threshold is then iteratively decreased by a small interval so that similar documents will gradually be merged into the same cluster. HAC ends when the similarity threshold has decreased to zero or we have the desired number of clusters.

Because the categories from which each document was obtained is known, the document clustering process is halted when majority of documents from different categories fall into their respective clusters and further decreasing the threshold will result in two clusters containing documents primarily from two different categories merged into one cluster. After document clustering, the clustering accuracy is calculated as the number of documents correctly clustered into their categories divided by the total number of documents.

3.6 Performance Results

Text document corpuses are clustered based on criteria listed in *Table 1* and are based on five different similarity measurement techniques VSM, LSI, PoS, CF-NE, and CF-E. The accuracies of text document clustering using these different methods are listed in *Table 2*.

Table 2: Clustering accuracy of VSM, LSI, PoS CF-NE, and CF-E Similarity Measurements

	VSM	LSI	PoS	CF-NE	CF-E
C-S-1	80%	68%	70%	86%	78%
C-S-2	50%	49%	45%	58%	51%
C-S-3	51%	47%	42%	57.3%	53%
C-S-4	44%	42%	36%	52%	42%

Results indicate that for each corpus C-S-*, CF-NE outperforms other methods by an average of +7% points, from the top-performer to the method with the second highest reported accuracy. On

average CF-NE performs +9.2% higher than the most used method LSI. The low score in clustering accuracy was expected for LSI. Because the test-set consisted of shorter documents, the documents did not provide enough co-occurrence information for the LSI similarity measurement to perform well.

In the experiments VSM is the second best measurement to adopt given a corpus consisting of shorter length text. The VSM measurement even outperforms CF-E. The third place ranking of CF-E is attributed to the exclusive use of edges in determining similarity. The edge represents the relationship terms shared, so CF-E is effectively determining similarity by deciding if the relationship is present in both documents being compared. Where CF-E fails is in the instance that two documents are indeed similar yet the terms in one document cannot be resolved semantically due to insufficient information – in this instance CF-E will report a false-negative.

The worse performer was Part-of-Speech tagging with the N*Grams similarity method. Because this method requires training data, a 66% split was performed on each corpus where 66% of the corpus was used in training the N*Grams model. The remaining documents were used in classification based on the training-set. The N*Grams method suffered in the performance tests due to the shorter-length of texts in the corpus as well as its inability to resolve synonymy and polysemy issues. Expectations for the N*Grams model were higher as it considers the probability of terms co-occurring in phrases or sentences in text but the result is not favorable.

Due to the positive results of CF-NE, in regard to accurately classifying shorter length documents, our experiments have shown that CF-NE results in higher accuracy when compared to VSM, LSI, and PoS Tagging using N* Grams Method for shorter length text. The successes of this experiment indicate that CF-NE should be explored further with shorter length medical abstracts. Results have also provided validation in regard to using the Concept-Forest as an alternate form of text representation for shorter length text.

The following chapter expands on the Concept-Forest by highlighting its advantages in

discovering relationships among terms and resolving polysemy occurrences in text; however this chapter also reviews the disadvantages of the Concept-Forest. Because we want to create an information retrieval system for medical abstracts and WordNet contains little semantic information on medical concepts we explore methods to increase what can be represented in WordNet by integrating supplemental medical vocabularies. The next chapter, Chapter 4, concludes the parsing linguistics component in the implementation of our information retrieval system.

Chapter 4

Proposed Concept-Forest for Medical Text

In creating an information retrieval system specifically for medical literature the system must be capable of recognizing and mapping relationships among medical terms and medical phrases. The material presented uses the WordNet Ontology in word-sense disambiguation and to resolve synonyms of terms in text written with common English terms by creating a Concept-Forest; however, the relationships that are captured by the Concept-Forest are completely dependent upon the underlying ontology. The WordNet ontology proves to be insufficient in recognizing medical concepts as it is an ontology containing common English terms.

4.1 Finding Additional Medical Vocabularies

There are two medical vocabularies that contain significant information on medical concepts that can resolve the synonymy issue of medical terms, the National Library of Medicine (NLM) Medical Subject Headings (MeSH) and the NLM Medical Thesaurus (Meta-thesaurus).

4.1.1 Medical Subject Headings (MeSH)

MeSH is the U.S. National Library of Medicine's controlled vocabulary used for indexing medical journals, articles, and abstracts in the MEDLINE database. MeSH terminology provides a consistent way to retrieve information that may use different terminology for the same concepts and imposes uniformity and consistency in the indexing of biomedical literature. It is also used in the query-parser portion of PubMed's information retrieval system to map a user's query to MeSH descriptors to then retrieve medical text that have been indexed with the same MeSH descriptor. Research [58] has shown that MeSH indexing alone in a retrieval system performs significantly worse

than free-text systems at 66% reduction of precision. Srviananas [58] also shows that Free-Text + MeSH indexing provides a slight 10.8% increase in precision using only a Free-Text indexing strategy; however, the drawback of using MeSH is in its small vocabulary and the number of medical concepts included in the MeSH hierarchy referring to medical subjects. MeSH is comprised of 27,000 MeSH descriptors and is a subset of the NLM Meta-thesaurus which contains 870,000 unique medical concepts. The MeSH controlled vocabulary represents less than 3.1% of the total available vocabulary of medical concepts. It is because of the paucity of medical concepts in MeSH and the results reported in [58] that the presented information retrieval system will leverage the NLM Meta-thesaurus to provide medical concepts and relations for concepts found in medical abstracts.

4.1.2 NLM Medical Thesaurus (Meta-Thesaurus)

The Meta-thesaurus is a database with tables listing information on medical concepts, relations (synonyms, acronyms, parent-child mappings), and numerous other relations. This special thesaurus is a large, multi-purpose, and multi-lingual vocabulary database that contains information about biomedical and health related concepts, their various names, and the relationships among them. The database holds many relationships (primarily synonymous), concept attributes, and some concept names and is updated by the NLM during annual Meta-thesaurus maintenance [53].

Of the tables present in the medical thesaurus the MRCONSO and MRREL relational tables are used. The MRCONSO table is used because it contains information to resolve synonymy problems that may arise in organizing medical text, information such as concept-names, spelling variations, and acronyms. The MRREL table contains information on inter-concept relationships.

To extend the medical concepts and medical relationships that can be represented in the Concept-Forest, Meta-Thesaurus concepts are integrated into WordNet.

Meta-Thesaurus Organization

The Meta-Thesaurus represents a concept by its associated cluster of synonymous terms and

contains approximately more than 870,000 concepts, each identified by a unique concept identifier. Each concept is accompanied by an associated set of lexical variants cumulatively numbering over 1.7 million terms with 2 million strings representing a variation in concept spelling identified by a string identifier. A depiction of concept organization as used in the Meta-thesaurus is shown in *Figure 7*.

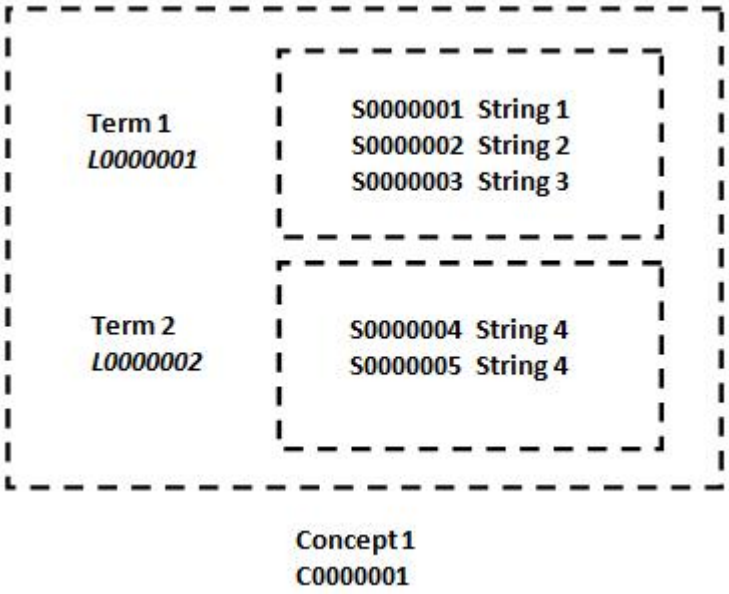


Figure 7: Meta-Thesaurus Concept Organization

A concept is a grouping of synonymous terms; furthermore, each synonymous term listed for a concept contains acceptable spelling variations. These variations are depicted as String 1….String 4, in the figure above, while the synonymous terms are depicted as Term1 and Term 2.

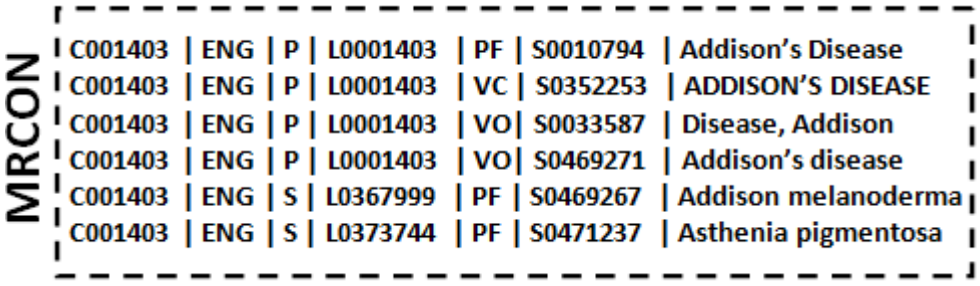


Figure 8: MRCONSO Sample Data

The MRCONSO table consists of several data columns but the two of interest are concept names and concept identifiers. The concept name may refer to a medical condition, appendages, diseases, pharmaceutical drugs, and others. The concept-name may be a single term, a phrase, or a string of terms. The concept-identifier is a unique seven character alpha-numeric string with a one-to-one mapping to the concept-name.

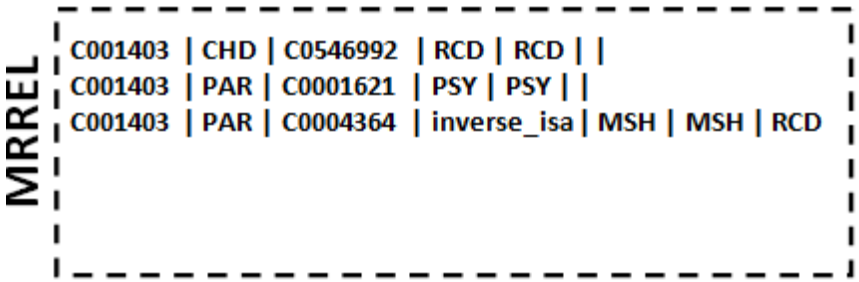


Figure 9: MRREL Sample Data

The second table used in finding relationships between medical concepts is the MRREL data table. The MRREL table is a Concept Relationship Table and captures inter-concept relationships between concepts known to the Meta-thesaurus.

4.2 Integrating Medical Vocabulary into WordNet

In using the Meta-thesaurus, the challenge is how to integrate its vocabulary into the WordNet Ontology to use its semantic relationships in constructing a Concept-Forest for medical texts. In WordNet terms are grouped into cognitive-sets. Each cognitive-set defines unique lexical relationships for its term. Because recognizing the synonym and polysemy relationships contained within a term's cognitive-set are vital components in creating the Concept-Forest, we seek to mimic the structure of WordNet's cognitive-set when retrieving medical concepts from the Meta-thesaurus.

In replicating WordNet's cognitive-set structure for a term, Meta-thesaurus concepts and their information from MRCONSO and MRREL are combined into a single logical unit. The MRCONSO

table provides spelling variations and synonyms for a medical concept, this data populates the synonyms section of a WordNet cognitive-set. To accommodate the medical inter-concept relationships from the MRREL table the cognitive-set is extended to include a medical relations component (medical_rel). Medical inter-concept relationships are then stored in the medical relations component.

Each unique concept is stored as in individual entry within the MRCONSO table (see *Figure 8*) of the Meta-Thesaurus database. We query the MRCONSO table for each term identified in the medical abstracts to retrieve additional information such as, spelling variations, synonyms, and acronyms. We use this additional informational to modify sysnset lists found for the same term or phrase in WordNet.

Once medical concepts are retrieved from querying the MRCONSO data table, each medical concept queries the MRREL data table to determine if any of the retrieved medical concepts share relationships; these relationships are stored in the extended medical relations component of the sysnset. In this table three columns are of use, the concept-identifier, the type of relationship (parent / child) and the concept-identifier it is related to.

The MRCONSO and MRREL Tables are used in discovering relationships between medical concepts that WordNet is ill-equipped to discover.

4.3 Identifying Medical Phrases, Morphology, and the SPECIALIST Lexicon

In integrating medical concepts from the Meta-Thesaurus into WordNet we must also be able to provide morphological information for medical terms to track term frequency and to ensure that we query the Meta-thesaurus tables with the base term for a medical concept. This work must also have the capability to identify medical phrases that appear in medical text as medical concepts may be not be single words but strings of words. WordNet provides a string morphing function (morph_str()) to translate common English terms to their base forms; however the Meta-thesaurus does not come pre-equipped with such methods.

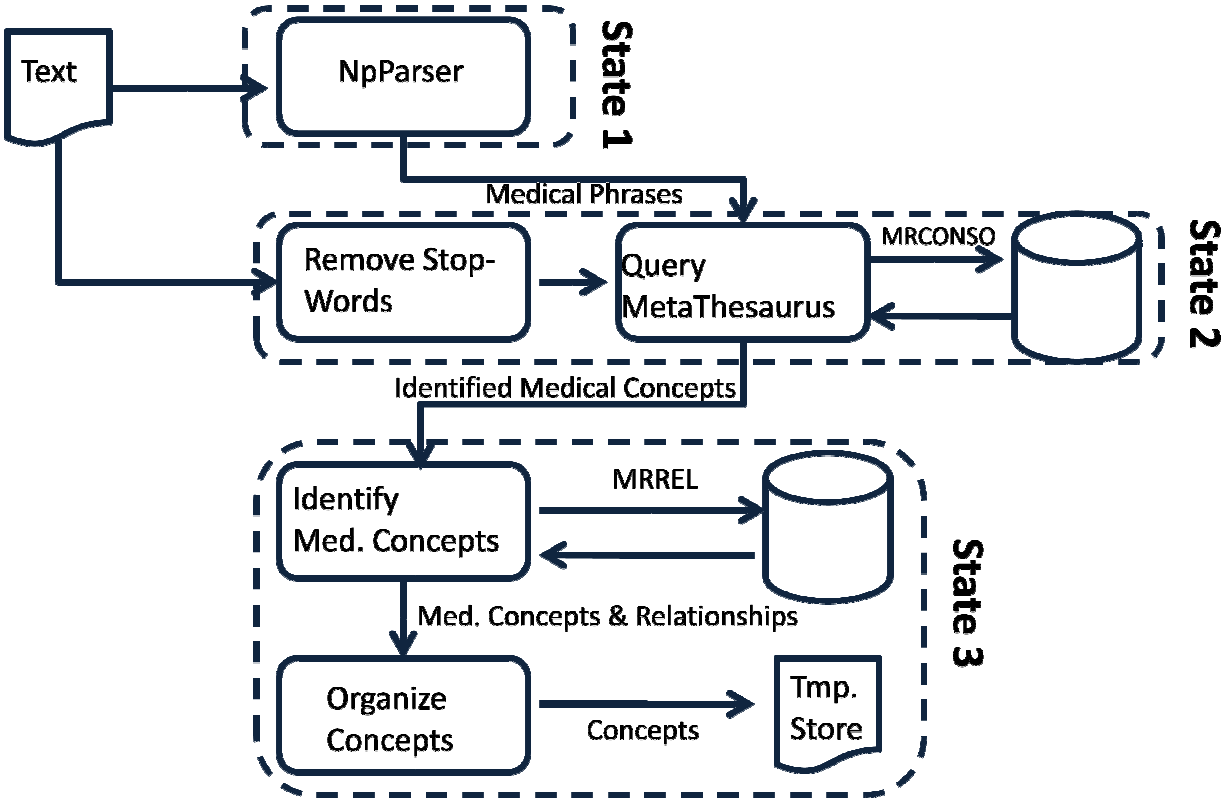
To solve the morphological challenges the NLM SPECIALIST is used. The lexicon is a large

syntactic lexicon of biomedical and general English, designed to provide the information needed for the SPECIALIST Natural Language Processing System. Coverage includes both commonly occurring English words and biomedical vocabulary. The lexicon entry for each lexical item records syntactic and morphological information for medical concepts and phrases [43]. The lexicon consists of lexical entries with one entry for each spelling variant for a particular part-of-speech.

To solve the challenges in identifying medical concepts, NLM provides open-source software available to license holders to leverage the data contained within the SPECIALIST Lexicon. Among the software provided the Noun-Phrase Parser (NpParser) is utilized in this thesis. The NpParser is a Java implementation of Thomas Rindflecsh's minimal commitment parser. The parser breaks sentences into phrases and is a minimal commitment barrier category parser. The minimal commitment analysis assigns under specified syntactic analysis to lexically analyzed input. The current emphasis is on noun phrases [43]. The proceeding section details how the NpParser and Meta-Thesaurus database tables are incorporated in constructing the Concept-Forests.

4.4 Medical Concept-Forest Algorithm

Extending the Concept-Forest so additional medical concepts and relationships can be captured will be partitioned into three stages: *Stage 1* – Identifying Medical Terms & Phrases, *Stage 2* – Retrieving Information for Medical Phrases and Terms, and *Stage 3* – Organizing Medical Concepts to be Integrated into WordNet. Figure 10 outlines the initial phases of extending the Concept-Forest.



itial Phases on Concept-Forest Extension

In the first stage each abstract passes through the NpParser to identify medical terms and phrases. Once these phrases are identified they are set-aside until the beginning of stage two.

In the second stage multiple tasks are occurring. The first item that occurs in the removal of all terms from the medical abstract that appears in the used stop-list. The second item that occurs is that each remaining term, along with the medical terms and phrases from stage one, is then submitted to the local Meta-Thesaurus to identify possible medical concepts. Identifying concepts occurs by querying the MRCONSO data table. In the event that a match occurs the concept-identifier of the matched concept is mapped to the queried medical term and they are set-aside until stage three. At the end of stage two we are left with terms and phrases that have been found by querying the MRCONSO data table.

The final stage aims to prepare the medical concepts found in stage two to be included in the Concept-Forest by emulating WordNet's structure (see *Figure 11*).

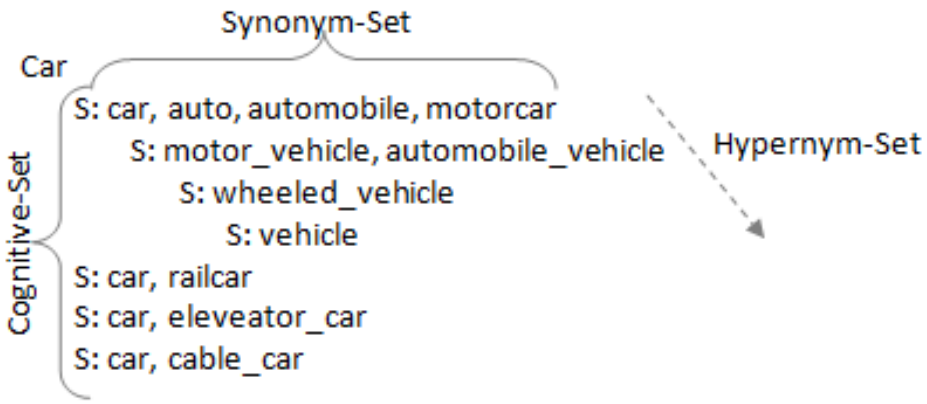


Figure 11: Structure of WordNet we seek to emulate.
S represents a Synset

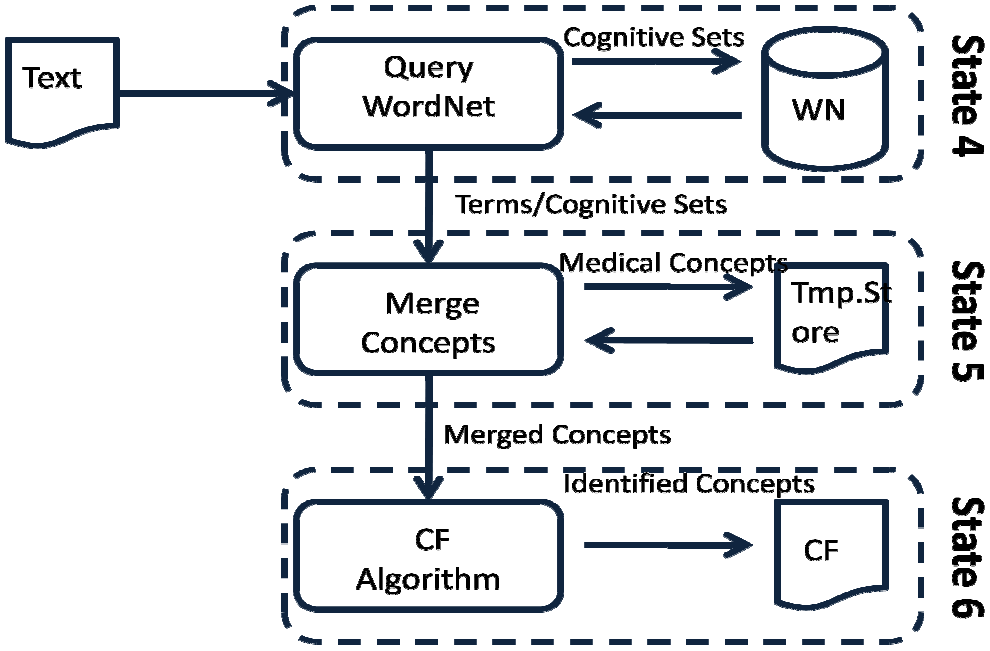
Emulating WordNet's structure allows minimum modifications to the relationship-finding portion of the Concept-Forest algorithm. The third stage is essential because it provides the "link" between two medical concepts and is brought forth by incorporating the MRREL Meta-Thesaurus data table. At this juncture we have terms mapped to medical concepts via the MRCONSO table. In this stage each concept from stage two is used to query the MRREL table to retrieve medical concepts that is related to its term. By comparing the relationships of two concepts, via MRREL, we can establish links between two concepts. Figure *12* displays a sample synset created from the MRCONSO table.

C0008049:
S: L0008049 Chickenpox
S: L0042334 Varicella
S: L6194548 Chicken Pox

Figure 12: Medical Concept Emulating WordNet Structure
S represents a Synset

Once the aforementioned steps have been performed the next step in the process can be divided into an additional three stages. The three stages can be partitioned into the following ideas: *Stage Four* – Document Preprocessing and Retrieving Cognitive-sets from WordNet, *Stage Five* – Merging Cognitive-sets within WordNet Synset Structure, and *Stage Six* – Term Sense Disambiguation and

Concept-Forest Creation. Each stage is further explained below beginning with stage four - Figure 13 outlines the remaining stages to Concept-Forest extension.



1

In stage four, Medical abstracts are stripped of formatting styles and terms are removed from the abstract that appear in the enlisted stop-list. After stop-words are removed terms are stemmed in preparation to be queried with WordNet. To conclude stage four each remaining term is sent to WordNet to retrieve cognitive-sets for noun and verb parts-of-speech. For a sample of data that can be returned per query see *Figure 11* on page 48. At the completion of stage four, terms have been mapped to cognitive-sets by WordNet and in stage five medical concepts retrieved from stage three are added to a term's concept-list.

In stage five medical concepts and terms discovered in stage three are read from disk and merged into existing cognitive-sets of terms. Because the structure of the medical concepts closely resembles that of the WordNet structure, "attaching" medical concepts to cognitive-sets is a matter of extending a

term's synset list to include medical concepts. For an illustration for a merged list see *Figure 14*.

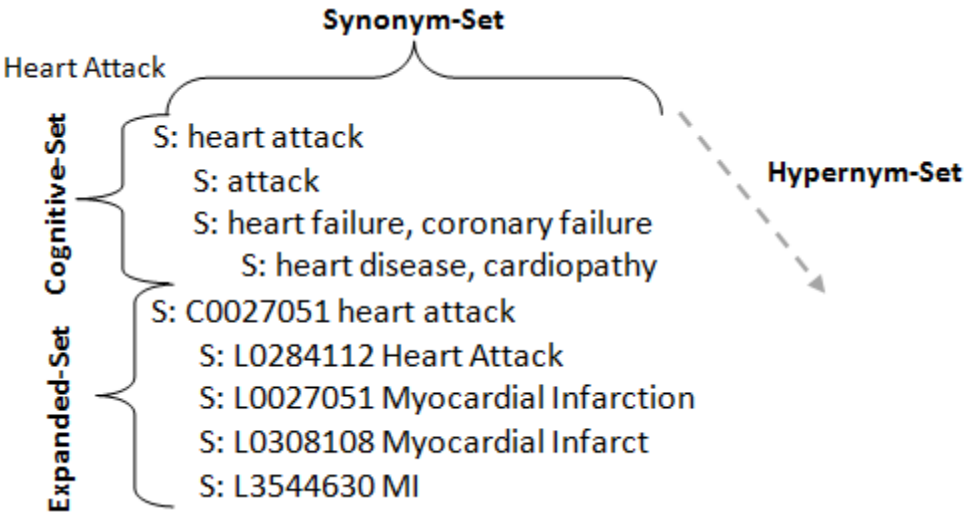


Figure 14: Merged Cognitive-Set List
S represents a Synset

Thus far we have remiss elaborating on the fate of phrases that were found in stage three. Because the phrases contain multiple words they cannot be attached to a single term unless the phrase also appears in WordNet. In the event that this occurs, as it does occur often, the medical phrase serves as a conceptually unified single term and is added to the total terms of the text. Interestingly to note, as will be explained further in upcoming sections, the medical phrase represent its own entity and will serves as a single node in the final concept-forest.

In the final stage, the concept-forest is constructed. The procedure begins by reviewing every term obtained from the text document, through stage five, to determine whether there are semantic relationships between their senses defined in WordNet and the Meta-Thesaurus. When mapping terms, given two terms (T_1 and T_2) obtained from the same text document, if their respective synsets S_1 and S_2 have a hypernym relationship, the synsetIDs of S_1 and S_2 are used to represent the concepts of T_1 and T_2 respectively, and other senses of T_1 and T_2 will be discarded. Meanwhile, an "is-a" relationship link is formed between the terms of T_1 and T_2 with the synsetID of S_1 and S_2 (which would

be identical due to S_1 and S_2 referencing a common concept). However, when mapping medical concepts if two terms T_1 and T_2 share and identical CUI then an edge is created from T_1 to T_2 with the CUI (acting as the synsetID) as the edge connecting the two. This approach may give rise to a document containing two ontology's – a WordNet induced ontology and a Meta-Thesaurus induced ontology and this is fine; however if a term T_1 shares a relationship with T_2 through WordNet and T_1 also shares a relationship with T_3 through the Meta-Concepts then the ontology's are unified into a single tree/ontology.

This process completes when all pairs of stemmed words are investigated. For instance, given a document containing words 'disease', 'sickness', 'influenza', 'drug' 'medicine', 'vaccine' and 'virus' we can construct a concept tree for terms "disease", "sickness" and "influenza" using "is-a" relationship link based on the hypernym relationship among these terms as shown in Figure 15. Similarly, a concept tree can be built for terms "drug" and "medicine".

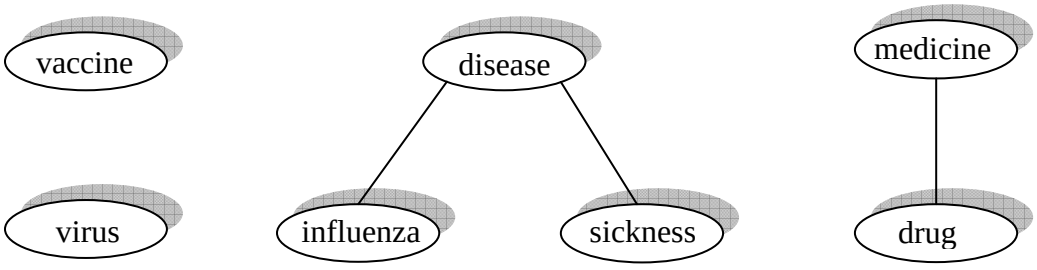


Figure 15: Sample Concept Forest

Lastly, because WordNet lacks sufficient information to find relationships between influenza, virus, and vaccine these terms are mapped with the concepts found in the Meta-Thesaurus. These three concept trees form a concept forest depicted in Figure 16 for the document; the dashed line represents the unifying of two ontology's due to 'influenza' being mapped to a term in WordNet and to a term in the Meta-Thesaurus. We note that the terms and not their related synsetIDs are shown in the concept

forest for demonstration purposes only. In actual concept forests, the synsetIDs are used to represent the edges between terms discovered utilizing WordNet, and concept-identifiers (CUI) represent the edges discovered through the Meta-Thesaurus.

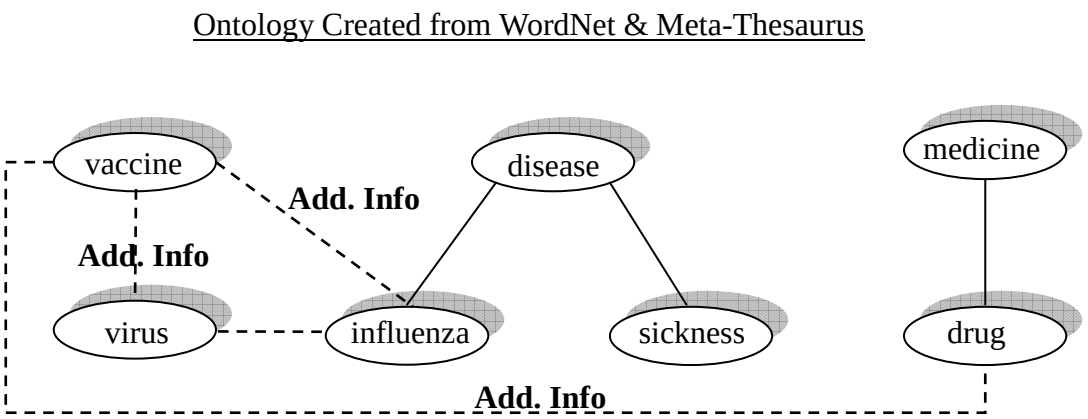


Figure 16: Sample Meta-Concept Forest

The concept-forest is now comprised of terms, phrases, relationships captured in WordNet (synsetIds), and relationships captured by the Meta-Thesaurus (concept-identifiers).

This chapter identified the shortcomings of the Concept-Forest using the WordNet Ontology. The shortcomings were the inability of the WordNet Ontology to provide synonymous or polysemus information for medical concepts. We then proposed a medical vocabulary, NLM Meta-thesaurus, to integrate into the WordNet Ontology. Integrating a medical vocabulary into the WordNet Ontology aids in resolving synonymy relationships of medical concepts while also affording the opportunity to discover relationships between two medical terms that otherwise would not be present in the WordNet Ontology. Next, the SPECIALIST Lexicon and Noun-Phrase parser (NpParser) was incorporated into building the Concept-Forest where the Lexicon provided assistance morphing a medical concept into its base form in preparation for a search in the Meta-thesaurus. The NpParser is used to identify medical concepts in medical text and prepares the discovered concepts to be queried in the Meta-

Thesaurus. Lastly, the Concept-Forest algorithm was modified to leverage the WordNet, Lexicon, NpParser, and the Meta-thesaurus.

This concludes the parsing linguistics component of the presented information retrieval system. In the following chapter the components of the concept-forest are incorporated into the design of the remaining phases of the presented information retrieval system, these phases are: indexing, query-parsers, and scoring & ranking. Each remaining phase of the retrieval system is presented further in the next chapter.

Chapter 5

Proposed Information Retrieval System

Thus far Chapter 1 through Chapter 4 has been devoted to the Parsing Linguistics phase in the presented information retrieval system, see *Figure 17*. In this chapter, the remaining phases of the retrieval system that are to be reviewed consist of four main components: updating stop-lists, creating the index, choosing a ranking function, and implementing a query-parser.

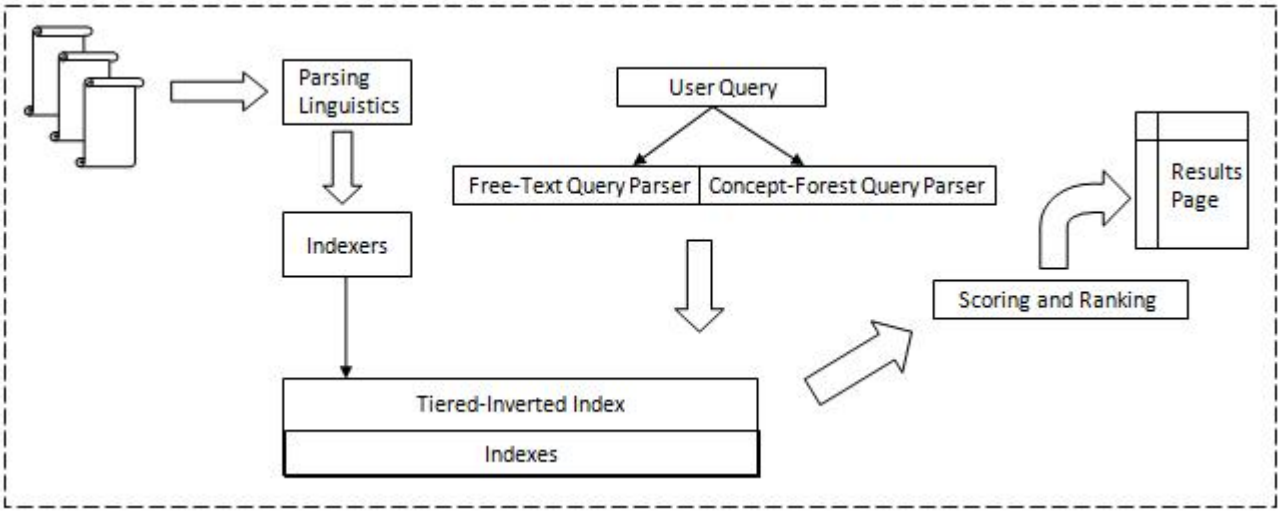


Figure 17: Information Retrieval System Design

5.1 Updating Stop-lists

A stop-list is a list of words that should be filtered or removed prior to processing medical abstracts or queries. Words in a stop-list are common adjectives, pronouns, and conjugations. The stop-lists consist of a predefined set of terms; the employed stop-list is one that is often used in retrieval and contains terms that are less likely to add significant semantic information to text. In addition the employed stop-list which is devoid of medical terminology is increased to include all terms/synonym identifiers/meta-thesaurus concepts that appear within more than 10% of the collection; this allows a custom tailored stop-list more representative of the medical vocabulary in the collection-set. The

thought behind updating the stop-list is that every corpus vocabulary is unique. It can be the case that there are a set of terms that do not appear in a standard stop-list but contain high prevalence throughout a collection. Because of the high occurrence frequency of this set of terms in context with the size of the collection-set, we deem that this set of terms are not ideal discriminators delineating differing medical abstracts and therefore this set of terms adds little additional semantic information to the corpus.

5.2 Creating an Inverted Index

Within the document collection each medical abstract is represented by its Pubmed unique document identifier. The PubMed identifier is an alphanumeric string consisting of seven characters that are assigned to a medical abstract when the abstract is added to the MEDLINE data repository. An inverted-index is an index or listing into a set of abstracts of the words that appear in the abstracts [38]. Each index entry gives the word and a list of abstracts in which the word occurs.

For each term in the document collection there is a list that records that a term appeared in an abstract. Each item in the list is referred to as an index. The complete list is conventionally called an inverted-list or inverted index. A sample inverted-index is illustrated in *Figure 18*.

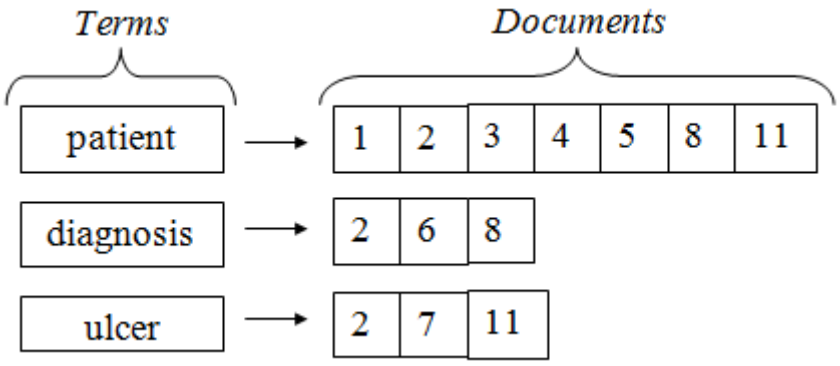


Figure 18: Example of a Posting-List/Inverted-Index

In indexing a medical abstract only the vocabulary of the abstract is used is the indexing process. When indexing a Concept-Forest more information is indexed. Because the Concept-Forest is comprised of the original vocabulary of the medical abstract and the relationships discovered between terms, each should be indexed. When indexing a Concept-Forest, the original terms that appear in the abstract are indexed along with synonyms for terms where the word sense has been resolved; the sense in which the term was used has to have been resolved as this is to prevent adding synonyms for terms for incorrect senses. Relationships between terms captured by the Concept-Forest are also indexed via the WordNet synset and Meta-thesaurus identifiers.

To gain the speed benefits of indexing at retrieval time, the index must be constructed in advance. There are four major steps in building the inverted index. These steps are:

1. Collect documents to be indexed
2. Tokenize text of document
3. Perform linguistic preprocessing, leading to a list of normalized tokens
4. Index the documents that each term appears in by creating an inverted index

As of Chapter 4, steps one through three have been performed. However, before proceeding to step four, in preparation of building the index each term in the medical abstract and Concept-Forest is sent through a porter stemmer to further alleviate inconsistencies in spelling variants. The Porter Stemmer is a process for removing the commoner morphological and inflectional endings from words in English.

5.3 Term Weighting and Document Ranking - TF-IDF and Okapi BM25

Using the Concept-Forest as an index means that we have a collection of Concept-Forest representations of medical abstracts in a central repository. Because the vocabulary is accessible we

can apply corpus wide statistical measures to heighten term and relationship importance. We want to use collection-wide statistical measures as opposed to using raw term frequency because using raw term frequency suffers from the problem of treating all terms in the collection equally when it comes to assessing relevancy on a query.

The term weighting and document ranking function used are the term-frequency - inverse document frequency (TF-IDF) measure and Okapi BM25 ranking function that has roots in probability and statistics. The TF-IDF is a statistical measure used to evaluate the importance of a term in a document in a collection. The weight of the term increases proportionally to the frequency in which the term occurs in a document but is offset by the frequency of the word relative to the complete test collection. TF-IDF assigns to term t a weight in document d that is: highest when t occurs many times within a small number of documents, lower when the term occurs fewer times in a document or the term occurs in many documents, and lowest when the term occurs in most documents in the collection [38]. TF-IDF was chosen due to the frequency in which it is used in information retrieval and text mining.

To construct the IR system we adopted many of the techniques used in the popular SMART retrieval system w/ a few additions. Each term, WordNet identifier, and Meta-thesaurus concept in each Concept-Forest is used in creating an inverted index. Secondly we employ the term-frequency-inverse document frequency (TF-IDF) as a term discriminator to gauge the similarity between an information request (user-query) and a document. Due to variations in TF-IDF we define the metric used further: term-frequency is computed by:

$$tf - idf_{t,d} = tf_{t,d} * idf_t$$

Equation 4: Term-Frequency Inverse Document Frequency Formula

where $tf_{t,d}$ is the term-frequency of a term t in document d, and idf is the natural term frequency where IDF is defined as:

$$idf = \log(\frac{N}{df_t})$$

Equation 5: Defining the Inverse Document Frequency

as the inverse document frequency where N, is the total documents contained within the collection and df_t is the quantity of documents that contain term t. Using TF-IDF, the score of a document-query match is defined as:

$$score(D,Q) = \sum_{t \in Q} tf - idf_{t,d}$$

Equation 6: Document Scoring Function

The Okapi BM25 weighting function was chosen for its wide and extensive use successfully across a range of collections and search tasks notably with the TREC evaluations. It was developed as a way to building a probabilistic model sensitive to term-frequency and document length. This algorithm models term-frequency and document length in the equation by introducing two free parameters k_1 and b. The k_1 parameter is a positive tuning variable that calibrates the document term frequency scaling; a value of zero corresponds to a binary model – either the word appears or it does not. A large value, greater than two, for the k_1 parameter indicates the use of raw term frequency. The second parameter b, (for 0 <= b <= 1), determines how to scale the document length. A value of 0 corresponds to no length normalization, while a value of 1 corresponds to completely scaling the term weight by the document length. It is reported that the preferred values for a system unconcerned with optimizing the parameters are 2 and .75 for k_1 and b respectively [38, 39]

The research ranking function Okapi BM25 [39], where the score for a document-query pair is defined

as:

$$score(D,Q) = \sum_{i=1}^{n} IDF(q_i) * \frac{f(q_i, D) * (k_1 + 1)}{f(q_i, D) + k_1 * (1 - b + b * \frac{|D|}{avgdl})}$$

Equation 7: Okapi BM25 Ranking Formula

where N is the total documents in the collection; k_1 and b are free parameters 2.0 and .75 respectively; avgdl is the average document length, and |D| is the length of the document in words. $f(q_i,D)$ is the term frequency of q_i in document D. And IDF(q_i) defined as:

$$IDF(q_i) = \log \frac{N - n(q_i) + .5}{n(q_i) + .5}$$

Equation 8: Defining the IDF for Okapi BM25

Where N is the total documents in the collection and $n(q_i)$ is the number of documents containing term q_i. The .5 is for terms that appear in over half of the collection-set; if a term occurs in over half of the documents in the collection the equation assigns a negative term weight [38]. Assigning negative term weights is unusual when retrieval systems employ a stop-list. Both TF-IDF and BM25 are used in evaluation.

5.4 Query Parsers

Query parsers resolve query terms to indexes to then retrieve documents that have been indexed with the same indexes. There are two major classes of query-parsers: global methods and local methods [17]. Global methods modify the users query before the query is submitted. Global methods are techniques for reformulating or expanding query terms. Popular techniques include spelling correction, and query expansion via the incorporation of automatic thesaurus generation or WordNet [17]. Local methods entail the user submitting a query, analyzing the top k documents

retrieved from the query, and then augmenting and resubmitting the query– usually transparent to the user. The basic methods of local methods include relevance feedback [7,36], pseudo-relevance feedback [32], and indirect relevance feedback. For the focus of this study global query methods are used.

In traditional information retrieval systems that use keyword indices or controlled vocabularies search-text ambiguities are ever-present. Because query search terms tend to be very short, from 2 to 12 words, it becomes important to find methods to exploit as much information as allowable from the limited query text, e.g. expanding search terms to include its synonyms, hypernyms, meronyms, etc.. Finding relationships between co-occurring terms in queries may also provide to be beneficial in retrieval. Query-parsers seek to facilitate the translation of a users search terms to match those of the retrieval systems indexing strategy. The indexing strategies used by the proposed retrieval system are: Free-Text Indexing, Meta-Thesaurus Indexing, and Concept-Forest Indexing.

The three query indexing strategies adopted in this research fall under global methods in that the strategies that will be presented modify the query before the query is used to retrieve results also called query expansion.

5.4.1 Free-Text Query and Free-Text Document Indexing

In free-text query indexing, the query is preprocessed to identify individual terms that are not in a stop list and then undergo porter stemming before results are retrieved. In free-text document indexing, each document in the corpora is preprocessed to remove terms that appear in a stop-list, the remaining terms then undergo porter-stemming, and the an inverted index is created for the document in the corpora.

5.4.2 Concept-Forest Query and Concept-Forest Document Indexing

In Concept-Forest query indexing strategy the user’s query is expanded to include more semantically relevant terms. Initially, a Concept-Forest is constructed from the original query terms and

the resulting Concept-Forest and all found terms are sent through a porter stemmer. After stemming the query is sent to retrieve search results. In Concept-Forest document indexing, a Concept-Forest is created for each document in the corpora. Then the terms in the Concept-Forest undergo porter-stemming, and then an inverted index is created on the Concept-Forest corpora for stemmed terms and edges.

5.4.3 Meta-thesaurus Query and Meta-thesaurus Document Indexing

In Meta-thesaurus query indexing, the query is preprocessed to identify medical terms and medical phrases. The terms and phrases identified are then queried by the Meta-Thesaurus to retrieve medical concept identifiers. The medical concept identifiers are then sent to retrieve search results. In Meta-thesaurus document indexing, each document in the corpora is preprocessed to identify medical concepts and medical phrases. Medical concept identifiers are then retrieved for the concepts and phrases and an inverted index is created for each document based on the found medical concept identifiers in the corpora.

This chapter detailed the remaining components of the proposed information retrieval system. Firstly, a data structure is needed to represent text and the terms that appear in the text. To address this issue an inverted index is used to record in a listing the vocabulary of the collection-set and the medical abstracts that each term in the vocabulary appears in. Secondly, an option for collection-wide statistical measures, TF-IDF or Okapi BM25, is presented to be used in the retrieval system. Both measures are used to assign weights to terms in the collection-set to distinguish among important terms and are based on a function of term-frequency. Lastly, three query-parsing strategies are explored: traditional Free-Text, Meta-thesaurus Concept, and Concept-Forest.

This concludes the design decisions chosen in proposed information retrieval system. The following chapter, Chapter 6, delineates the way in which the IR Systems will be compared, performance metrics used and the implementation of the MeSH IR system. Chapter 6 also includes

experiments to test the effectiveness of our ontology indexed IR system to that of the MeSH indexed IR system.

Chapter 6

Comparing Information Retrieval Systems

A performance study is conducted on four differing retrieval systems based on representing medical abstracts as: Free-Text, Free-Text with a controlled vocabulary, controlled vocabulary only, and Ontology. Each retrieval system brings forth a unique approach to representing and indexing medical abstracts for MEDLINE. The approach taken to represent and index text in an important component in the design of an information retrieval system such that each retrieval system is reviewed before the performance study begins.

Free-Text retrieval systems are the most prevalent search systems on the web. This is primarily so because the ease at which this type of system can be deployed. For example the popular open-source MySQL relational database is distributed with indexing logic pre-installed; all one needs is to provide the document copora and a Free-Text system can be created in minutes. These types of systems rely on keyword-based indices; the terms that appear in documents are the terms used to retrieve them. Using keyword-indices is a syntax-based form of retrieval that disregards semantically related documents that do not share the same vocabulary as the user's search query (a problem of synonymy). These systems also provide results that may be inconsistent with user expectations as they do not consider the context in which terms are used in the search query (a problem of polysemy).

To combat the problem of polysemy and synonymy specialized retrieval systems have began to incorporate the inclusion of controlled vocabularies. Free-Tex t+ controlled vocabulary retrieval systems are as good as the maintained vocabulary. These systems index document corpora with a controlled vocabulary and terms appearing in text. Document retrieval is dependent upon users being

knowledgeable of the vocabulary when formulating queries for effective searches. Because vocabularies are constantly growing, retrieval systems such as the NLM PubMed, create automated methods to map user-queries to vocabulary indexes for document retrieval; however, due to the natural evolution of language, controlled vocabularies are rarely complete. These systems also share problems of synonymy and polysemy in cases where every synonym instance isn't captured and when every sense of a term's usage isn't captured in the controlled vocabulary. The Free-Text + controlled-vocabulary based system is the Free-Text + MeSH retrieval system that will be presented later in the chapter. The controlled-vocabulary only retrieval system is similar to the free-text + controlled vocabulary system except that terms in a document are not included in the indexing process.

Ontology based retrieval systems resolve synonym and polysemy problems that Free-Text and controlled vocabulary systems are not equipped to address. Because ontology's concretely define terms and relationships between terms, indexing documents by the ontology are growing in interest; however unlike Free-Text retrieval systems an ontology based system isn't a straight-forward process to implement. The advantage of this type of system is that keywords and any resolved relationships between terms (synonymous or polysemus) are indexed for a document. Terms in a users search query are automatically expanded to include synonymous relationships if the context of the search query can be resolved. Other relationships found in a user's search-query are also expanded and included in retrieval. This type of retrieval system is only as good as the constructed ontology which it leverages. The ontology-based system is the Concept-Forest retrieval system presented in this dissertation.

The remaining sections of this chapter detail the implementations of the MeSH retrieval system, the metrics used that allow the comparison of disparate retrieval systems, the OHSUMED Data Collection-set used in experiments, and a performance analysis of each type of retrieval system presented.

6.1 Free-Text + Medical Subject Headings (MeSH) Retrieval System

Thus far the implementation details of the ontology based (Concept-Forest) information retrieval system has been review along with Free-Text based retrieval systems. Attention will now be given on the design details of the MeSH retreival system. Srinivasan [58] used the SMART Information Retreival system to conduct IR experiments using MeSH, Free-Text, and combinations for indexing. The results of the experiements were that MeSH +Free-Text indexing outperformed Free-Text only indexing strategies. In an effort to recreate these successes to allow a comparative study for our own methods, we employed the use of the SMART system. We were unable to reproduce these results due to severe installation troubles, outdated documentation, and antiquated mailing-lists; however [58] concisely outlined the methodology and this is what was followed and will be presented.

6.1.1 Free Text + MeSH Query and Document Indexing

In Free-Text + MeSH query-indexing MeSH descriptors are retrieved for terms appearing in the search-query. Then, the Free-Text is processed to remove terms that appear in a stop-list and undergo porter stemming. Together the original stemmed query and any MeSH concepts found are resubmitted for retrieval. In Free-Text + MeSH document indexing, each document in the corpora is preprocessed to identify MeSH descriptors and all Free-Text terms are stemmed via the porter stemmer. An inverted index is created for each document in the corpora based on the found MeSH descriptors and Free-Text stemmed terms.

6.1.2 Term Weight Assignment

The term-weighting schemed used in [26] was one of three parts: the term frequency component, the inverse document frequency component and the normalization component. The formula is given below:

$$(.5 + .5 * \frac{tf}{\max\_tf\_in\_text}) * \ln(\frac{N}{n})$$

Equation 9: SMART Term Weighting Formula

Where N is the number of documents in the collection-set, *n* is the number of documents with the term in question, and *tf* is the frequency of the term in the document. The .5 is for assigning terms appearing in more than half of the corpus a negative weight.

6.1.3 Free Text + MeSH Scoring Function

Retrieval is conducted by computing the similarity between corresponding query and document-indexing vectors. When there are two index vectors for a document and a query, similarity between the document and query is computed by taking the cosine-similarity of the free-text vectors and added to the cosine similarity of the MeSH vectors. The similarity is computed as follows:

$$Sim(D, Q) = delta * sim(FT\_vectors) + sim(MeSH\_vectors)$$

Equation 10: Similarity Measurement used in SMART

The parameter delta is varied over the vales 2, 1.75, 1.5, 1.3, 1, 0.8, 0.66, and .57 where the delta represents the relative emphases placed on the two types of vectors. These similarities values are used to rank the items that will be shown to the user.

6.2 Performance Indicators

The two most frequent and basic measures for judging information retrieval effectiveness are precision and recall. Precision and recall are set-based measures and are used in the evaluation of unranked text classification. These measures are computed using unordered sets of documents. Firstly we define recall and then precision:

Recall (R) is the fraction of relevant documents that are retrieved. Defined as:

$$\text{Recall} = \frac{\#\text{ (relevant items retrieved)}}{\#\text{ (relevant items)}} = P(\text{retrieved} \mid \text{relevant})$$

Equation 11: Recall Computation

Precision (P) is the fraction of retrieved documents that are relevant. Defined as:

$$\text{Precision} = \frac{\#\text{ (relevant items retrieved)}}{\#\text{ (retrieved items)}} = P(\text{relevant} \mid \text{retrieved})$$

Equation 12: Precision Computation

To evaluate the ranked retrieval results these measures need to be extended. In ranked retrieval, appropriate sets of retrieved documents are given by the top kth (10%, 20% ….100%) retrieved documents. For each such set precision and recall values can be computed. The precision and recall values computed can then be plotted to give a precision-recall curve. The precision-recall curve tends to be saw-toothed. In performance reporting the interpolated precision (p_{interp}) is used to remove the jaggedness and smooth the curve. In interpolated precision, at a certain recall level r is defined as the highest precision reported for any recall r` >= r:

$$P_{interp}(r) = \max_{r'>=r} p(r')$$

Equation 13: Interpolated Precision Equation

The justification behind interpolated precision is that users would be prepared to search through more documents if it would increase the percentage of the viewed set were relevant.

The traditional method to narrow down this information into one indicator is to use the eleven-point interpolated average precision. For each query or information-need, the interpolated precision is measured at the eleven standard recall levels of 0.0, 0.1, 0.2.., and 1.0; a recall levels corresponds to a percentage of the results retrieved and ranked. Then for each recall-level we then calculate the arithmetic mean of the interpolated precision at that recall level for each information need in the test collection [38]. The eleven-point average precision then becomes the average of the eleven standard

recall points.

6.3 Experiment Dataset - OHSUMED

To facilitate comparison of our findings with researchers in an identical area of research the OHSUMED collection is used in testing. OHSUMED is a clinically-oriented MEDLINE subset, consisting of 348,566 references covering all references from 270 medical journals over a five-year period (1987-1991). In creating the OHSUMED dataset novice physicians using MEDLINE generated 106 queries. Physicians were asked to provide a statement of information about their patients as well as their information need (query). Each query was later replicated by four searchers, two physicians experienced in searching and two medical librarians. The results were assessed for relevance by a different group of physicians, using a three point scale: definitely, possibly, or not relevant. OHSUMED is the collection of these relevant judgments.

This dataset has been extensively utilized [16,19,50,57,58] to carry-out indexing experiments. Also [58] uses this collection set and has judged the effectiveness of free-text queries to MeSH indexed documents/queries and uses the SMART retrieval system to report findings. Srinivasan, [58] stops short of comparing Free-Text + MeSH document indexes to an indexing strategy that utilizes the UMLS Meta-thesaurus and/or WordNet.

Design of Experiment

In these experiments the performance of a system is measured through the use of the 11-point average precision measure. Four information retrieval systems are compared based on their indexing and retrieval strategies. The systems being compared are: Free-Text, Free-Text + MeSH, Meta-Thesaurus, and Concept-Forest. A diagram of the experiment outline is depicted in Figure *19*

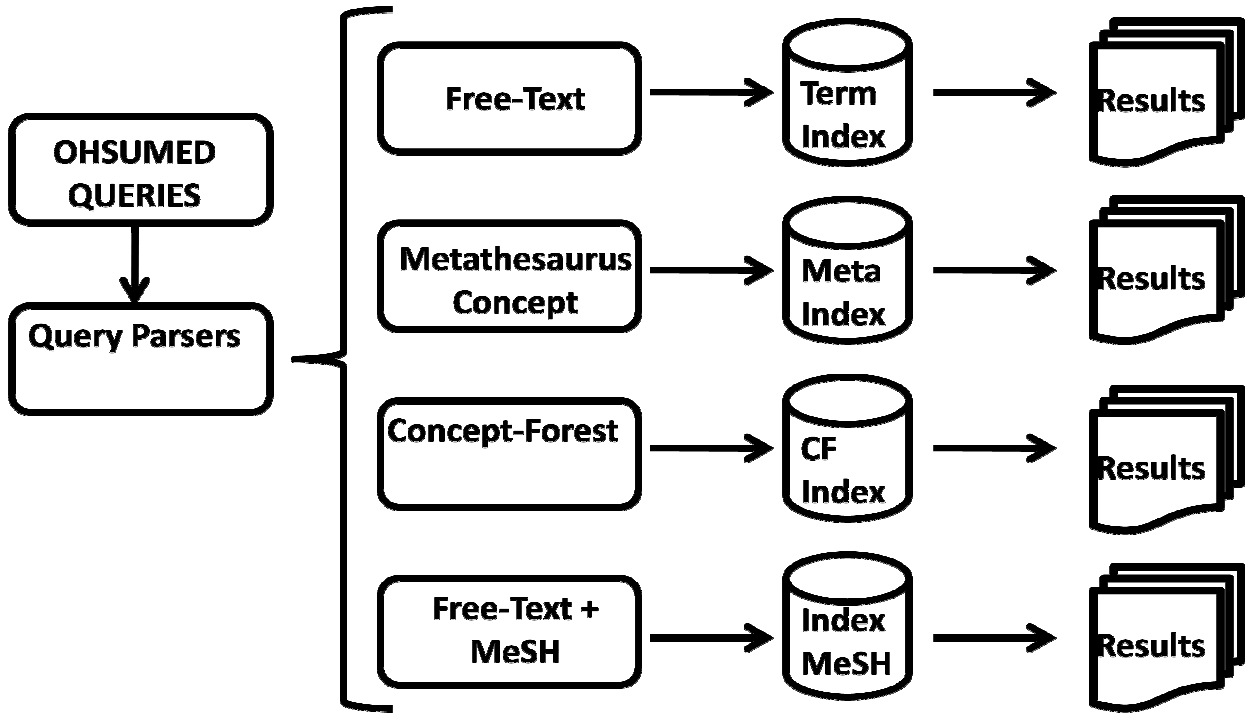


Figure 19: Experiment Design

Firstly, we have defined four retrieval strategies Free-Text, Free-Text + MeSH, Meta-Concepts, and Concept-Forest. The experiment will be conducted as follows:

I. For each Retrieval Strategy we issue the 106 queries

II. After the results from each query is returned, the recall, precision, and interpolated precision is calculated at the 11-pt standard recall levels

III. The mean is computed for the interpolated precision among all 106 queries to yield the 11-pt average precision of the IR System

IV. The 11-pt avg precision of each system is then reported

We then compare the results of all four retrieval strategies to determine the better IR-System.

Three tests were conducted using our developed IR-System. The tests employed the two dominate ranking functions Okapi BM25 and TF-IDF. The three tests compare free-text document indexing with free-text query indexing, Meta-thesaurus document-indexing with Meta-thesaurus query indexing, and Concept-Forest document indexing with Concept-Forest query indexing. We then compare these tests to Free Text + MeSH document indexing with Free Text + MeSH query indexing, as shown in *Table 3*.

Table 3: Retrieval Strategies

Retrieval Strategy	Document Indexing	Query Indexing
RS-D-1	Free-Text	Free-Text
RS-D-2	Meta-Concepts	Meta-Concepts
RS-D-3	Concept-Forest	Concept-Forest
RS-S-4	Free Text + MeSH	Free Text + MeSH

6.4 Performance Results

6. 4.1 Results of Developed IR-System

In configuring the IR-system we have tested three configurations. The first configuration expresses an information need by a free-text query while using a free-text inverted index and TF-IDF as the scroing function. The second configuration expresses the information need through the use of the Concept-Forest and uses a Concept-Forest inverted index and TF-IDF scoring function. The last confiugation is identical to the second configuration but replaces the scoring function with Okapi BM25. Results using the OHSUMED dataset is provided in *Figure 20* and *Figure 21*.

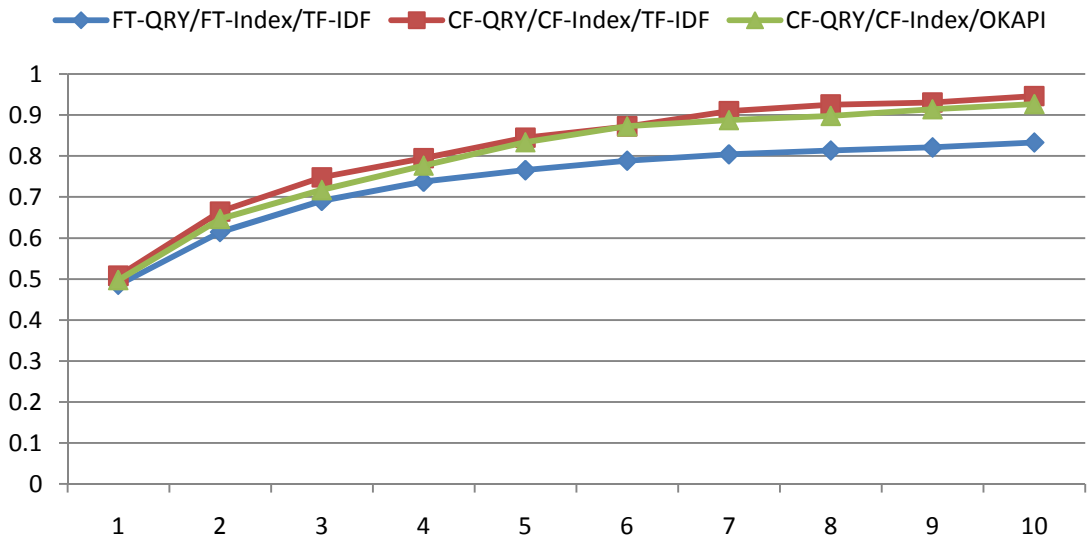


Figure 20: Recall@k for Developed IR-System using OSHUMED

Figure 20 displayes the recall @ k, where k represents the percentage of documents retrieved. *Figure 20* shows a dominant performance in recall of our IR-system when using the CF-Query/CF-Index/TF-IDF configuration – this configuation achieves a recall of greater than 90% of documents retrtieved. The Okapi BM25 is a close second, also showing a recall of greater than 90%; however the worst performer with this dataset is using the FT-Query/FT-index/TF-IDF configuation, reporting a recall of approximately 82%. The difference between the top performer and the worst is more then ten percentage points.

The large disparity among the top-performer (CF) and the worst performer (Free-Text) in recall is due to the retrieval strategy. As stated previously, the Free-Text retrieval system is completely dependent upon the query continaing the terms in the documents that will be retrrieeved. These systems retreival is entirely dependent on the overlap between query terms and terms in documents. This is not to say that Free-Text systems contain sets of documents that are irretreivable, however, these systesms may require a user to perform multiple searches varying the vocabulary in each query to find the desired documet.

The CF retrieval system expands query terms to include corresponding synonym sets and any relationships found in the query, e.g. relationships found in WordNet or medical relationships found in the Meta-Thesaurus. Because the Concept-Forest retreival system resovles sysnonyms and contextual relationships of terms in a query for the user, results show an increase in recall by +10% percentage points. These resuts translate into a reducution in search efffort by the user requesting a document by retreiving more relevant documents for a user's initial query.

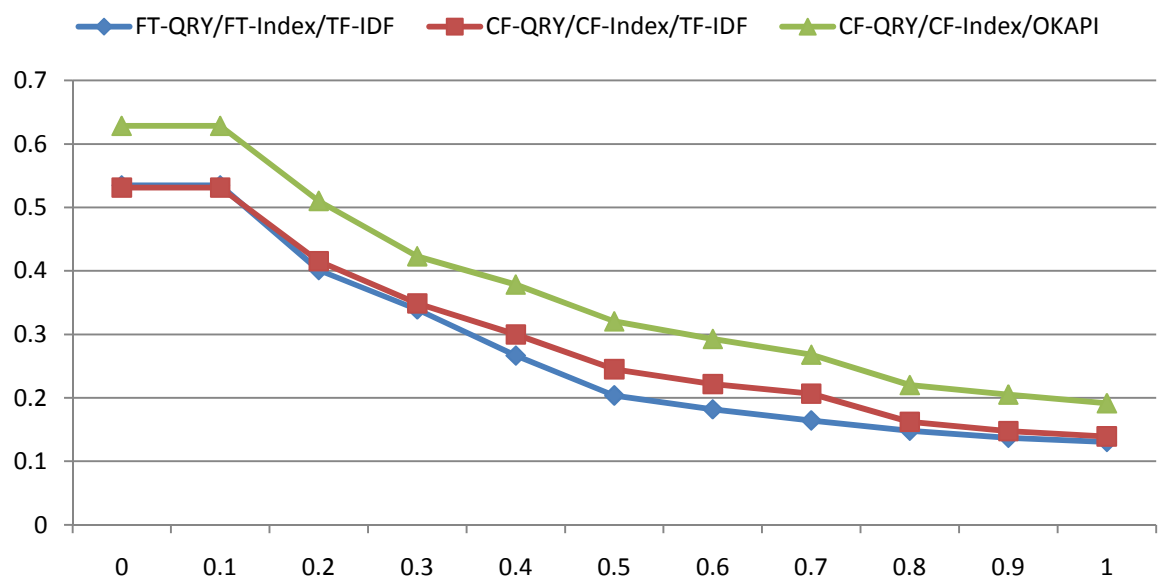


Figure 21: Precision-Recall Curve on OHSUMED Dataset

Figure 21 depicts the precision-recall curve using interopolated precision. Precision is essentially, how much "junk" or irreleveant documents are retrieved for an information-need. From *Figure 21* the CF-Query/CF-Index/Okapi configuation serves as the best filter with an 11-point average precision of 31%. Followed by its TF-IDF counterpart and worst performer, FT-Query/FT-Index/TF-IDF with a score of 26% and 24% respectively.

In studying the performance of Free-Text retrieval systems in regard to their reported precision, anaylsis has found that because the context of the query is not determined, each sense a term can be mapped to is retrieved simply because it occurs in the document; this type of retrieval leads to many retrieved documents but few releant ones.

The Concept-Forest retrieval systems offer much higher discriminating power between documents because of the inclusion of a terms synonyms, but more importnatly because of the Concept-Forests capability to resolve query terms context. The ability to resolve the context in which a term is used in a query in combination with resovling the context of terms appeariong in medical abstracts in the document corpora has proven to have merit and is apparent in the favorable performance in precision by retreiving less irreleveant documents than Free-Text retreival systems.

Results from both figures lead to the conclusion that resolving a query terms synonmys lead to an increase in recall while resovling the context of the terms in a query lead to higher precision. *Table 4* shows a more detailed view of the precision reported at the 11-standard recall levels for RS-D-* on our developed IR System. Table *4* shows the Interpolated precision reported for each recall level for each scoring method, TF-IDF and Okapi BM25.

Table 4: 11pt-Average Precision across Tested Sets

11-Standad Recall Levels	Okapi BM25			TF-IDF			Interpolated Precision
	RS-D-1	RS-D-2	RS-D-3	RS-D-1	RS-D-2	RS-D-3	
0	.6336	.3066	.6282	.5345	.3066	.5451	
.1	.6336	.3066	.6282	.5345	.3066	.5451	
.2	.5058	.2662	.5101	.4012	.2662	.4165	
.3	.4176	.2455	.4229	.3392	.2455	.3476	
.4	.3437	.2355	.3784	.2663	.2355	.2707	
.5	.2832	.1963	.3204	.2035	.1963	.2107	
.6	.2507	.1579	.2927	.1817	.1579	.1862	
.7	.2275	.1199	.2678	.1641	.1199	.1676	
.8	.2106	.1132	.2199	.1477	.1132	.1530	
.9	.1975	.1072	.2050	.1366	.1072	.1423	
1	.1867	.1094	.1914	.1302	.1094	.1351	
11pt Avg	**.2611**	**.1967**	***.3124***	**.2271**	**.1689**	***.2716***	

Table 4 is consistent with results reported by other researchers regarding RS-D-1/TF-IDF; for Free Text indexing and querying, results are between .22 and .24. However, RS-D-3/OKAPI results in an 11-pt average precision of .3124 – this is a 37.5% increase over free-text queries. RS-D-3/TK-IDF show a 19.5% increase over the lower scoring RS-D-1 strategy. Also RS-D-3/OKAPI outperforms RS-D-3/TF-IDF by 15%. RS-D-3/OKAPI success is attributed to the Concept-Forest representation of medical abstracts. The Concept-Forest alternative method to text-representation has lead to an increase in recall due to its method of word-sense disambiguation for including correct synonyms(terms) and its methods at discovering term context has lead to increased precision by including contextual relationships (edges) of the Concept-Forest. These preliminary results give merit to the Concept-Forest based index. The results listed above will be the standard when comparing the

developed Concept-Forest index information retreival system to a Medical Subject Headings controlled vocabualry and Free-Text information retrieval systems.

6.4.2 Results Concept-Forest System compared to Free-Text + MeSH System

Compared to the results of RS-S-4 which uses Free-Text + MeSH query and document indexing strategies and an altered TF-IDF scoring function on the OHSUMED dataset based on the 11-pt average precision statistics the presented IR system is an overall better system as reported. See *Table 5*.

Retrieval Strategy	11-pt Avg Prec
RS-D-1 / OKAPI	.2611
RS-D-2 / OKAPI	.1967
RS-D-3 / OKAPI	**.3124**
RS-D-1 / TF-IDF	.2271
RS-D-2 / TF-IDF	.1689
RS-D-3 / TF-IDF	.2716
RS-S-4 / Free Text + MeSH	.2819

Table 5: 11pt-Avg. Prec Comparison of
Concept Forest (CF) to results in
[1] OHSUMED Dataset

From the work done in [58] it can be substantiated that a Free-Text and MeSH IR System has its merits. Also as reported, the Free-Text and MeSH solution reports a two percent increase in 11-pt average precision when compared to a Free-Text only configuration. Results further show a Meta-thesaurus-only indexing solution is improbable due to being the lowest score present.

In recreating [58] the most notable change was in the size of the MeSH concepts that were available during the administration of the test when initially performed. When [26] initially performed the experiment there were approximately 14,000 MeSH concepts available. Presently, there are approximately 27,000 MeSH concepts.

From the results it can be confirmed and substantiated that a Free-Text and MeSH IR System has its merits. Also as reported, the Free-Text and MeSH solution reports a 7.9% increase in 11-pt

average precision when compared to our Free-Text only configuration and a 7.8% increase from results reported in [58] which were recorded to be .2614; the 7.8% increase in attributed to the increase in the size of the MeSH controlled vocabulary from 14K to 27K.

Using the Concept-Forest, the IR system retrieves a 19.6% increase in 11-pt average precision over the results reported for RS-D-1 and RS-D-2 solutions using the Okapi BM25 scoring function. The IR system scores a 19.5% increase using the TF-IDF scoring function over RS-D-1 solutions. With the composed IR system our Concept-Forest based approach performs better than free-text and MeSH document indexing represented by a 10.8% increase in 11-pt average precision. Research results also indicate that using the Meta-thesaurus concepts only to index documents is unreliable as a standalone solution; Srinivasan, has also proven that a MeSH standalone solution is unreliable

The results indicate that similar to using a MeSH only solution a Meta-thesaurus only solution doesn't provide added benefit over traditional free-text queries; however a Concept-Forest indexed system does provide an added benefit. Results indicate that MeSH makes significant contributions to retrieval performance but also that retrieval with a medical subtext should be re-examined and the use of outside controlled vocabularies and lexical databases should not be completely discounted.

In the next chapter we move forward in taking the success of the developed IR system and shift focus to the design and implementation of an online architecture. Chapter 7 offers a simple software architecture outlining components and software used as well as introducing a newer method in presenting retrieval results to users via Suffix Tree Clustering where results are ‘grouped’ into semantically related clusters before being presented to the user.

Chapter 7

On-line Architecture and Implementation

7.1 Overview

Chapter 7 serves to take the Information Retrieval system developed in Chapter 6 and to create a simple online implementation. The online component will accept a user's query and then transform that query into a Concept-Forest. The Concept-Forest will then be submitted to a Concept-Forest indexed database to retrieve like medical abstracts. The results retrieved will then be grouped into semantically related clusters using Carrot^2 (see Section 7.1.1). Once clusters have been formed the results will be formatted and displayed to the user.

7.1.1 Suffix-Tree Clustering using Carrot^2

Carrot^2 is an open-source search results clustering engine [66]. Carrot^2 exposes the clustering algorithms Lingo, Suffix-Tree Clustering (STC), and Lingo3G. STC also referred to as PAT-Tree or position tree is a compact representation of a trie corresponding to the suffixes of a given string where all nodes with one child are merged with their parent [63]. Carrot^2 automatically organize small collections of documents, e.g. search results into thematic categories. In this thesis STC has been used.

7.1.2 Why Suffix-Tree Clustering

Chapter 6 reviewed the performance results of the designed information retrieval system in terms of precision-recall curves, recall @ k measures, and the 11-pt average precision statistic. In doing so it became apparent that the implemented IR system suffers from inconsistently retrieving rlevant top ten

results. Although the IR-system achieves an 11-pt average of .31 and a top recall of greater than ninety-percent, the precision at the ten percent recall level is averaged at approximately .6 – this means that a little over half of the top 10 documents retrieved will be relevant. Most IR-systems surveyed contained a much higher top 10 recall statistic.

To curb the effects of this blip we employ the use of STC. STC takes the retrieved results and organizing these results around document similarities. Being that these documents were retrieved using a Concept-Forest index, the relevancy can still be assumed; we've just opted to present the end-user "clusters" of semantically related information and against a top N of search results. *Figure 24* captures a screen-shot of expected results of a submitted query.

This concludes the Online Architecture and Implementation phase of the thesis. A summary of the thesis is presented in the following chapter.

7.2 Online Architecture

The online-architecture can be partitioned into five major components with each being addressed in this chapter. The five components are the Client Machine, the Server Machine, Three Core Databases, an Entity Layer (a logic abstraction), the Presentation Layer (a logic abstraction), and the Parsing Linguistics Software. Each is delineated further below.

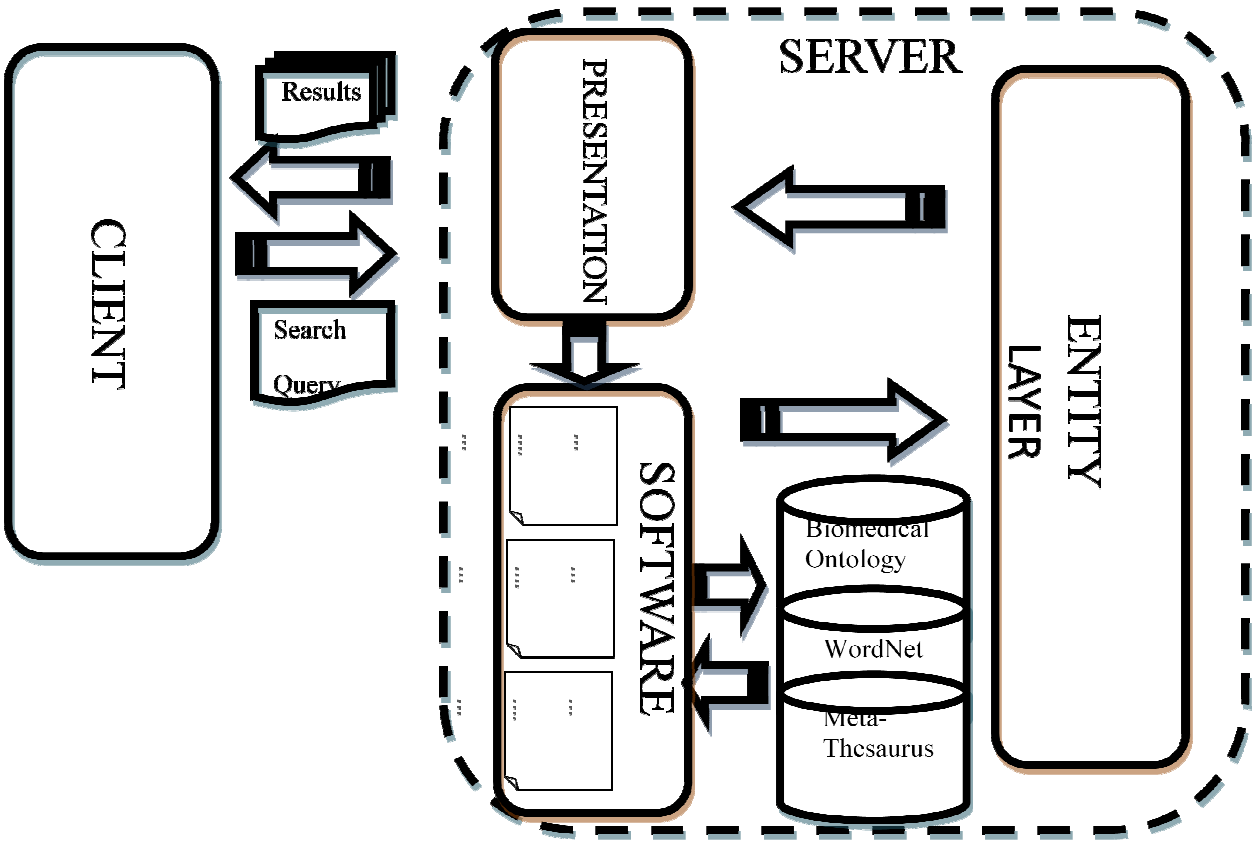


Figure 22: Software Architecture

7.2.1 Client Machine

The client is any machine that issues a service request to the Apache WebServer.

7.2.2 Server Machine

The server is based upon the LAMP stack or software bundle; Linux, Apache, MySQL, and PHP/Perl/Python. Fedora Core 8, Linux kernel 2.6.26.8-57.fc8 is adopted along with the Apache Tomcat WebServer, version 2.2.14. PHP, version 5.3.0, Perl version 5.8.8, MySQL version 4.1, and Carrot^2 version 3.1.1 are also employed. The server machine resides on a DELL desktop equipped with a 1.0 GHz Intel Pentium IV processor and 512 MB RAM.

7.2.3 Core Databases

The databases are essential to the functioning of the search engine / IR-system. The first

database contains the Concept-Forest bases indices used to in information retrieval. The remaining two databases provide the means for user-query transformation. User query-transformation is the process by which an expressed information need is converted to the Concept-Forest equivalent. For this transformation to take place two databases must be present, WordNet version 2.4 (or later) and the 2009 release of the NLM Metathesaurus (or a later release).

7.2.4 Parsing Linguistic Software

The parsing linguistic software serves as initial gateway to query-transformation. A user's query is submitted to this stage to be transformed to its equivalent Concept-Forest. The parsing linguistic software is also responsible for issuing the final query submitted to retrieve documents from the indexed database.

7.2.5 Entity Layer

The entity layer masks the presence of the suffix-tree clustering algorithm. The paring linguistic software retrieves documents from the Concept-Forest Indexed database and submits these results to the entity layer for clustering. The entity layer then submits the final clusters to the presentation layer for a visual layout.

7.2.6 Presentation Layer

The presentation layer serves as the entry point into the search-service. This component is comprised of HTML and PHP and is responsible for sending user-queries to the parsing linguistic software. This layer is also responsible for receiving results of requests from the entity layer and formatting those results in a visually pleasing manner.

7.3 Process Flow

7.3.1 Technical View

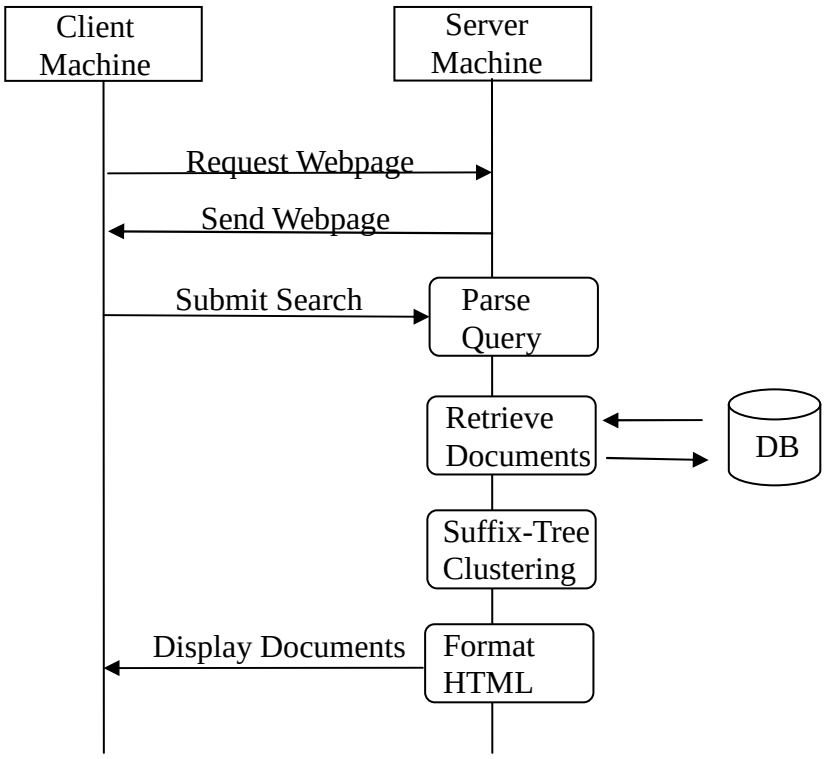


Figure 23: Sequence Diagram of Web Implementation

Interaction with the search system begins with a request from the client (as shown in Figure 25). The WebServer responds to the client by sending the client a webpage that contains forms to accepts and submit user input. The client may enter search-terms via the returned webpage and submit the search terms via an HTML POST. Once a search-query has been posted through the presentation layer, control is funneled into the parsing linguistic software. In this stage the software references WordNet and the NLM Meta-thesaurus to create a Concept-Forest representative of search-query terms. Once a Concept-Forest is generated, the software then queries the Concept-Forest indexed database to retrieve search results. After search results are obtained the parsing linguistic software passes

control to the entity layer.

The entity layer receives the retrieved query results from the parsing linguistic software layer. The entity layer then uses the suffix-tree clustering algorithm to group the results of the retrieved documents into semantically related categories. In this stage the entity layer also pre-formats HTML/PHP/JavaScript and hands control back to the presentation layer where search results are displayed to the user.

7.3.2 User View

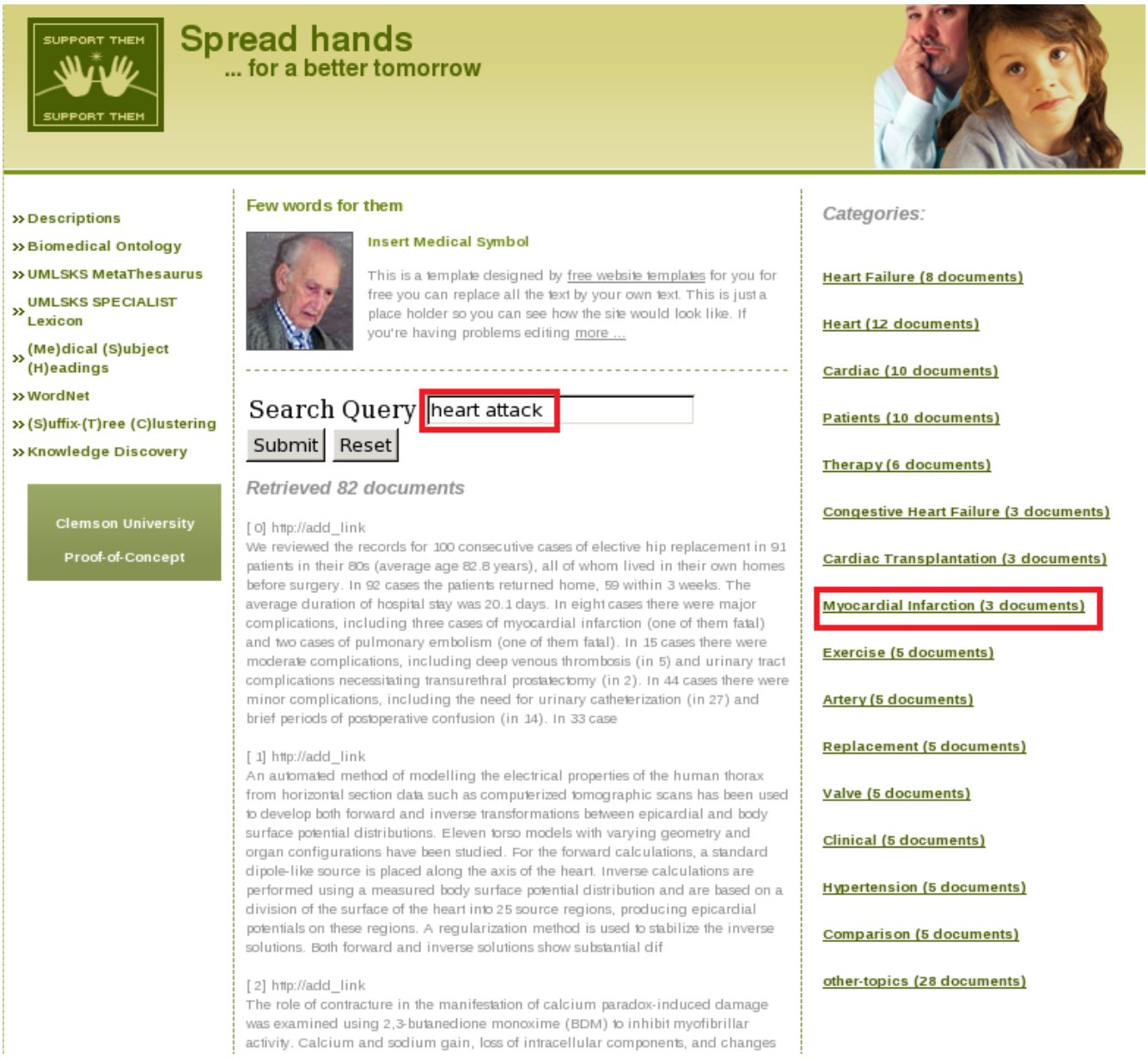


Figure 24: Sample Results from IR System

The layout of the system is shown above. The webpage is partitioned into three sections. The first section to the immediate left contains links that provide information about the retrieval system. The middle section is comprised of a form that allows users to submit queries to the system. The middle section is also where results are shown after a user submits a query. The final partition to the far right consists of the semantic clusters found through STC using the retrieved results from the user's

query as input.

As shown in the above figure, after a query is submitted results are retrieved and displayed as a Top N result posting initially. After submitting a query users than have the option of selecting subsets of the retrieved documents by browsing clusters that may be more semantically aligned with the user's initial query. If a user selects a cluster, then the documents that fall into the category of the cluster are displayed in the middle partition.

The last and final chapter provides a summation of the work presented while also addressing the shortcomings of this work and providing future directions and final thoughts.

Chapter 8

Conclusion

8.1 Summary

We have proposed a search engine suitable for the medical research community to facilitate effective searches without introducing the overhead of learning new search technologies or controlled vocabularies. The presented Ontology-Indexed information retrieval system has proven to retrieve fewer medical abstracts that are irrelevant to a user's search query and to retrieve more medical abstracts that are relevant to the same search query; this has been shown through experiments which reported higher recall and precision when compared to traditional Free-Text retrieval systems or systems which utilize a controlled vocabulary such as Medical Subject Headings (MeSH). This retrieval system allows medical researchers, doctors, and patients to search a medical corpus without having significant medical training or knowledge of the vocabulary while also requiring minimal system maintenance.

In this thesis we have proposed and implemented an information retrieval system that utilized an ontology index for information retrieval. The need for ontology arose from the complications of classifying and retrieving text from medical literature. The ontology was created with the aid of the lexical database WordNet and the National Library of Medicine's Meta-thesaurus. The techniques and algorithms developed and displayed in this thesis serve to resolve wide-spread problems stemming from synonym identification, and term-sense disambiguation. Our efforts have culminated into the construction of the Concept-Forest, a term-sense disambiguation and synonym and term-sense resolution tool for representing text. We have shown how similarities of two Concept-Forests could

be quantified and shown that our methods outperform existing methods for small to midsize collection-sets. Lastly, we've stepped through the design and implementation of an IR-System that leverages the Concept-Forest representation of medical text and have shown that a Concept-Forest or Ontology Indexed outperforms the highly used Medical Subject Headings controlled vocabulary. Towards this end we have also outlined two challenges, processing time and changing vocabulary, that need to be surmounted if we expect a wide area of interest in adopting this technology. Great gains have been made under the guise of this research such that anyone with an interest could adopt or further our work.

8.2 Future Directions

8.2.1 Ontology Representation \ Ontology Merge

There are 870,000 medical concepts in the National Library of Medicine medical thesaurus, to decrease the processing time in creating a concept-forest for text WordNet could be expanded locally to include these terms that exist in the medical thesaurus but not inside WordNet. Medical concepts that are found in WordNet could be expanded to include additional information that the medical thesaurus provides. Integrating the medical thesaurus into WordNet removes the performance bottleneck of multiple queries to a separate database. Developing a method for merging the ontology's also serves to place WordNet as 'living' ontology in such that other existing ontology's may be merged with an existing knowledge source.

8.2.2 Information Retrieval System Configuration Changes

The initial experiments for Free-Text+MeSH/Cosine-Coefficient solution fared better then the Concept-Forest/TF-IDF configuration but did not outperform the Concept-Forest/Okapi BM25 configuration. To further understand the differences between the two systems, Concept-Forest and Free-Text+MeSH research should be conducted to ascertain the differing retrieval strategies under nearly identical configurations. Specifically, because the presented information retrieval system were bare systems in which the main contributors were indexes and scoring functions, we believe that the

scoring functions should be identical in future tests. The 11-pt average precision measure is an indication of the better overall system, however, using identical scoring functions may provide additional insight as to reasons one system outperforms another.

8.2.3 Retrieval Feedback and the Ontology / Language Models

To further our information retrieval system by increasing its 11-pt average precision score, additional work should be devoted to the area of retrieval feedback. The objective of retrieval feedback is to modify the user's original query into one that is more effective for retrieval. We propose a study in pseudo-relevance feedback, also known as blind retrieval feedback. The method is to do normal retrieval, in response to a user's query, to find an initial set of the most relevant documents to a user's query. The method then assumed that the top k ranked documents are relevant, and does retrieval feedback under this assumption by modifying the original query to include terms that are common across the top k documents. We propose searching the retrieved k documents and finding a common relationship with the use of the ontology, and then augmenting the user's query to contain found relationships and then resubmitting the query; this may take significant processing time to create the ontology. We also suggest the use of language models in retrieval feedback, discovering the terms or sentences that are most likely to be contained with a majority of the top k retrieved documents and then resubmitting the user's query with the found terms/sentences.

8.3 Ontology-Index Limitations

This thesis showcases the advantages of adopting the notion of the ontology is information retrieval; however leverage the ontology is experiments is not a straight-forward process. In this thesis we delineate what it means to constitute an ontology, the means by which ontology's are considered similar, and a means by which we can increase the universe of discourse by the addition of supplemental data contextual to the domain of interest. Each element builds upon the next until the ontology meets the advances of modern-day information retrieval systems. It is where the two meet,

the idea of the ontology and Information Retrieval Systems, where we are afforded the opportunity to coalesce the two to create a very unique IR-system that we have proven outperforms IR-systems that rely on traditional keyword searches, or systems that are reliant on the inclusion and maintenance of a controlled-vocabulary. However, our approach is not without faults, namely,

- Processing Time - the length of time undergone to convert a single medical abstract to its Concept-Forest equivalent
- Changing Vocabulary – the constant modification of terms used in the National Library or Medicine's Meta-thesaurus and Princeton's Lexical Database WordNet

8.3.1 Processing Time

Perhaps the single most important issue impeding wide acceptance and subsequent adoption by the National Library of Medicine or the medical research community is the length of time required to process and index a single medical document. The NLM / Pubmed has compiled a data warehouse of more than 3 million digital medical abstracts with an annual addition of approximately 400,000 new records per year. On code-analysis we have identified the primary bottleneck as belonging to the sections of code dedicated to medical concept resolution; meaning that although we have a local version of the NLM Meta-thesaurus we are still incurring significant costs to access the data. We believe this issue to be a surmountable one in that experiments were conducted on a less than top-end DELL workstation. Furthermore, we have identified opportunities for parallelism, increasing the amount of work done in an identical duration of time. From the beginning phases of our research, we've made great gains applying techniques to smaller sized documents or corpuses. The material as presented can be readily adopted at agencies or companies that house smaller medical corpuses. Our methods may provide an added benefit to smaller sized datasets as it removes the burden from the adopting agency to create a controlled vocabulary as in the instance of NLM / Pubmed and the Medical Subject Headings.

8.3.2 Changing Vocabulary

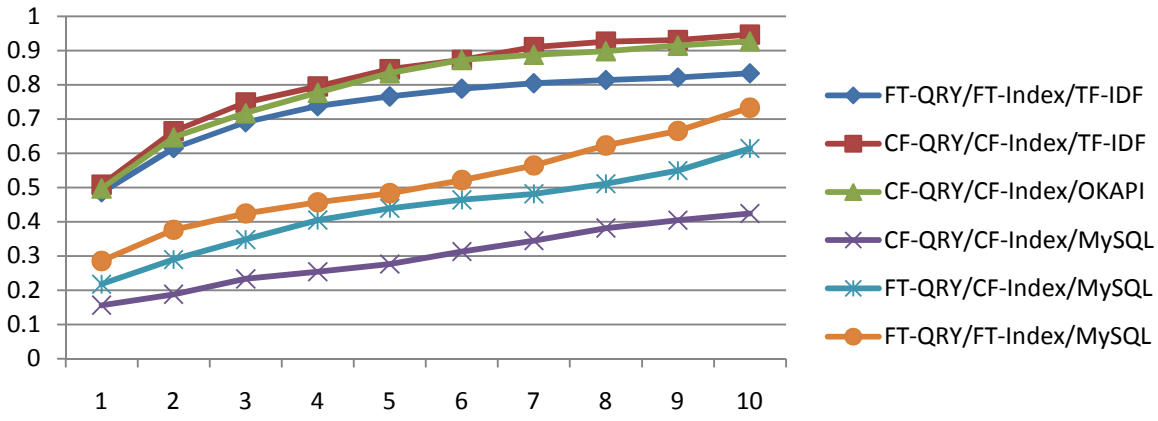
The second issue that arises with a wide deployment on a large dataset is the issue of updating created ontologys that reflect current nomenclature found in NLM Meta-thesaurus and WordNet. As language and relationships between concepts evolve it would be ideal if there existed an automated process that would allow the updating of ontologys with that of changes to the core vocabulary. The information retrieval system as presented is unequipped to ensure this scenario is the least impactful; this means that if significant changes were committed to the NLM Meta-thesaurus or WordNet the entire document collection could need to be re-executed and indexed a second time. Fortunately, the controlled vocabularies used to build the ontology are updated very infrequently and the terms added are of a small order; this fact increases the lifetime of an ontology-indexed database. This issue narrows the potential interest-base by removing those agencies with a hard business need to be the most-up-to-date.

The proposed information retrieval system is not a free-standing, self-monitoring system; the IR-System does require annual maintenance – ensuring the latest version of WordNet and the NLM Meta-thesaurus – as any evolving entity requires, however, it is the amount of maintenance we wish to scale-back.

APPENDICES

Appendix A

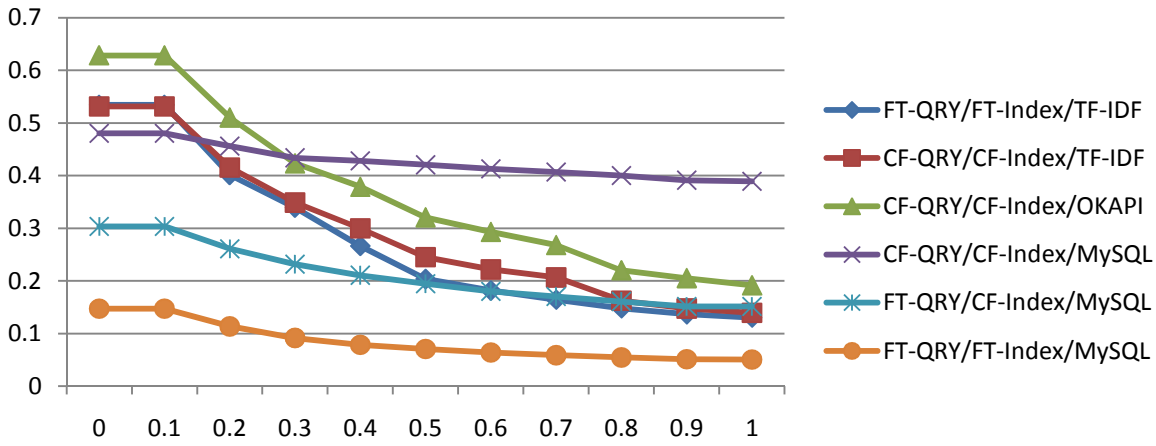
A: Recall @K and 11-Point Average Precision for OHSUMED Experiments



A-1 Recorded Recall for All Retrieval Strategies for OHSUMED Experiments

FT-QRY/FT-Index/TF-IDF	CF-QRY/CF-Index/TF-IDF	CF-QRY/CF-Index/OKAPI	CF-QRY/CF-Index/MySQL	FT-QRY/CF-Index/MySQL	FT-QRY/FT-Index/MySQL
0.485565825	0.507853367	0.49742	0.15649	0.217613	0.285511
0.614771021	0.664251718	0.646775	0.188194	0.28998	0.376778
0.690666602	0.747882625	0.717243	0.233315	0.348211	0.424395
0.737605518	0.794867155	0.776871	0.2544	0.404922	0.456829
0.765837063	0.845002706	0.83402	0.276376	0.438991	0.483474
0.788767738	0.872397667	0.872602	0.313139	0.463932	0.521785
0.804357944	0.909845344	0.887755	0.345071	0.481539	0.56445
0.813784083	0.925532811	0.89821	0.381605	0.510925	0.622735
0.821230935	0.930432566	0.914449	0.404668	0.549062	0.665321
0.833345289	0.946290672	0.926982	0.424267	0.613858	0.732992

A-2 Reported Recall Values for OHSUMED Experiments



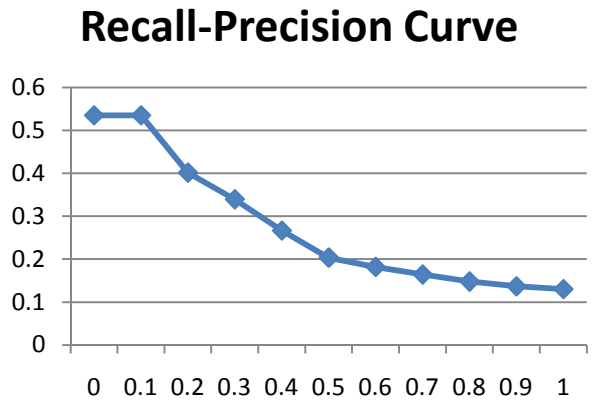
A-3 11-pt Average Precision for All Retrieval Strategies for OHSUMED Experiments

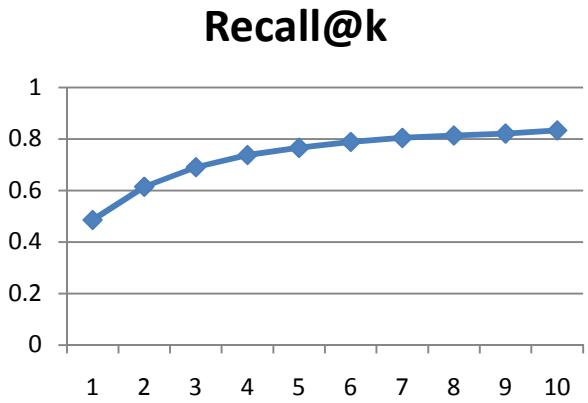
FT-QRY/FT-Index/TF-IDF	CF-QRY/CF-Index/TF-IDF	CF-QRY/CF-Index/OKAPI	CF-QRY/CF-Index/MySQL	FT-QRY/CF-Index/MySQL	FT-QRY/FT-Index/MySQL
0.534560535	0.531303217	0.628277	0.480076	0.303211	0.147067
0.534560535	0.531303217	0.628277	0.480076	0.303211	0.147067
0.401288829	0.415100964	0.510193	0.456143	0.260937	0.113447
0.339233551	0.348666253	0.42297	0.433751	0.231501	0.091465
0.266386619	0.29987571	0.378477	0.428014	0.210642	0.078497
0.203519804	0.245019074	0.320432	0.420423	0.194617	0.07035
0.181749002	0.221653897	0.29276	0.412933	0.179769	0.063931
0.164139157	0.206422404	0.267811	0.406679	0.170332	0.058695
0.147768425	0.162142188	0.219968	0.400014	0.160713	0.054504
0.136677747	0.147455193	0.205098	0.391116	0.151218	0.051057
0.130288622	0.139155434	0.191439	0.388885	0.151242	0.050651

A-4 Reported 11-pt Average Precision for OHSUMED Experiments

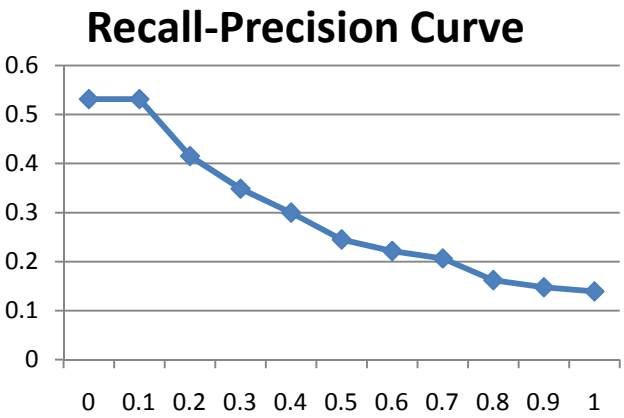
Appendix B

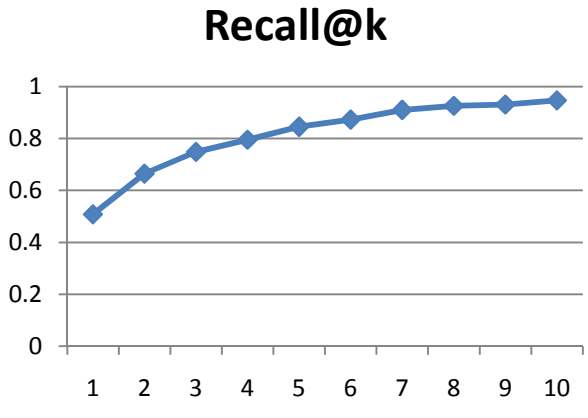
B: Recall-Precision Curves for OHSUMED Experiments



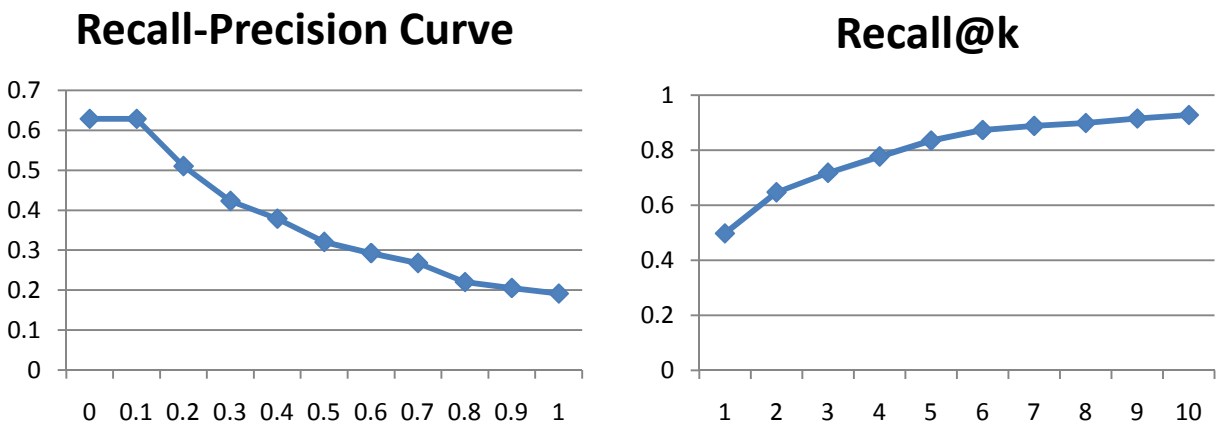


B-1 Free-Text Query and Free-Text Index and TF-IDF Scoring Function

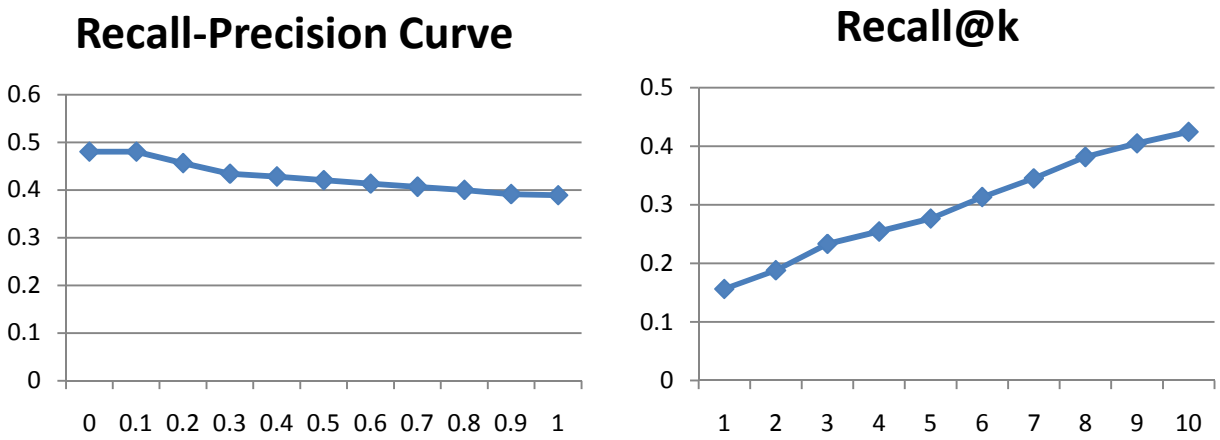




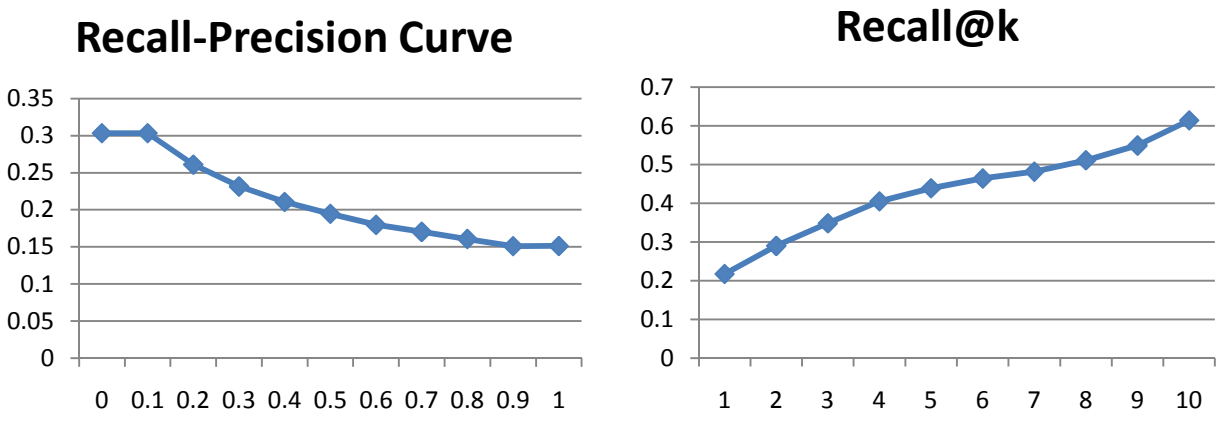
B-2 Concept-Forest Query and Concept-Forest Index and TF-IDF Scoring Function



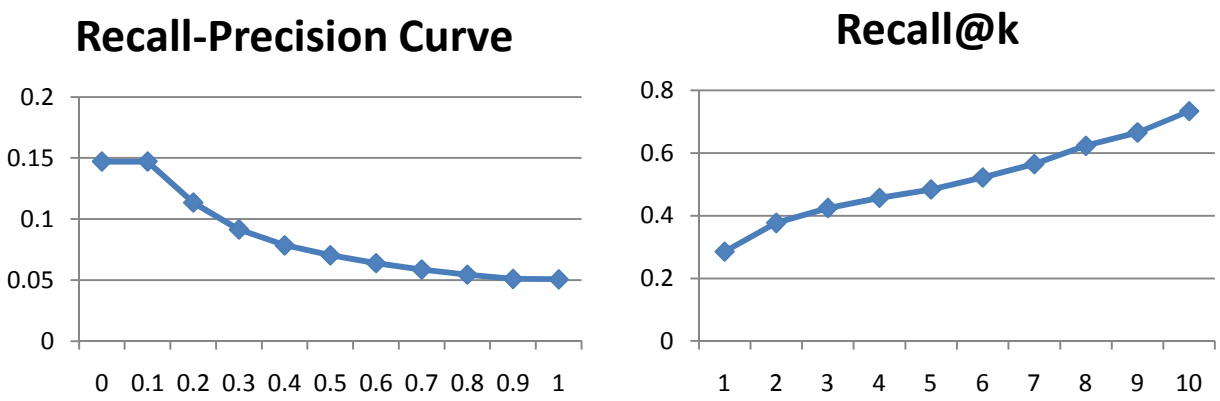
B-3 Concept-Forest Query and Concept-Forest Index and Okapi BM25 Scoring Function



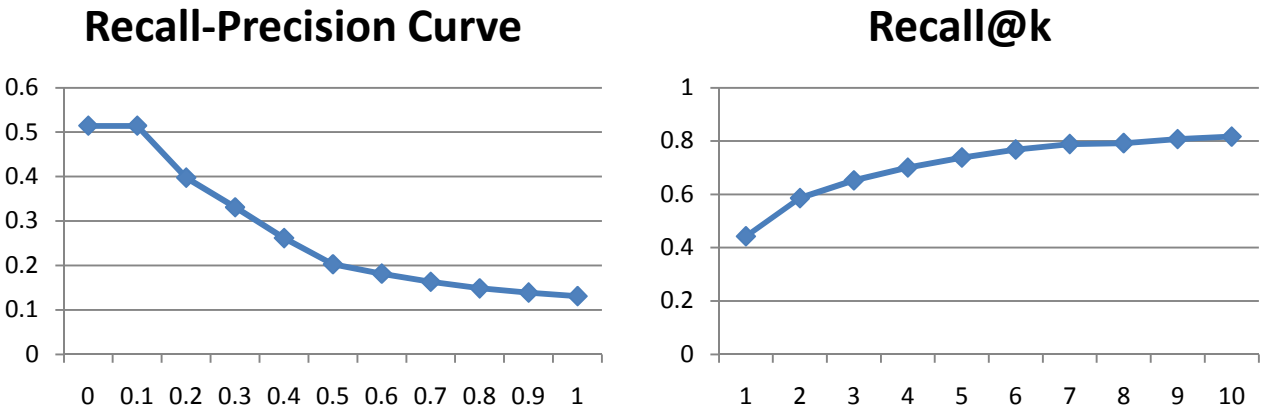
B-4 Concept-Forest Query and Concept Forest Index with default MySQL Scoring Function



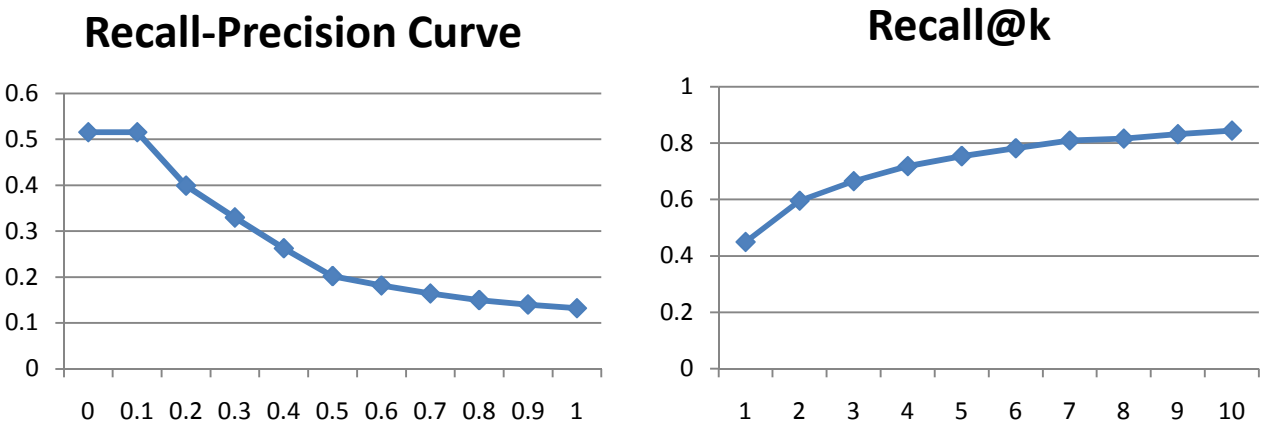
B-5 Free-Text Query and Concept Forest Index with default MySQL Scoring Function



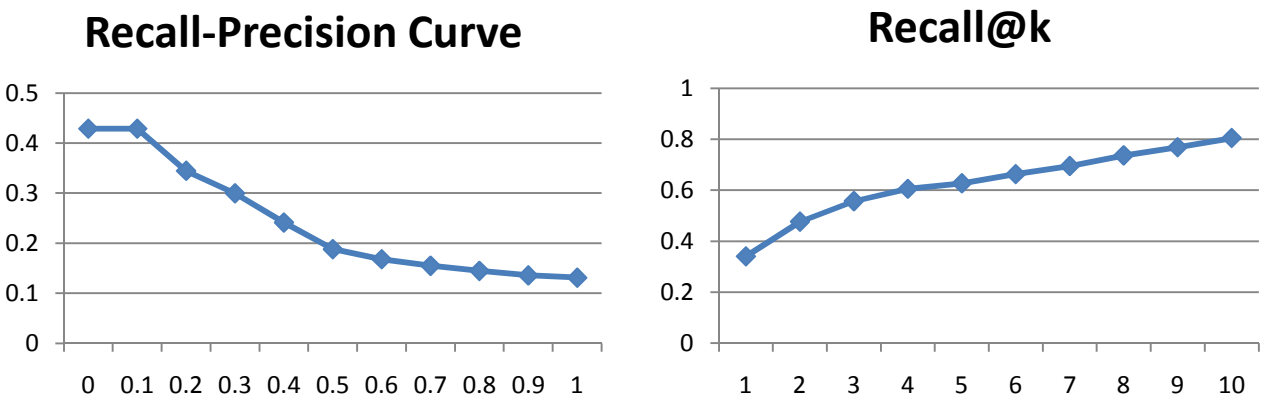
B-6 Free-Text Query and Concept Forest Index with default MySQL Scoring Function



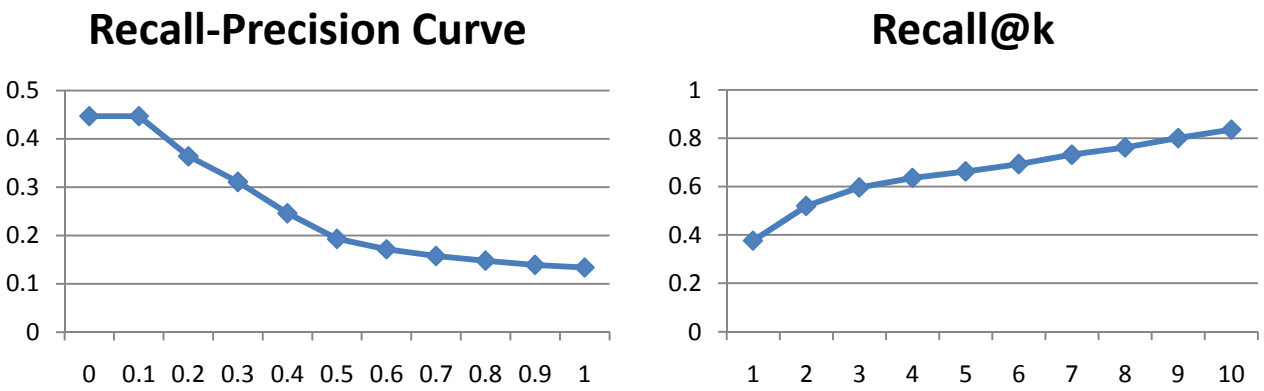
B-7 Free-Text Query and Free-Text Index with VSM Scoring Function



B-8 Concept-Forest Query and Concept-Forest Index with VSM Scoring Function



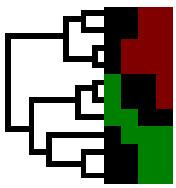
B-9 Free-Text Query and Free-Text Index with VSM Scoring Function



B-10 Free-Text Query and Free-Text Index with VSM Scoring Function

Appendix C

C: Software used in Creating the Ontology-Indexed Retrieval System

LingPipe Natural Language Processing Suite version 2.5		MySQL Relational Database Version 4.1	
Java Runtime Environment Version 1.5		Apache HTTP Server Version 2.2.14	
Hypertext Preprocessor Version 5.3.0		Princeton University WordNet Version 2.5	
Perl Version 5.8.8		Carrot Squared Version 3.1.1	
GNU Scientific Library gsl.1.9			
Cluster 3.0			

C-1 Open-Source Software Used.

Bibliography

[1] Aronson, A. (2001). Effective Mapping of Biomedical Text to the UMLS Metathesaurus: The MetaMap Program. *AMIA Symposium.*

[2] Aronson, A., & Marcetich, J. (1997). A MEDLINE Indexing Experiment Using Terms Suggested by MetaMap. *AMIA.*

[3] Aronson, A., & Rindflesch, T. (1994). Exploiting a Large Thesaurus for Information Retreval. *NLM Intelligent Multimedia Information Retreival Systems Management.*

[4] Aronson, A., Gay, C., & Kayaalp, M. (2005). Semi-Automated Indexing of Full Text Biomedical Articles. *MIA Symposium*, (pp. 271-275).

[5] Aronson, A., Kim, W., & Wilbur, W. (2001). Automatic MeSH Term Assignment and Quality Assessment. *AMIA Symposium*, (pp. 319-323).

[6] Bhogal, J., Macfarlane, A., & Smith, P. (2007). A Review of Ontology Based Query Expanasion Information. *Information Processing and Management: an International Journal* , 47-56.

[7] Blaz, F., Dunja, M., & Marko, G. (2005). Semi-Automatic Construction of Topic Ontology. *SiKDD Conference on Data Mining and Data Warehouses.*

[8] Borgelt, C., & Numberger, A. (2004). A Fast Fuzzy Clustering of Web Page Collections. *Statistical Approaches for Web Mining (SAWM).*

[9] Dang, C., Luo, X., & Zhang, H. (2008). WordNet-Based Summarization of Unstructured Document. *WSEAS Vol 7*, (pp. 1368-1374).

[10] Deerwester, S., Dumais, S., Fumas, T., & Landauer, G. (1990). Indexing by Latent Semantic Analysis. *Journal of the American Society for Information Science Vol 41 No. 6* , 391-407.

[11] Dimitrios, M. (2005). Word Sense Disambiguation for Exploiting Hierarchical Thesauri in Text Classification. *PKDD*, (pp. 181-192).

[12] Eccles, S. S.(2006). *Automatic Term Mapping*. Health Sciences Library.

[13] Elliman, D., Rafael, J., & Pulido, G. (2001). Automatic Derivation of On-Line Document Ontology. *15th Eruopean Conference on Object Oriented Programming.* Budapest, Hungary.

[14] Fellbaum, C. (1998). *WordNet: An Electronic Lexical Database.* The MIT Press.

[15] GOOGLE. (n.d.). *GOOGLE*. Retrieved from http://www.google.com

[16] Hersh, W., & Hickam, D. (1992). A Comparison of Retrieval Effectiveness for Three Methods of Indexing Medical Literature. *American Journal of the Medical Sciences* , 292-300.

[17] Hersh, W., & Hickam, D. (1995). Information Retrieval in Medicine: The SAPHIRE experience. *Journal of the American Society for Information Science* , 743-747.

[18] Hersh, W., Buckley, C., Leone, T., & Hickam, D. (1994). OHSUMED: An Interactive Retrieval Evaluation and New Large Test Collection for Research. *ACM SIGIR Conference on Research and Development in Information Retrieval*, (pp. 192-200).

[19] Hersh, W., Hickam, D., & Leone, T. (1992). Words, Concepts, or both Optimal Indexing Units for Automated Information Retrieval. *Symposium on Computer Application for Medical Care* (pp. 633-648). Washington, DC: American Medical Informmation Association.

[20] Hersh, W., Hickam, D., Haynes, R., & McKibbon, K. (1994b). A Performance and Failure Analyis of SAPHIRE with MEDLINE Test Collection. *American Medical Informatics Association* , 51-60.

[21] Hontho, A., Madche, S., & Staab, S. (2001). Ontology-based Text Clustering. *Proceedings of the IJAI-2001 Workshop Text Learning: Beyond Supervision*, (pp. 48-54). Seatle, USA.

[22] Hotho, A., Staab, S., & Stumme, G. (2003). Ontologies Improve Text Document Clustering. *Proceedings of the IEEE Internation Conference on Data Mining*, (pp. 541-544).

[23] Hung, C., Wemter, S., & Smith, P. (2004). Hybrid Neural Document Clustering Using Guided Self Organization and WordNet. *IEEE Intelligent Systems, Vol 19. No. 2*, (pp. 68-77).

[24] Jimeno-Yepes, A., & Berlanga-Llavori, R. (2009). Ontology Refinement for Improved Information Retrieval. *Information Processing & Management: Special Issue on Semantic Annotations in Information Retrieval.*

[25] Jing, L., Zhou, L., Ng, M., & Huang, J. (2006). Ontology Based Distance Measure for Text Clustering. *SIAM Text-Mining Workshop.*

[26] Kehagias, A., Petridis, V., Kaburlasaso, V., & Fragkou, P. (2001). A Comparison of word-and-sense based Text Categorization using Several Classifcation Algorithms. *Journal of Intelligent Information Systems* .

[27] Khan, L., & Luo, F. (2002). Ontology Construction for Information Selection. *Proceedings of the 14th IEEE International Conference on Tools with Artificial Intelligence*, (pp. 122-127). Crystal City, VA.

[28] Kim, J., Ohta, T., Tateeisi, Y., & Tsujii, J. (2003). GENIA Corpus - A Semantically Annotated Corpus for Bio-text Mining. *Conference of BioInformatics*, (pp. 180-182).

[29] Koller, D., & Sahami, M. (1998). Hierarchically Classifying Documents Using Very Few Words. *Proceedings of teh 14th International Conference on Machine Learning.*

[30] Kushal, D., & Lawrence, S. (2003). Mining the Peanut Gallery: Opinion Extraction and Semantic Classification of Product Reviews. *Proceedings of the 12th International Word Wide Web Conference (WWW)*, (pp. 529-528). Budapest, Hungary.

[31] Landauer, T. Foltz, P., Laham, D. (1998) An Introduction to Latent Semantic Analysis. *Discourse Processes Vol 25 No. 2,* (pp. 259-284)

[32] Lee, C., Jian, Z., & Huang, L. (2003). A Fuzzy Ontology and Its Applications to News Summarization. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics Vol 35 No 5,* (pp. 859-990).

[33] Leroy, G., & Chen, ,. H. (2001). Meeting Medical Terminology Needs - The Ontology Enhanced Medical Concept Mapper. *Journal IEEE Transactions on Information Technology in Biomedicine* .

[34] Lipika, D., Ashish, C., & Sachin, K. (2006). Generating Concept Ontologies through Text Mining. *IEEE/WIC/ACM Interncation Conference on Web Intelligence,* (pp. 23-32).

[35] Liu, Y., Scheuemann, P., Li, X., & Zhu, X. (2007). Using WordNet to Disambiguate Word Senses for Text Classification . *ICCS 2007 Part III LNCS 4489,* (pp. 780-788).

[36] Luca, E., & Nmberger, A. (2004). Ontology Based Semantic Online Classification of Documents: Supporting Users in Searching the Web. *Proceedings of European Symposium on Intelligent Technologies (ENUITE).*

[37] Mandar, M., Singhal, A., & Buckley, C. (1998). Improving Automatic Query Expansion. *ACM SIGIR Conference on R&D in Information Retrieval.*

[38] Manning, C., Raghavan, P., & Schutze, S. (2008). *Introduction to Information Retrieval* . Cambridge, MA: Cambridge University Press.

[39] Melucci, M. (2008). A Basis for Information Retrieval. *TOIS Vol 26,* (pp. 1-41).

[40] Medicine, N. L. (2004 ,November 10). *PubMed's Automatic Term Mapping Enhanced*. NLM Tech Bull., p.1

[41] Nahm, U., & Mooney, R. (2002). *Text Mining with Information Extraction.* AAAI Technical Report - SS-02-06.

[42] National Library of Medicine PubMed. (2009, November). *MeSH Browser*. Retrieved October January, from MeSH Browser: http://www.nlm.nlih.gov/mesh/mbinfo.html

[43] National Library of Medlicine. (2009, November). *SPECIALIST Lexicon*. Retrieved November 2009, from National Library of Medcine Unified Modeling System: http://lexsrv3.nlm.nih.gov/SPECIALIST/index.html

[44] Navigli, R., Velardi, P., & Gangemi, A. (2005). Ontology Learning and its Application to Automated Terminology Translation. *IEEE Inteligent Systems, Vol 42. No. 3* , 22-31.

[45] Neoh, H., & Na, J. (2008). Effectiveness of UMLS Semantic Network as a Seed Ontology for Medical Domain Ontology Building. *ASLIB Proceedings Vol 60*, (pp. 32-46).

[46] Parsons, K., & McCormac, A. (2009). The Use of a Context-Based Information Retrieval Technique. *DSTO*.

[47] Rajman, M., & Besancon, R. (1997). Test Mining: Natual Language Techniques and Text Mining Applications. *IFIP Working Conference Semantic Database.*

[48] Rinaldi, A. (2009). An Ontology-Driven Approach for Semantic Information Retrieval on the Web. *ACM Trans Internet Technology*, (pp. 1-24).

[49] Sabou, M. (2005). *Learning Web Service Ontolgies: An Automatic Extraction Method and tis Evaluation, Ontology Learning, and Population.* IOS Press.

[50] Salton, G., Wong, A., & Yang, C. (1975). A Vector Space Model for Automatic Indexing. *Communications of the ACM Vol. 18, No. 11*, (pp. 613-620).

[51] Scott, S., & Matwin, S. (1998). Text Classification using WordNet Hypernyms. *Proceedings of the Conference Association for Computational Linguistics*, (pp. 38-44). Somerset, New Jersey.

[52] Sedding, J., & Kazakov, D. (2004). WordNet-based Text Document Clustering. *Proceedings of the 3rd Workshop on Robust Methods in Analysis of Natural Language Data (ROMAND)*, (pp. 104-113). Geneva.

[53] Sheet, M. F. (n.d.). *MEDLINE Fact Sheet*. Retrieved from National Library of Medicine: http://www.nlm.nih/gov/pubs/factsheets/meldine.html

[54] Simone, T., & Kazakov, D. (2005). Using WordNet Similiarity and Antonymy Relations to Aid Document Retreival. *RANLP Recent Advances in Natural Language Processing*, (pp. 21-23). Boroveta, Bulgaria.

[55] Siou, O., Chin, N., Kulathuramaiyer, A., & Yeo, W. (2006). Automatic Discovery of Concepts from Text. *IEEE/WIC/ACM Interational Conference on Web Intelligence (WI 2006)*, (pp. 1056-1049).

[56] Soo, V., & Lin, C. (2001). Ontology-Based Information Retrieval in Multi-Agent System for Digital Library. *Proceedings of teh Sixth Conference on Artificial Intelligence and Applications*, (pp. 241-246). Taiwan.

[57] Srinivasan, P. (1995). Exploring Query Expansion Strategies for MEDLINE. *Journal of the American Medical Information Association, Vol 3* , 157-167.

[58] Srinivasan, P. (1996). Optimal Document-Indexing Vocabulary for MEDLINE. *Information Processing Management Vol 32 No. 5*, (pp. 503-514).

[59] Steinback, M., Karypis, G., & Kumara, V. (2000). A Comparison of Document Clustering Techniques. *KDD Workshop on Text Mining.* Boston, MA.

[60] Subramaniam, L., Sougata, M., & Kankar, P. (2003). Information Extraction from Biomedical Literature: Methodolgy, Evaluation, and an Application. *12th International Conference on Information and Knowledge Management.*

[61] Sugumaran, V., & Storey, V. (2002). Ontologies for Conceptual Modeling: Their Creation use, and Management. *International Journal of Data and Knowledge Engineering Vol. 42 No. 3* , 251-271.

[62] Tanabe, L., Scherf, U., Smith, L., & Lee, J. (1999). MedMiner: An Internet Text-Mining Tool for Biomedical Information with Application to Gene Expression and Profiling. *Conference of BioTechniques*, (pp. 1210-1217).

[63] TREC. (2006). *Reuters-21578 Text Categorization Collection*. Retrieved 12 1, 2006, from TREC Evaluations: http://kdd.ics.uci.edu/databases/reuters/21578/reuters21578.html

[64] Vizenor, L., Bodenredier, O., Peters, L., & McCray, A. (2006). Enhancing Biomedical Ontology's through Alignment of Semantic Relationships: Exploratory Approaches. *AMIA Annual Symposium*, (pp. 804-808).

[65] Wang, J., & Taylor, W. (2007). Concept Forest: A New Ontology-Assisted Text Document Similarity Measurement. *International Conference on Web Intelligence.*

[66] Weiss, D., & Osinski, S. (2009, September). *Carrot Squared - Open Source Framework for Building Search Clustering Engines*. Retrieved September 2009, from Carrot Squared: http://project.carrot2.og/

[67] Widyantoro, J., & Yen, A. (2001). A Fuzzy Ontology-Based Abstract Search Engine and its User Studies. *Proceedings of the IJACI-2001 Workshop Text Learning: Beyond Supervision*, (pp. 48-54). Seattle, USA.

[68] William De Luca, E., & Numberger, A. (2006). Using Clustering Methods to Improve Ontology-Based Query Term Disambiguation. *Int. J. Intell Syst Vol 27 No. 7*, (pp. 694-709).

[69] Yang, C., Salton, G., Wong, A. (1975). A Vector Space Model for Automatic Indexing. *Communications of the ACM* (pp. 613-620)

[70] Yang, Y. (1994). Expert Network: Effective and Efficient Learning from Human Decisions in Text Categorization and Retrieval. *ACM SIGIR Conference in Information Retrieval*, (pp. 13-22).

[71] Yang, Y., & Chute, C. (1993b). An Application of Least Squares fit Mapping Test. *ACM SIGIR Conference on Research and Development in Information Retrieval*, (pp. 281-290).

[72] Yang, Y., & Chute, C. (1993a). Words or Concepts: the Feature of Indexing Units and their Optimal use in Information Retrieval. *Symposium on Computer Applications for Medical Care* (pp. 685-689). Washington, DC: American Medical Informatics Association.

[73] Youjin, C., & Jong-Hyeok, L. (2005). Practical Word-Sense Disambiguation Using Co-occuring Concept Codes. *Machine Translation*, (pp. 59-82).