1-1-2013

A Relevance Feedback-Based System For Quickly Narrowing Biomedical Literature Search Result

Massuod Hassan Alatrash

Wayne State University,

A RELEVANCE FEEDBACK-BASED SYSTEM FOR QUICKLY NARROWING BIOMEDICAL LITERATURE SEARCH RESULT

by

MASSUOD HASSAN ALATRASH

DISSERTATION

Submitted to the Graduate School

of Wayne State University,

Detroit, Michigan

in partial fulfillment of the requirements

for the degree of

DOCTOR OF PHILOSOPHY

2013

MAJOR: COMPUTER ENGINEERING

Approved by:

_____

Advisor Date

_____

_____

_____

DEDICATION

I would like to express my deep and sincere grateful to the people who provide me with full support and love during the first stage of my life: my father, my mother, and my old brother Hussain. Three of them fully support and sacrifice the most for me to secure me a happy life. Appreciation is also due to all of my other brothers and sisters. I would like to thank my cousin Mosbah for all of his support in my life.

To my family, wife and two sons, Hassan and Osama, for their full support, and patience during my Ph.D. study.

ACKNOWLEDGEMENTS

I would like to express my sincere appreciation to Professor Hao Ying, who contributed tremendous time to guide my research. Appreciation is also due to Professor Ming Dong, Professor Nabil Sarhan, and Professor Feng Lin for their constructive comments and valuable suggestions. I also would like to thank Dr. R. Michael Massanari and Dr. Peter Dews for their contributions in the evaluation of the experiment results and for their valuable suggestions.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Problem Statement

The number of published literature increases considerably every year, which leads to growing of available online information content. The online literature is an important source of information that helps people locate their information need. Since almost all type of information resource stored in electronic format, for instance, online databases and digital libraries, search process for the most relevant information from online literature becomes increasingly important task. The increase of online literature makes the search process for the most relevant information extremely expensive, and time-consuming task and leads to sifting through many results to find the relevant ones. There exist several hundreds of search engines and online databases for literature information retrieval. The search engines and online databases usually return a long list of result that satisfies the user's search criteria. The returned list of hits is often too long for the user to go through every hit if he/she does not exactly know what he/she wants or/and does not have time to review them. In today's life cost setting, the user often does not have time to sift manually the long returned list of hits to find the exact information; he/she may be able to review a couple of the first hits of the returned result, but he/she cannot go through the whole list.

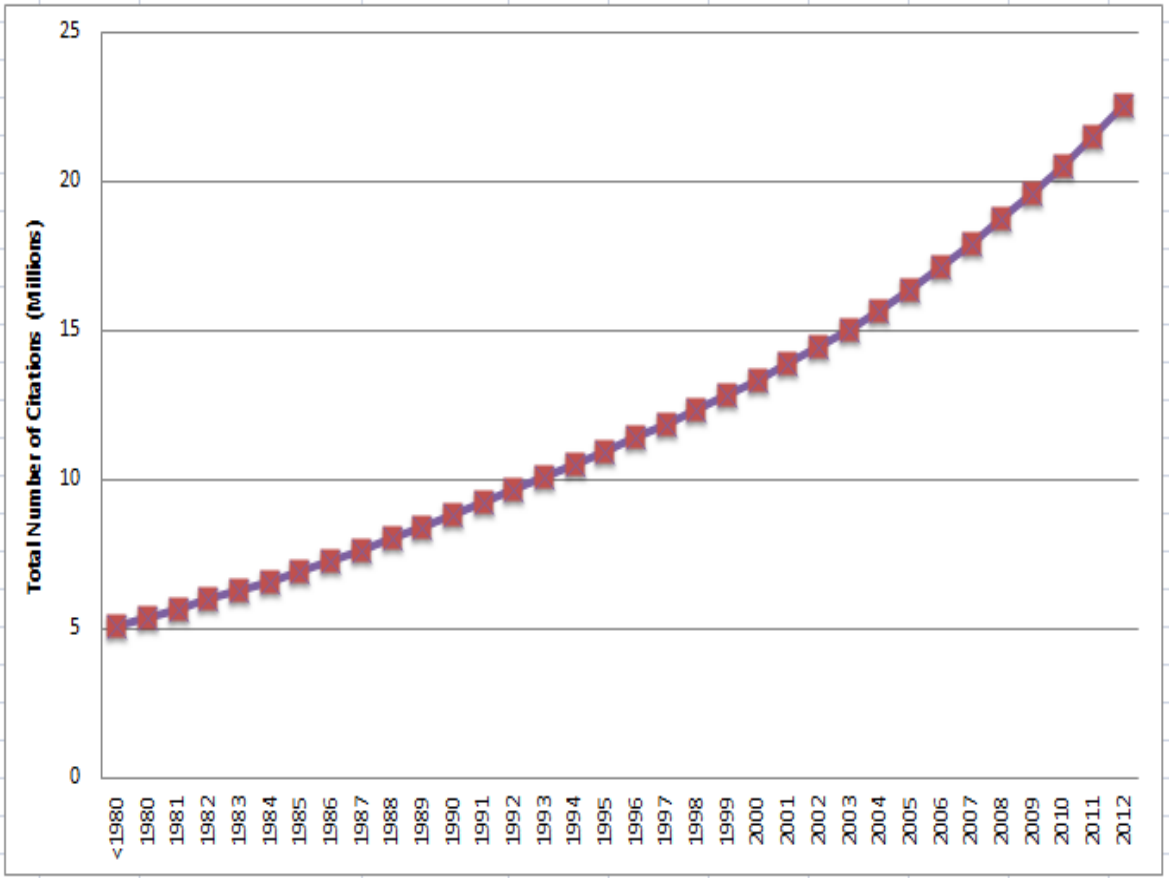

Figure 1.1: Annual biomedical citation grow rate in PubMed to 2012

My focus is on biomedical literature search. The biomedical field publishes a high volume of articles every year, and many of these are now available online in electronic format, which can be an important source that helps healthcare providers and clinicians make decisions in the patient care where they often need to consult the literature on the latest information in patient treatment [1-3]. For example, PubMed [4], which is a free key resource for medical professionals and biomedical researchers around the world, comprises over 23 million publications for biomedical literature from several resources (August 2013). It is developed by the National Center for Biotechnology Information (NCBI) at the National Library of Medicine (NLM). More than 500,000 new publications are added to

PubMed every year [4]. Moreover, the size of the biomedical literature has grown at double-exponential rate over the last three decades in PubMed as shown in figure 1.1 and pointed out in other previous work [5, 6]. The data that is used to construct figure 1.1 is obtained from [7]. Although, increasing the availability of online biomedical literature and fast growing of the computer networks facilitate the access to the information, the online literature search often provide users with more information than needed, much of which is irrelevant [1, 3, 8, 9]. Therefore, the tremendous increase of biomedical knowledge resources in electronic form has generated a great deal of interest [9]. The traditional used sources for biomedical information retrieval (e.g., PubMed) often return a list of documents in response to a user's query whereas the number of returned documents from large knowledge repositories is large [1, 10]. For instance, sending “Dobutamine” (a drug name) as query to PubMed database returns long result list of hits containing more than eight thousand citations. Figure 1.2 shows the grow rate of “Dobutamine” citations in PubMed to 2012. Other queries may result in retrieving a huge number of hits such as “breast cancer” which retrieves more than two hundred thousand hits. Adding more search criteria to the query may reduce the number of the retrieved result but is still return long list of hits that can be hard for the user to handle. However PubMed is a great source for up-to-date biomedical information, its users are usually overwhelmed by the huge retrieved list of hits [6] where more than one-third of PubMed queries returns more than one hundred hits [11]. Therefore, literature review is a time-consuming burden because it is hard to find relevant information in short time.

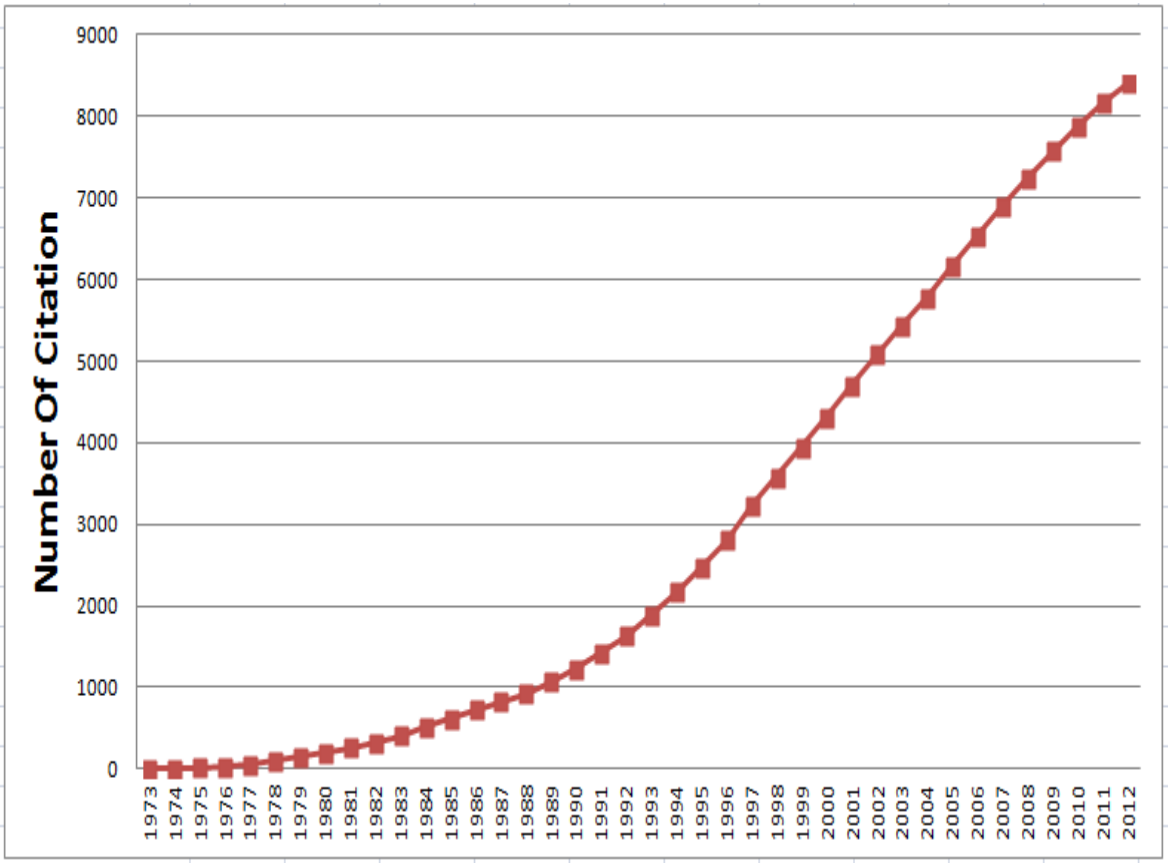

Figure 1.2: Annual Dobutamine citations grow rate in PubMed database to 2012

The medical professionals often consult the up-to-date medical information in patient treatment from biomedical information retrieval engines (e.g. PubMed) which return long list of hits in response to the user's query [1, 3, 6, 11]. The returned list of hits is often too long for the user to review if he/she does not know exactly what he/she wants. Thus, the problem of information overload forces us to sift manually through long returned list of documents to find the relevant and exact information. In today's healthcare cost settings, the medical provider is constantly under time pressure. Consequently, he/she does not have time to spend seeking the wanted information by manually reviewing the long returned list from the existing search engines and online databases. The healthcare provider

is sometimes unable to find the exact relevant information in short time because he/she has to spend more time than available (e.g. the patient is waiting in the clinic for the physician decision) to sift through too much information found by the existing search engines [1, 12]. Furthermore, the physician often knows only vaguely what he/she wants but does not often know exactly what he/she is looking for until he/she has found it. Finding the relevant medical information is an important issue that has received serious attention [1, 13-15]. Therefore, getting the relevant medical information support for decision making within a very short period of time (e.g. at point of care) from the vast online medical literature is an important endeavor.

To address the literature information overloaded problem, researchers have developed search engines concerning academic literature. There exist two types of general-purpose academic search systems: 1) open-domain search engines covering all topics such as Google Scholar, and Microsoft Academic Search, and 2) domain-specific search engine such as PubMed [4] covering biomedical domain. The general-purpose search systems focus on searching large collections to find documents that are relevant to a query. Moreover, there exist special-purpose deep search systems that provide pre-extracted information from published literature in biomedical domain such as, iHOP [16], PubMeth [17], and PPI-finder [18]. They use information extraction techniques to extract the information and the relations from literature abstracts. Although, there exist other types of biomedical literature search systems, studies showed that PubMed is one of the most resources frequently used by healthcare providers, medical professionals, and biomedical

researchers around the world especially in large hospitals [1, 12, 15] where it receives over 70 million queries every month [19].

By using the general-purpose academic search systems (e.g. PubMed or Google scholar), users can use successive queries with multiple criteria, such as journal, title, year, and authors to narrow down searches. These strategies require query construction skills. Furthermore, search-narrowing decisions are mostly dependent on the user's pre-existing knowledge. For instance, the user must know in advance the authors or the journal of interest or the time period of the publication when he/she is conducting a search. Otherwise, the user cannot get the relevant articles in a short time. Consequently, it is not always clear how to narrow the search to focus on the most relevant articles without manually filtering articles one by one based on their contents. The engines work well if the provider knows exactly what he/she wants and/or has time to go through the information found by the engines. Nevertheless, if the provider knows only vaguely what he/she wants, it would be difficult to come up with precise search criteria. Consequently, the search engines could return a (long) list of documents in response to the user's query, which makes the search for the exact relevant information possible only after sifting through the list one by one [1, 9, 10, 20].

The general-purpose search systems provide flexible query interface with many advanced search options, but produce a long list of matching documents, which the user have to manually review in order to find the answer to his/her query. The special-purpose deep search systems have some drawbacks. They are limited to serve special type of queries that match their objectives. For instance, PubMeth is limited to proteins'

information and relations related only to Cancer, and PPI-finder maintains just information on protein-protein interactions. In addition, the functional of the special-purpose deep search systems' query is limited where the user can only use protein/gene name, drug name, or disease name. Moreover, they use only literature' abstracts, not the full articles that provide more information since they use information extraction techniques to extract the information and the relations.

1.2 Introduction to Relevance Feedback

Relevance feedback refers to an interactive process that helps to improve the retrieval efficiency. In other words, relevance feedback is a strategy of using feedback, implicit or explicit, from previous search result to produce a new search result that is more closely related to what the user wants. Figure 1.3 shows a relevance feedback system. The increased availability of online literature that contain a wealth of knowledge requires using users' feedback to provide enhanced retrieved result and integrate the wealth of knowledge. Studies have showed that the retrieved results can be much improved by providing the user feedback [21-24].

Relevance feedback is widely used to reformulate user query based on rating document as relevant or irrelevant. In traditional relevance feedback technique, the initial query is modified using new words from a previously retrieved top-ranked or identified documents that have been judged for relevance by the user. When a user submit a query, an information retrieval system would first return an initial set of result hits and then ask the user to judge whether some hits, the top-ranked, are relevant or not; after that, the system would reformulate the query based on the user's judgment, and return a set of new results.

Therefore, relevance feedback helps the user to find what he/she is looking for by refining the used search query and naturally guided him/her in the direction of his/her interests.

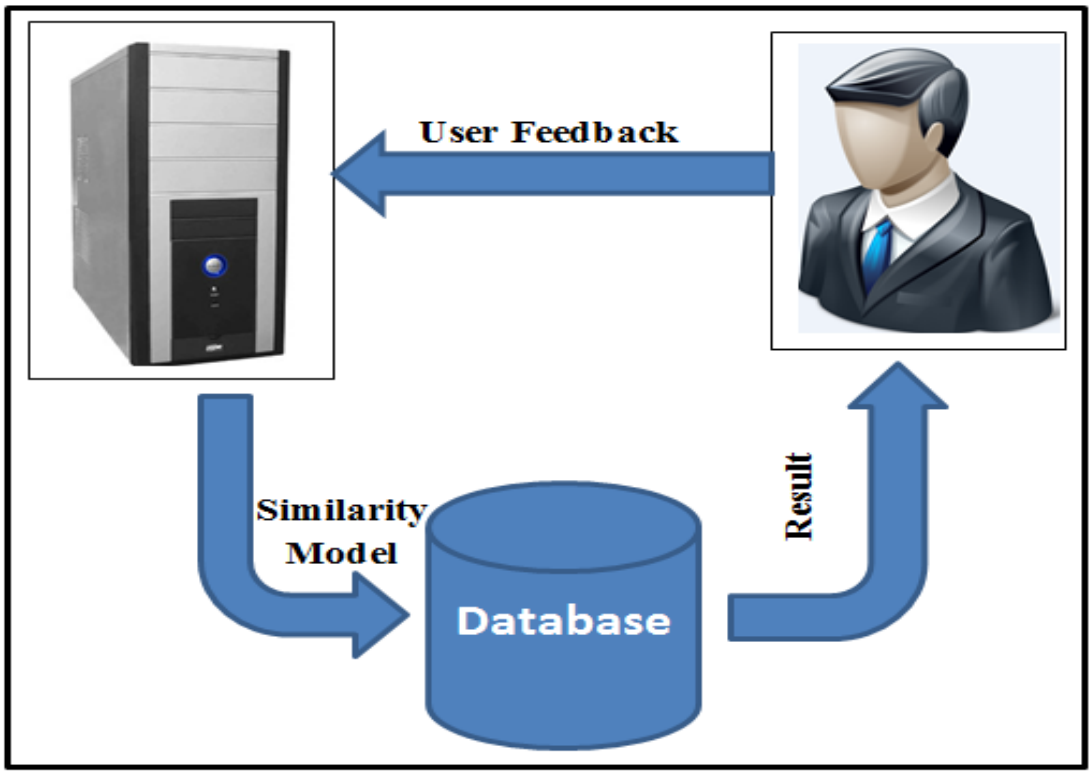

Figure 1.3: Relevance feedback system

Relevance feedback can be classified into three categories: Explicit Relevance Feedback (ERF), Implicit Relevance Feedback (IRF), and Pseudo Relevance Feedback (PRF). In ERF, the users explicitly identify documents as either relevant or irrelevant, while in IRF the user feedback is implicitly obtained from the users' behavior. The PRF do not require user input or evaluation where the initial search is modified based on the most highly-ranked documents in the initial retrieval result set. Relevance feedback technique requires extraction and computation of certain features that can distinguish different elements from the collection and provides much more information than traditional

keywords search techniques [22, 23]. Relevance feedback has been adopted in text retrieval in the form of reformatting the query, such as query expansion either with or without term reweight, personalizing query, and term reweighing. It is also applied to online retrieval system such as video retrieval and e-commerce recommendation where the system monitors the user's preferences [22, 23]. Moreover, relevance feedback technique extensively becomes an essential component for content-based-image-retrieval systems [24].

1.3 Literature Review

This section first provides a literature review of using relevance feedback mechanism in text information retrieval. Then, it presents a background about literature search in the biomedical field.

1.3.1 Literature Search Using Relevance Feedback

In paper [25], the authors applied Pseudo relevance feedback (PRF) technology to enhance retrieval effectiveness where they used explicit relevance feedback to define features and built classification model to predict the performance of Pseudo relevance feedback. The work [26] studied XML query expansion based on Pseudo Relevance Feedback (PRF) technique. It proposed a keyword expansion method based on extended Vector Space Model (VSM), where the good relevant document collection is obtained automatically by search results clustering and the terms with high weight are selected as good expansion terms. It analyzes and applies some features and factors affecting weight for term weight computation. Paper [27] proposed a log-based relevance feedback framework in 3D model retrieval system. It collects users' relevance feedback as a log data

and uses support vector machine as relevance feedback learning method. The framework computes the relevance function on the collected user feedback log-data; next, it combines the relevance information with regular relevance feedback for the retrieval task. The work in [28] introduced a relevance feedback method by adapting a query language model to the topics of documents by using top n retrieved documents. The model adaptation is performed for document retrieval in an online and unsupervised manner. The study [29] proposed to execute relevance feedback on keyword space. The relevance feedback is supposed to work with interactive keyword map system, which visualizes the relationship between keywords extracted from retrieved results

Wang, et al. investigated the problem of negative feedback in language models in [30]. They proposed to exploit negative feedback to improve retrieval accuracy for the difficult query that none of the top-ranked documents is relevant to it. Li and Wang [31] investigated a method to improve the retrieval efficiency by using a query-specific density clustering in the context of information retrieval on the grounds of improved retrieval effectiveness in a fully automatic manner and without relevance information provided by human. The work [32, 33] introduced a general methods that combine the positive and negative relevance feedback techniques based on query feedback model. They used both types of feedback to modify and expand the user's query model. The proposed method takes into account the positive and negative feedback together. Bidok and Moosavi proposed a relevance feedback learning method for query expansion. In this method, the whole set of documents are classified according to existing feedbacks. The documents that are classified with low certainty are chosen to get explicit feedbacks from user; while,

documents that are classified with high certainty are used in query expansion process as implicit relevance feedbacks [34]. In [35] authors introduced an entropy-base query expansion with a reweighting (E_QE) approach. The approach revises the queries during the iterative retrieval process. The approach delivers the users' queries based on their information-seeking behaviors. Paper [36] proposed a relevance feedback retrieval system to improve the searching results. The system is built on the Indri toolkit, using pseudo relevance feedback method.

1.3.2 Literature Search in Biomedical Domain

There are two types of literature search systems. The first type is general-purpose academic search system, which has two subtypes: open-domain search engines, such as Google Scholar and Microsoft Academic Search, and domain-specific search engines such as PubMed. They focus on searching large literature collections to find documents that are relevant to a user's query. The second type is special-purpose deep search systems. They provide pre-extracted information from published literature in biomedical domain such as STRING [37], EDGAR [38], PubGene [39], MeInfoText [40], GOAnnotator [41], and EBIMed [42]. These special-purpose deep search systems use text mining and natural language processing techniques (information extraction) to provide pre-extracted information and relations from the literature. Information extraction, which is a section of natural language processing, extracts information from natural language text. Information extraction technology arose to address effectively the need for efficient processing of texts in specialized domains [43]. Biomedical information extraction focuses on extracting information from biomedical literature and Electronic Medical Records. It has emerged as

a research field due to the increasingly enormous amount of literature published in biomedicine and fast adoption of electronic medical records. The special-purpose deep search systems provide pre-extracted information and relations from the literature in a structured way using natural language processing techniques.

The iHOP (Information Hyperlinked over Proteins) is used to find out a network of protein. It links a given protein or gene name to its corresponding database records. iHOP provides genes and proteins network with hyperlinks extracted from millions of PubMed document abstracts. The iHOP network contains 5 million sentences and 40,000 genes from various organisms [16]. PPI-Finder (Protein-Protein Interaction) finder mines human protein-protein interactions (PPIs) from PubMed abstracts based on their co-occurrences frequencies, and extract the semantic description of interaction from occurring documents. PPI-Finder also combines known protein interactions with co-occurred terms in Gene Ontology (GO) database [44] to infer possible human protein-protein interaction [18]. STRING (Search Tool for the Retrieval of Interacting Genes/Proteins) is a database that predicts protein-protein interactions which can be direct (physical) or indirect (functional) associations. STRING derives the associations from four sources: genomic context analysis, high-throughput experiments data, co-expression data, and previous knowledge mined from literature and databases. STRING integrates interaction information for large number of organisms from the four mentioned sources [37]. EDGAR (Extraction of Drugs, Genes and Relations) is a biomedical system that is used to extract relations and information between drugs and genes relevant to cancer from the biomedical literature. The EDGAR extracts Cancer-related information and relations between drugs, genes, and

cells from abstracts in PubMed database using syntactic and semantic information about related terms found in biomedical literature. The EDGAR uses two existing biomedical sources: Unified Medical Language System and PubMed database [38].

PubGene system provides a graphical protein interaction network based on protein-protein literature co-citations. The system extracts implicit and explicit knowledge from biomedical databases to create gene-to-gene co-occurrence network by analysis of abstracts and titles from PubMed database [39]. MeInfoText system studies epigenetic modifications of the gene methylation and its association with cancer. It provides detailed association information about gene methylation and Cancer using text mining from biomedical literature. MeInfoText offers protein-protein interaction and biological pathway information [40]. PubMeth is another system that studies epigenetic modifications of the gene methylation and its association with cancer. The PubMeth provides genes that are reported as methylated in various cancer types. It uses text mining techniques and manual annotation to provide the association information obtained from PubMed literature abstracts [17]. The GOAnnotator is text mining system that allows text extraction based on GO annotations for a given protein identifier. It links the given protein names to text extracted from PubMed abstracts, which are associated with GO terms. The text selection is based on the similarity between the abstracts and the term. The GOAnnotator provides evidence text in literature for GO annotation of the given proteins [41]. The EBIMed is a text mining system that provides an overview table of co-occurring concepts based on PubMed abstracts for a given query protein. The Table contains other proteins, Gene Ontology (GO) terms, drugs and species [42].

There also has been recently an attempt to fill the gap between the general-purpose search and domain-specific deep search systems in biomedical field. Choi et. al. [45] proposed BOSS that is search engine for biomedical literature. BOSS uses information extraction techniques to index biomedical literature abstracts that are used for the search process. Moreover, researches have recently studied how to improve PubMed search quality using relevance feedback mechanism [21], such as Misearch [46] and RefMed [47]. MiSearch [46] is an online biomedical literature search tool that ranks citations by using implicit relevance feedback. It uses user click-through mechanism as user's feedback for identifying terms relevant to user's information need. RefMed [47] is another biomedical literature search tool that uses relevance feedback, machine learning and information retrieval techniques. It retrieves search results based on user queries then asks him/her to provide his/her feedback on relevant documents. Next uses the user's feedback to learn RankSVM ranking algorithm.

1.4 Research Objectives

Reducing the number of relevant documents for the healthcare provider to read within a very short period of time is indispensable for on-the-spot searching. One way for the reduction is to use more keywords in the search criteria; however, this can lead to related documents to be removed by the search engine. Another way to improve the retrieval accuracy is via the relevance feedback technique [21, 48], which is a strategy of using user's feedback, implicit or explicit, on previous search result to produce a new search result that is more closely related to what the user wants [21, 22]. My goal is to investigate and develop a search system using relevance feedback methodology for

biomedical literature search within a very short period of time (e.g. at the point of care) to increase the retrieval result accuracy. The main objectives are for this system to have the ability to analyze automatically a large number of retrieved documents and generate automatically short and coherent hits using the relevance feedback approach, Unified Medical Language System (UMLS®) [49], and information extraction mechanism, which is part of text mining techniques where information and relation are extracted from machine-readable literature [50]. The UMLS is a set of medical term database and software that classify and define the health and biomedical language developed by U.S. National Library of Medicine. The proposed biomedical literature search system will utilize UMLS to provide a higher rate of relevant articles for the subject that physicians are interested in within a very short period of time and thus reduce user's time and effort by filtering out less relevant articles. The relevance feedback mechanism will be conducted automatically with minimum user interaction; where the user needs only to provide whether the documents are relevant to his/her preference or not, he/she does not go further in details such as, why and what.

1.5 Original Contributions

My main contributions are as follows:

- A biomedical literature search system that uses relevance feedback mechanism, fuzzy logic, text mining techniques and Unified Medical Language System has been developed [51, 52]. The biomedical literature system is developed to assist healthcare providers to find more related documents using relevance feedback mechanism. It extracts meaning and semantic relations between texts to calculate

and find the similarity between documents. The biomedical literature search system extracts and decodes information from the documents and uses the extracted information to filter unwanted documents and ranks the related ones based on the user preference. The developed biomedical system has the ability to search large document set and find the most related articles based on the user preference.

- An approach that extracts PDF features and use these features to filter unwanted documents with the help of type 2 fuzzy logic has been developed [51]. The extracted internal features can provide an appropriate way that filters unwanted and unrelated documents and then ranks the documents in a great manner that is more close to what the user wants.
- A fuzzy-based ranking approach has been developed [52]. The ranking approach uses fuzzy logic, text mining techniques and Unified Medical Language System. The ranking process is utilized based on fuzzy logic and Unified Medical Language System knowledge resources. The fuzzy ranking approach uses semantic type and meaning concepts to map the relations between texts in documents.

These contributions make the proposed biomedical literature search system unique, which are: utilize the full articles that provide more information than the articles' abstracts, employ information extraction technique to extract the hidden information and relations between texts, and use the relevance feedback strategy in the search process that increases the retrieval result accuracy. I use the full-text document rather than the abstract of the document because I believe that more information is presented in the documents main body sections such as, Result and Discussion; while the document's abstract provides short

description of the important information. Since the information extraction technique locates and extracts the hidden information and relations among the texts, I employ this technique to extract automatically semantic and meaning relations between text documents and therefore search for similar documents. Moreover, I involve the relevance feedback mechanism in the search process to improve the search retrieval accuracy. By using the user feedback from previous search result, more accurate and related search result is provided to him/her based on his/her own preference.

To the best of my knowledge, the proposed biomedical literature search system is the first biomedical literature search system that extracts semantic relation (semantic meaning and type) between texts and at the same time uses relevance feedback mechanism. Moreover, it uses fuzzy logic methodology and text mining techniques in biomedical literature retrieval, ranking and search process. However, both of previously mentioned tools (RefMed, and of Misearch) use relevance feedback technique in biomedical literature retrieval search process and ranking, they are lack of extracting the semantic relation (semantic meaning and type) between texts using Unified Medical Language System knowledge sources. I believe UMLS provides more reliable retrieval result in the field of biomedical and life science.

1.6 Introduction to Fuzzy Logic

Lotfi A. Zadeh introduced the concept of Fuzzy logic and sets in 1965 [53] which is a mathematical logic for dealing with uncertainty. In practice, fuzzy logic means computation with words. Fuzzy logic provides a way for representing existing of imprecision and uncertainty such as in the language when giving description as hot, cold,

low, high, short, and tall. In other words, fuzzy logic provides a mechanism to model the uncertainty associated with lack of information and vagueness. For example, when we describe a person as tall person; what is the boundary of that description? Is a 6 inches person can be described as tall? What is about 5.9 inches person; is he/she also tall or short? Therefore, there is not unique definition of language description. In fuzzy, the variables such as Temperature, and Humidity are called in fuzzy linguistic variable [54]. The linguistic variable contains a set of linguistic terms, which represent the values that may take the linguistic variable at different states [55]. Figure 1.4 illustrates a fuzzy membership variable "Temperature" and its linguistic terms "cold, warm, and hot". The linguistic terms are assigned fuzzy set. Fuzzy logic uses membership functions to deal with the uncertainty and vagueness. The fuzzy sets have a membership degree from zero to one.

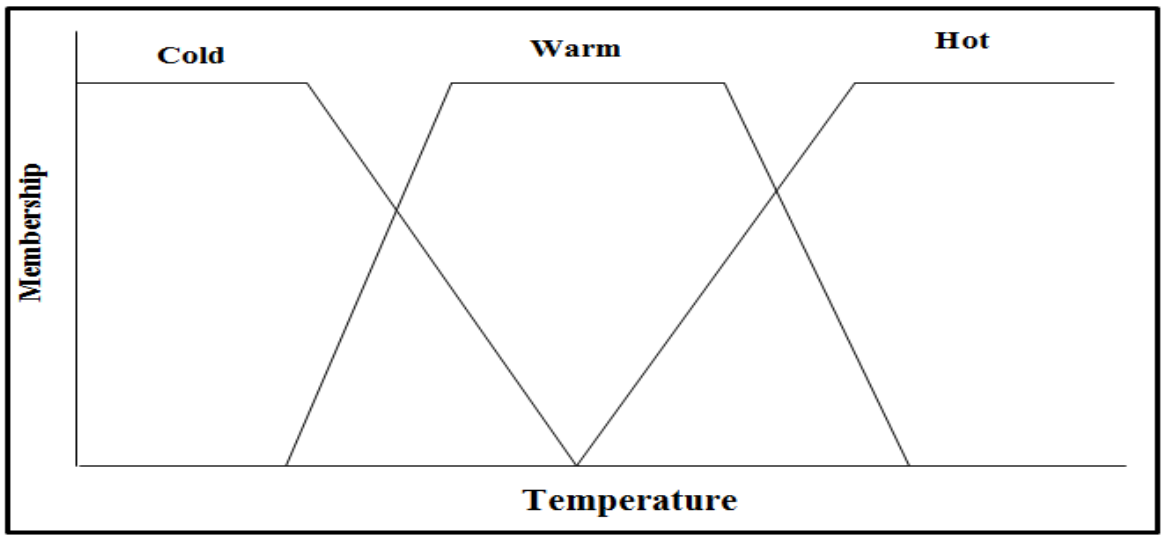

Figure 1.4: Fuzzy membership function

Fuzzy system, fuzzy inference system, consists of a set of rules with an "if – then" structure, fuzzification, defuzzification. The purpose of fuzzification process is to convert the crisp input value to a fuzzy membership, and defuzzification process is to convert the

fuzzy output membership value to a crisp value. Figure 1.5 represents a fuzzy inference system structure.

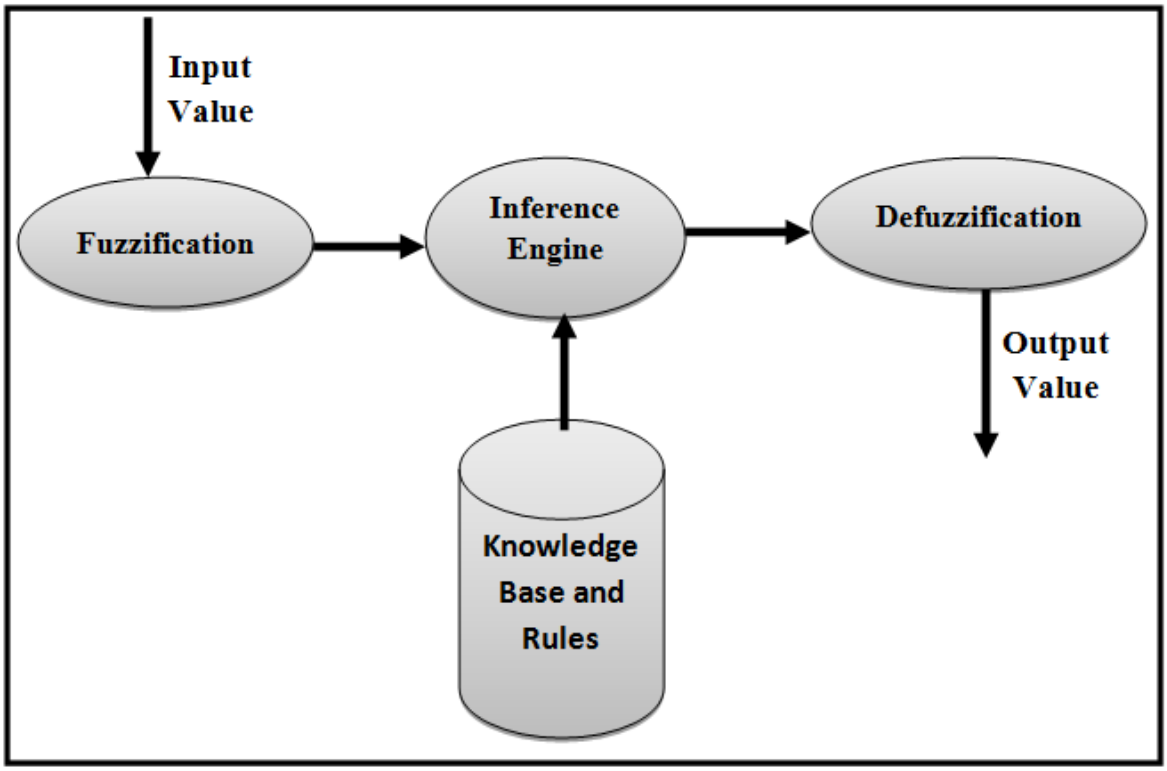

Figure 1.5: Fuzzy inference system (FIS)

The concept of type-2 fuzzy sets was originally introduced by Zadeh in 1975 [54]. Type-2 fuzzy sets are characterized by a fuzzy membership function where the fuzzy degree of membership is a type-1 fuzzy set and there is uncertainty about the membership grades themselves. In other words, the characterization of a type-2 fuzzy set allows it to have an upper membership function (UMF) and a lower membership function (LMF); these functions are type-1 fuzzy set membership functions. The interval between upper and lower functions represents the footprint of uncertainty (FOU), which is used to characterize type-2 fuzzy set. Figure 1.6 shows a triangular interval type-2 membership function. The

embedded fuzzy set is type-1 fuzzy set [56]. A type-2 fuzzy set can be useful when uncertainties are present [54, 57-60]. Most applications use interval type-2 fuzzy sets in type-2 fuzzy logic systems because of the computational complexity of using general type-2 fuzzy sets.

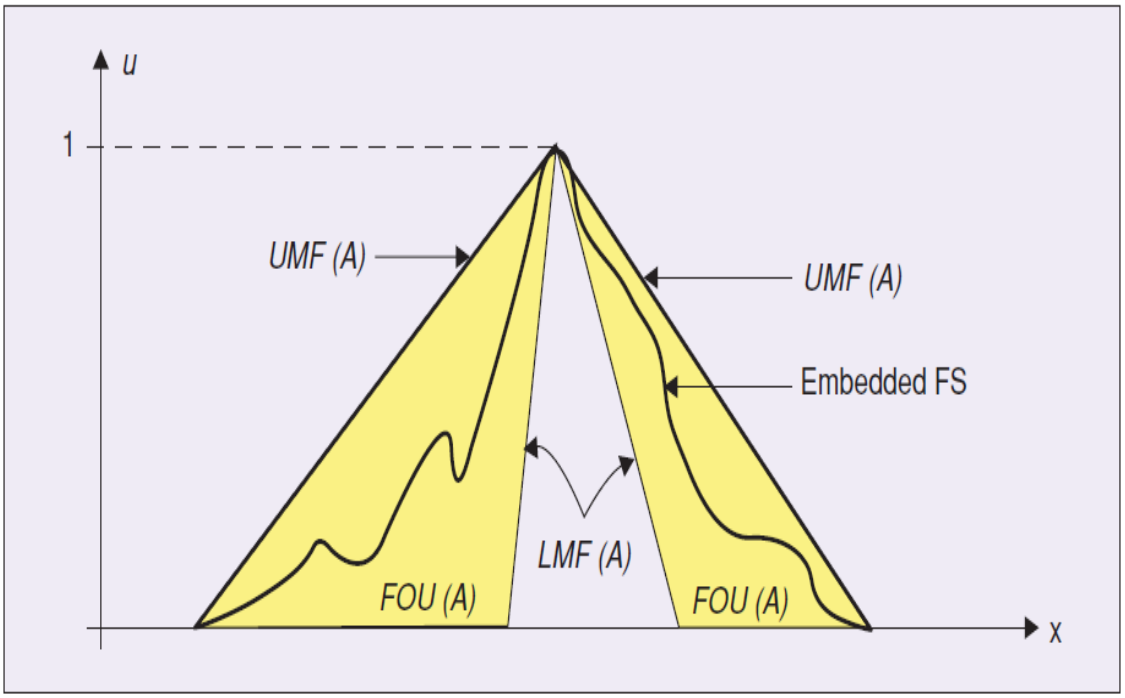

Figure 1.6: Triangular interval type-2 fuzzy membership function [56]

The computations associated with interval type-2 fuzzy sets are easier to perform, which makes an interval type-2 fuzzy logic quite practical [56, 59]. Figure 1.7 depicts Type-2 fuzzy logic system that contains four components: Fuzzifier, Rules, Inference Engine, and Output Processing. The crisp inputs (x) are first transformed into fuzzy input sets in the Fuzzifier. The obtained fuzzy input sets are then mapped into fuzzy output sets by the Inference Engine based on the applied IF-THEN rules from the Rules. The fuzzy output sets are then reduced to fuzzy type-1 sets by the Type-Reducer. Finally, the

generated type-1 sets are transformed into crisp outputs (y) by the Defizifier. The system produces two groups of outputs, which are crisp, and fuzzy type-1 set outputs. It is dependent on the application to use either of them; however, most of the engineering applications require crisp output [56].

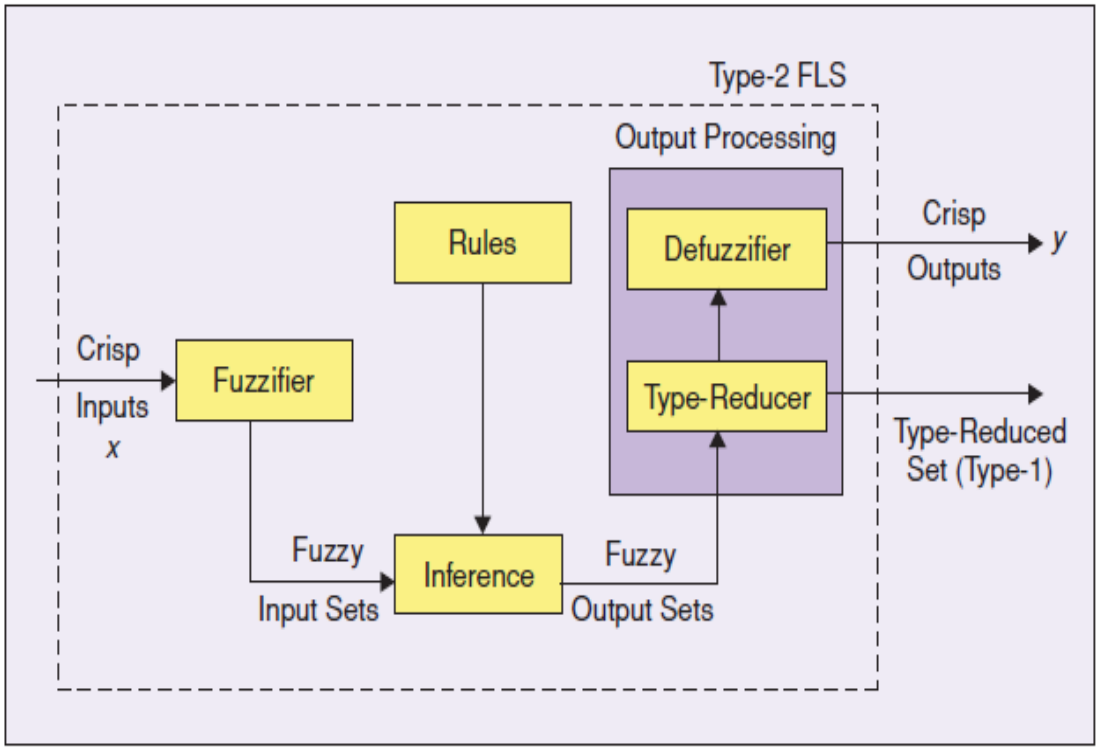

Figure 1.7: Type-2 fuzzy logic system [56]

1.7 Introduction to Unified Medical Language System (UMLS)

The U.S. National Library of Medicine started the Unified Medical Language System (UMLS) in 1989 as an attempt to accomplish and achieve wide and complete image of the medical knowledge and to connect the individual vocabularies among each other. The UMLS consists of three knowledge resources that fulfill different functions, which are a Metathesaurus, a Semantic Network, and a SPECIALLIST Lexicon. The

Metathesaurus stores the concepts, the Semantic Network holds all categories and relations for the concepts and the SPECIALLIST Lexicon generates indexes to Metathesaurus. The Metathesaurus is a multi-purpose large vocabulary database contains information about biomedical and healthcare related concept, their various names, and the relation among them. It is built from the electronic versions of many different thesauri, classifications, code sets, and lists of controlled terms used in patient care, health services billing, public health statistics, indexing and cataloging biomedical literature, and/or basic, clinical, and health services research. It contains more than 2 million concepts, their names, and other attributes from more than 100 different source terminologies, classifications, and thesauri. Each concept in the Metathesaurus that represents a meaning (sense) has its unique and permanent concept identifier (CUI) and is linked to the other two knowledge sources, which provide additional information [49].

The main purpose of Semantic Network is to afford a set of useful and additional information about relationship between all concepts represented in the UMLS Metathesaurus and to provide a consistent categorization for all of them. The Semantic Network contains more than 130 semantic types and 54 relationships. Where a semantic type is a broad subject categorization assigned to a CUI and a semantic relation is the relationship between two semantic types. Each concept in the Metathesaurus is assigned at least one semantic type. The SPECIALIST Lexicon includes biomedical and common English vocabularies. For every term in the lexicon, the syntactic, morphological, and orthographic information are recorded. This information is necessary for the SPECIALIST natural language processing system. The lexical tools use the SPECIALIST natural

language processing system to normalize strings, index words and find lexical variants [49]. Therefore, the UMLS is more than just a dictionary of different word meanings but also a framework encoded with different semantic and syntactic structures such as information includes related concepts, semantic types and semantic relations. Hence, it can be used for analyzing the biomedical text. The UMLS knowledge sources have been applied to several biomedical information extraction areas, such as query expansion [61], classification [62, 63], organization of search results [64], matching patient records to relevant articles [8], and medical question answering [65].

1.8 Dissertation Outline

The following is the outline of the remaining chapters of this dissertation. Chapter 2 provides the overall relevance feedback-based biomedical literature search system design and architecture. This chapter describes the main components of the system. Chapter 2 also presents in details how the extracted PDF features are used to narrow the search result with the help of fuzzy logic. Moreover, chapter 2 describes how the relevance feedback mechanism, Unified Medical Language System, fuzzy logic, and text mining techniques are utilized in the search process.

Chapter 3 provides the relevance feedback-based biomedical literature search system evaluation, which involved two domain experts. It describes the experiments design and shows how the data set is created. Furthermore, chapter 3 presents the results and illustrates the effectiveness of using relevance feedback mechanism, UMLS, fuzzy logic, and text mining in the search process and relevancy ranking. In chapter 4, a fuzzy logic-based ranking approach is developed under UMLS knowledge

sources to rank biomedical literature search result. Chapter 4 explains how the fuzzy logic-based ranking method is utilized using UMLS meaning and semantic type features. It also shows the usefulness of using the developed ranking method compared with other ranking methods. Finally, Chapter 5 concludes this dissertation and provides the future directions.

CHAPTER 2

THE OVERALL RELEVANCE FEEDBACK-BASED BIOMEDICAL LITERATURE SEARCH SYSTEM DESIGN AND ARCHITECTURE

This chapter introduces the overall relevance feedback-based biomedical literature search system design and architecture. It also provides a description of the main system components and their implementation. Moreover, it explains in details how the fuzzy logic and sets, UMLS, and text mining techniques are utilized in the search process and ranking the relevancy.

2.1 Overall Literature Search System Design

Figure 2.1 shows the overall biomedical literature search system design. The overall literature search system consists of two main components or units: 1) fuzzy logic-based PDF features extraction component, and 2) relevance feedback-based literature search component. The goal of the first component of the system is to search the user initial query and retrieve the matched documents from online repositories. Then, narrow the number of returned documents to the user by filtering out less related documents. Whereas the goal of the second component is to use the user's selected documents from the initial result as relevance feedback to improve the retrieval result. The first component of the system consists of two processes: 1) search and retrieve documents, and 2) document filtering.

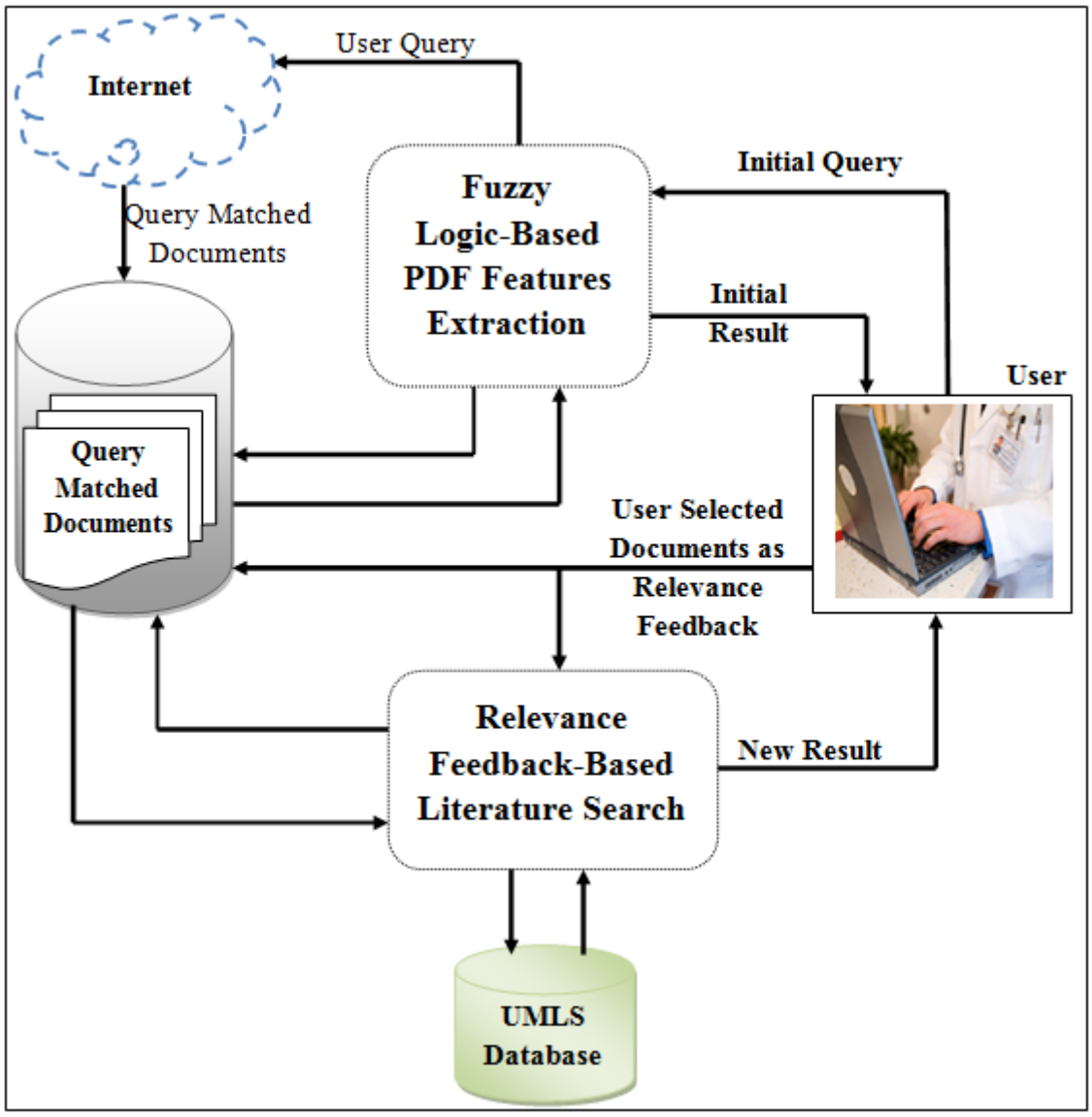

Figure 2.1: The overall biomedical literature search system design

The first process sends the user's initial query and retrieves the matched documents from distributed online repositories. The second process narrows the number of provided documents that match the user's query as initial result to the user. The user reviews the initial search result obtained from the first component. If he/she does not find what he/she is looking for, then he/she selects some documents from the previous search result and uses

them as relevance feedback and search again by second component. The second component of the system (relevance feedback-based literature search) uses relevance feedback mechanism, text mining technique, and Unified Medical Language System (UMLS) to perform the search process.

2.2 Main Component 1 - Fuzzy Logic-Based PDF Features Extraction

This section introduces the first component of the system that extracts PDF features to narrow the literature search result with the help of type-2 fuzzy logic [51]. This component tries to provide an accurate and short result in short time according to healthcare providers' query by automatically examining large number of electronic biomedical literatures. The process requires document identification, document filtration, and text extraction. Since several PDF features are extracted to differentiate between the documents, these features may have different weights. Moreover, the provided keywords may also have different importance. Therefore, a special technique that can provide different levels of weight is used. Weights obtained by polling a group of experts will often be different for the same task because the experts will not necessarily be in agreement; this indicates the presence of uncertainty in getting experts' opinions. While type-1 fuzzy logic can handle uncertainty, type-2 fuzzy logic may be advantageous in handling more complex uncertainty [56, 59, 66]. The extra available dimensions in type-2 fuzzy logic operations give more degrees of freedom for possible better representation of uncertainty. Therefore, type-2 fuzzy sets have the potential to provide better performance than type-1 fuzzy sets in decision-making. Hence, type-2 fuzzy logic was used.

The following section reports how this component is designed and how type-2 fuzzy logic is utilized in the search strategy, which involves extracting features in the PDF full-text to narrow the search result (i.e., searching in the Result Section of the article and determining whether the article contains a figure and/or a table). The technique, including the fuzzy module, is implemented using Java programming language. Unlike most search engines, which are interactive in nature, the system is designed to function automatically once it is fully developed. Therefore, it can be employed for literature monitoring, one of the eventual targeted application areas of the technique.

2.2.1 Design

The fuzzy logic-based PDF features extraction component consists of two processes: (1) searching and retrieving articles in PDF format, and (2) filtering the articles to narrow the end result. Figure 2.2 illustrates the fuzzy logic-based PDF features extraction component and shows how its processes are connected with each other and with the other components. It also demonstrates how the user interacts with the system by sending the query and retrieving the result. The searching and retrieving process (PubMed's search engine) composes of two stages, which are searching PubMed database and retrieving full-texts in PDF format from distributed databases. The PubMed's search engine (searching and retrieving) is the first process of the fuzzy logic-based PDF features extraction component. Importantly, this engine is not the interactive search engine provided on the PubMed web site. Rather, it is a Java program that was implemented to remotely access and search, via the Internet, the PubMed database and automatically retrieved the citations. It also downloads the PDF full-texts in batch mode from the related

online databases and repositories. It takes search criteria as input and provides articles in PDF format as output. The aim of this process is to prepare the PDF articles for the next process of this component.

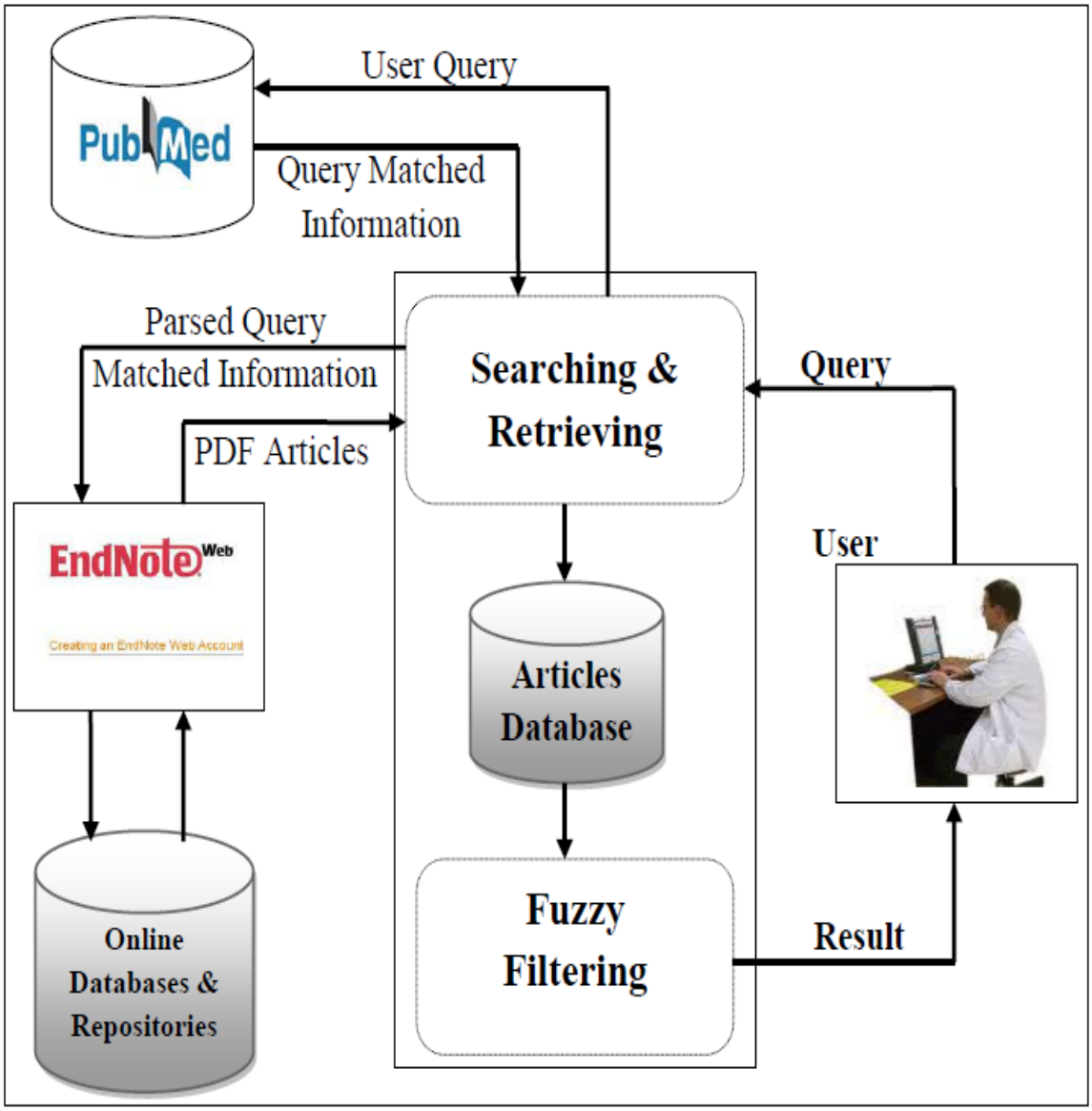

Figure 2.2: The fuzzy logic-based PDF features extraction component

The second process of the fuzzy logic-based PDF features extraction component is fuzzy filtering; it extracts and decodes features embedded in the PDF documents using user

parameters and search criteria. Figure 2.3 depicts the fuzzy filtering sub-system design. The filtration process is performed in stages followed by the fuzzy ranking process. One of the important roles of the fuzzy filtering process is to decode explicit and implicit PDF features, which are used for filtering and sifting unwanted articles. PDF articles contain several features that are used to differentiate between them, and they have different levels of importance. Hence, weights for the features are calculated. Some of the features do not exist in the existing search engines, which include (1) medical cues and outcomes expected to be in the Results Section of an article, (2) Boolean AND and OR operators for the cues and outcomes, (3) weights to keywords and features, and (4) finding if the article contains a table and/or a figure. In the end, the PDF articles are ranked according to their relevancy to the user search criteria. The ranking process of the article's relevancy to the user search criteria is explained later in more detail.

The filtering process is utilized using a combination of fuzzy logic and text mining techniques. The search algorithm of the fuzzy filtering process is designed in such a way that the search is performed according to features in the PDF articles. The fuzzy filtering system consists of three fuzzy subsystems, which are fuzzy keyword weights, fuzzy feature weights, and fuzzy relevancy ranking that indicates how each article is relevant to the user's search preference and the degree of the relevancy. Weights for the same keyword and feature obtained from different experts are often different because experts do not necessarily agree with one another. In other words, there exists uncertainty within the obtained weights of the keywords and features. Therefore, interval type-2 fuzzy sets is

used to represent the fuzzy weight of each keyword and calculate the weight for each feature.

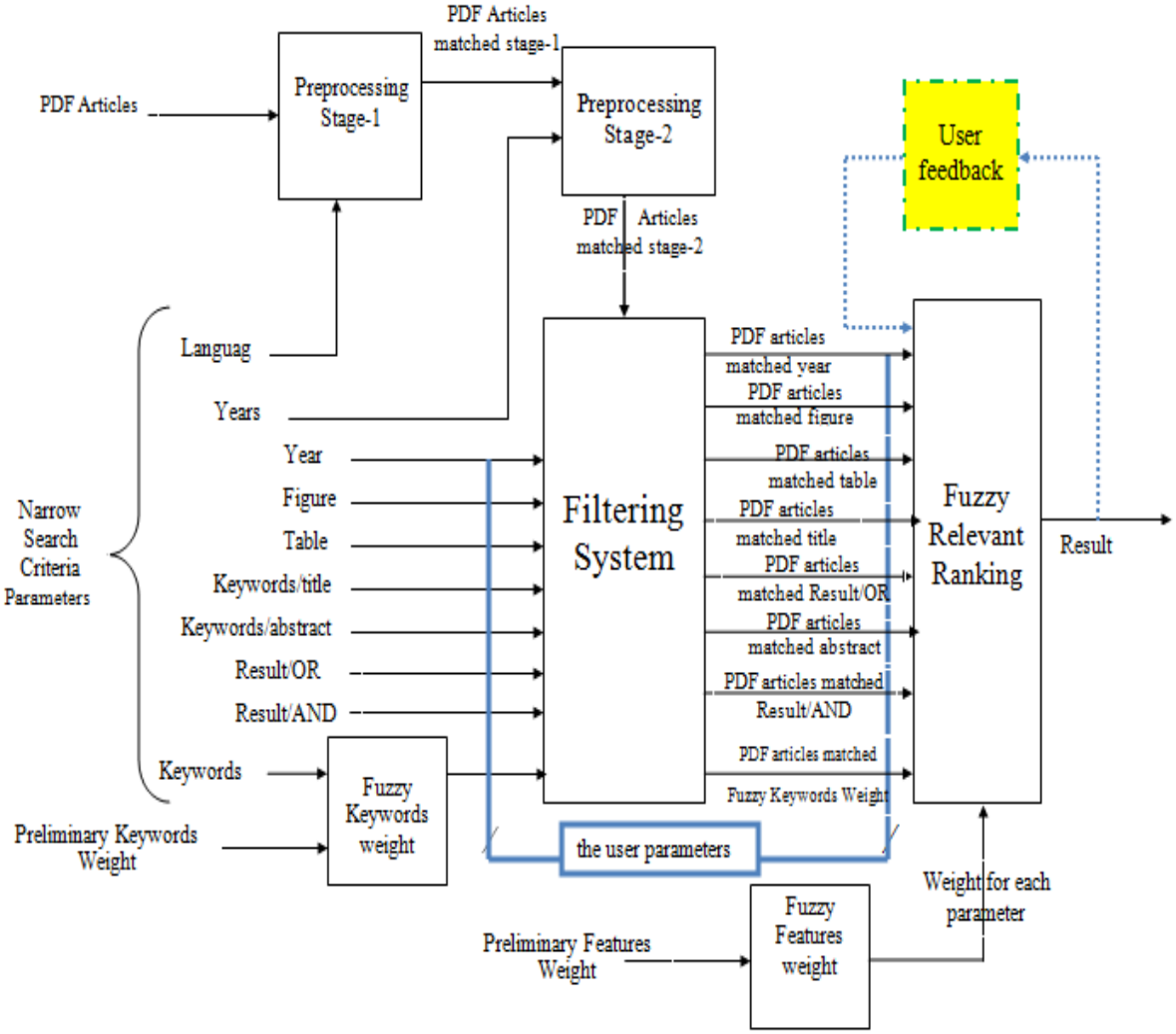

Figure 2.3: Fuzzy filtering sub-system design

The following features in the PDF articles are used to allow the user narrows his/her search result:

- Search the desired information in the Results Section of an article. For a well-organized article, the Introduction Section should provide background information about the main topic, whereas the Method Section should mainly focus

on the methodology of the study. The Results Section and the optional Discussion Section should more likely contain the information that the healthcare provider is looking for - clinical findings of the study [8, 13].

- Determine whether the article contains a table or a figure. Tables and figures are features that can be used to distinguish between documents. Their existence can provide some hints concerning whether certain useful information that the user seeks exists in the article.

The fuzzy relevancy ranking system determines the degree of the articles' relevancy by calculating the cumulative weight for each article, which is the sum of all the weights for the Boolean features, the search criteria parameters that the article satisfies and the weight of the fuzzy keywords feature multiplied by the keywords frequency range of the article as in equation (2.1). The total fuzzy keywords frequency feature of an article is computed by equation (2.2), while the keywords frequency range of an article is calculated based on its total fuzzy keywords frequency.

$$CW = \sum_{i=1}^{n} w_i * Fa_i + w_{fk} * Kf_r \quad (2.1)$$

Where CW refers to cumulative weight, n is the number of Boolean features, W_i represents the weight of i-th Boolean feature, and Fa_i is the feature occurrence factor of the article for the i-th Boolean feature, where $Fa_i \in \{0,1\}$ (it is 1 if the feature appears in the article and 0 otherwise). In addition, w_{fk} is the weight of the fuzzy keywords feature, and Kf_r is the keywords frequency range of the article.

Equation (2.2) explains how to calculate the total fuzzy keywords frequency T_{FK} factor of an article. This factor, total fuzzy keywords frequency, is then used to compute the keywords frequency range Kf_r of the article.

$$T_{FK} = \sum_{j=1}^{m} t_j * wk_j \quad (2.2)$$

Here T_{FK} refers to the total fuzzy keywords frequency of the article, m is the number of keywords, t_j presents the frequency of the j-th keyword, and wk_j represents the fuzzy weight of the j-th keyword.

2.2.2 Implementation

This component was built using JAVA programming language and was tested under Linux and Windows 7 operating systems using a machine that has 6 gigabytes RAM, 850 gigabytes hard drive, and a 2.7 GHz Pentium dual-core processor. A Java program was built to implement the searching stage of the searching and retrieving process of the fuzzy logic-based PDF features extraction component of the system. The program provides remote interface connection to the National Center for Biotechnology Information E-utilities. Figure 2.4 shows the implemented graphic user-interface of the search stage for PubMed's search engine process of the fuzzy logic-based PDF features extraction component. The E-utilities is a collection of web-based programs that provides a remote programming interface to the Entrez system, a National Center for Biotechnology Information's primary text-based search and retrieval system. It integrates the PubMed database and 39 other biomedical literature databases [4]. The system sends queries to

PubMed via E-Utilities' ESearch service, and retrieves citations via E-Utilities' EFetch service.

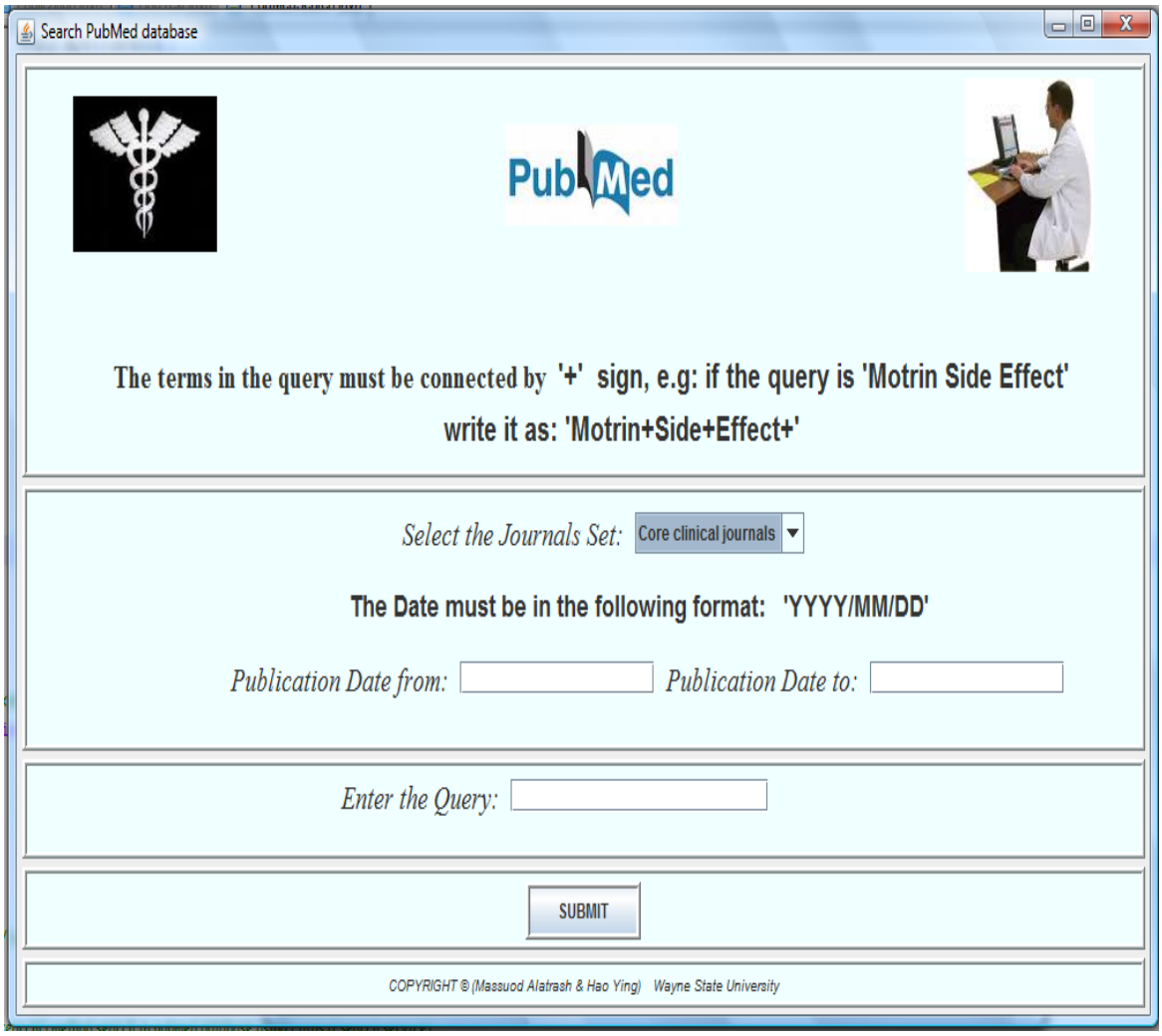

Figure 2.4: The PubMed search engine process graphic user interface

This stage of the system acted as glue between the user environment and the World Wide Web using remote access. It extracts the complete citation automatically from the PubMed database, including authors' name, citation title, journal name, content of abstract, keywords, volume number, issue number, issue date, and page numbers without user interaction during the operations. In addition, Medical Subject Heading (MeSH) terms is

used to provide more accurate search result. Medical Subject Heading terms are in the National Library of Medicine's controlled vocabulary. It is used to ensure a consistent way to retrieve information that may use different terminologies for the same concept, yielding a more consistent citation set [4]. More specifically, the ESearch server is used, which searches PubMed for the records that match the provided query and returns some value attributes, such as Query_key and Web_Environment. Also the EFetch server is used, which retrieves the requested records using the returned value attributes of the ESearch server as a list in the requested format. The retrieved citation information was then saved in a structured text file to be used for retrieving the citations' PDF full-texts.

The second stage of searching and retrieving process (retrieving full-texts) retrieves the full-texts in PDF format automatically in batch mode. For this purpose, EndNote X3 is used, a popular reference management software package, to automatically download and organize the articles in PDF format. EndNote X3 has the ability to import a large amount of citation information as a text file and automatically search for their corresponding PDF files from various online repositories and databases in batch mode [67]. The text file containing the citation information obtained from the PubMed database was imported to EndNote X3 and configured using a customized filter implemented for this purpose to match the citation information with the EndNote X3 reference fields. After the import, the citation information is automatically filled into the appropriate fields of the references in EndNote. Then, the PDF full-texts of the citations that are either available online free or Wayne State University library is subscribed to were automatically downloaded to EndNote in batch mode.

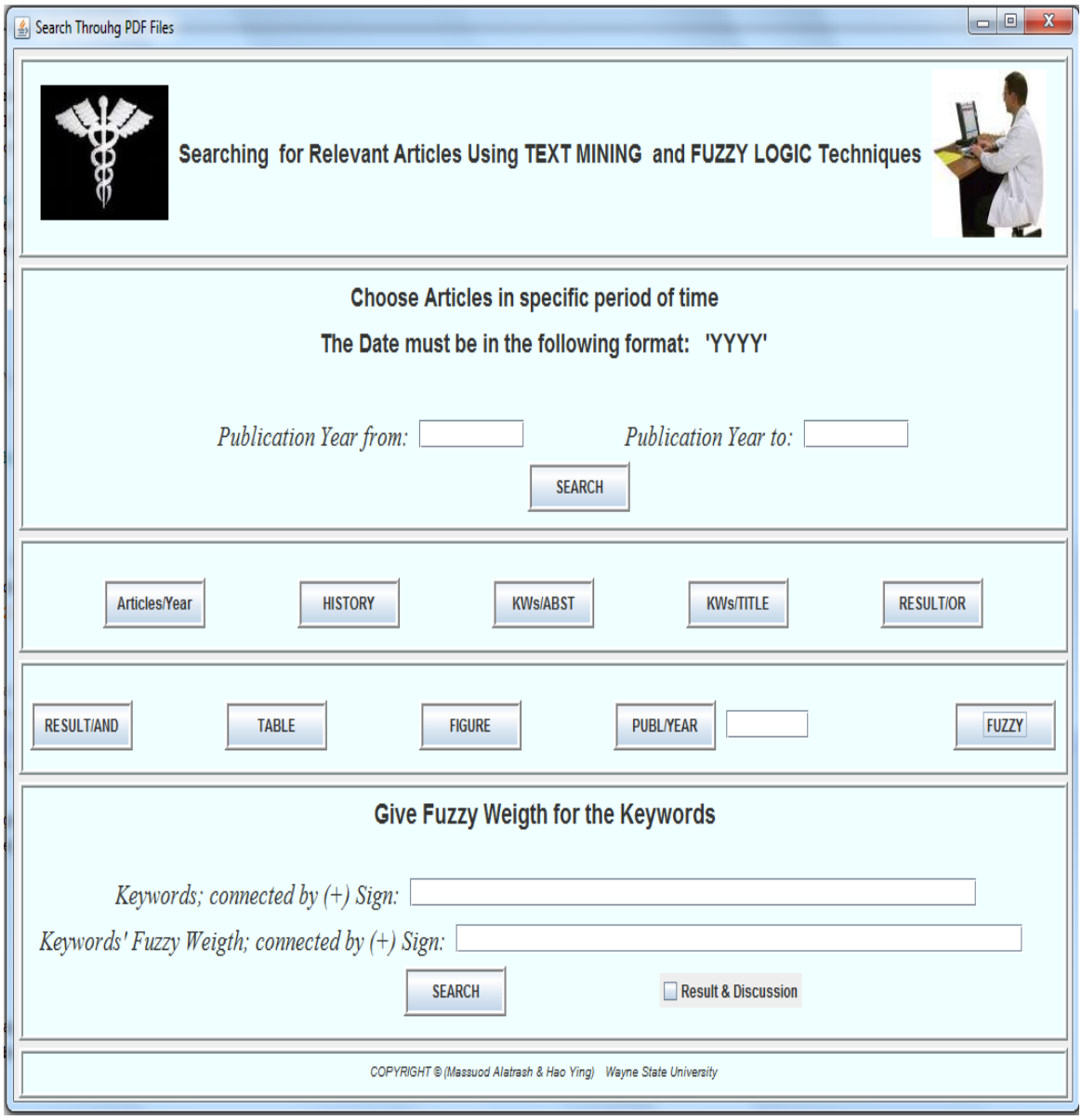

Figure 2.5: The fuzzy filtering process graphic user interface

The fuzzy filtering process of the fuzzy logic-based PDF features extraction component of the system was implemented by utilizing a Java model that searches for relevant articles using text mining and fuzzy logic techniques. The IDRsolutions JPedal (Java PDF Extraction Decoding Access Library), a PDF Java open source library, [68] is used to extract the PDF articles and to make them searchable. The implementation of the

filtering process is performed in such a way that facilitates the user interaction with the technique. For instance, the user can choose to execute each feature by clicking a button; she/he also can select one of the two options: (1) to search the whole document, or (2) to search only the Results Section and/or the Discussion Section by checking a box. The system provides a graphic user interface that simplifies user interaction with the search algorithm as shown in figure 2.5. The system is also able to provide a distribution of the PDF articles over a time period chosen by the user in tabular form. The search history is also provided to the user.

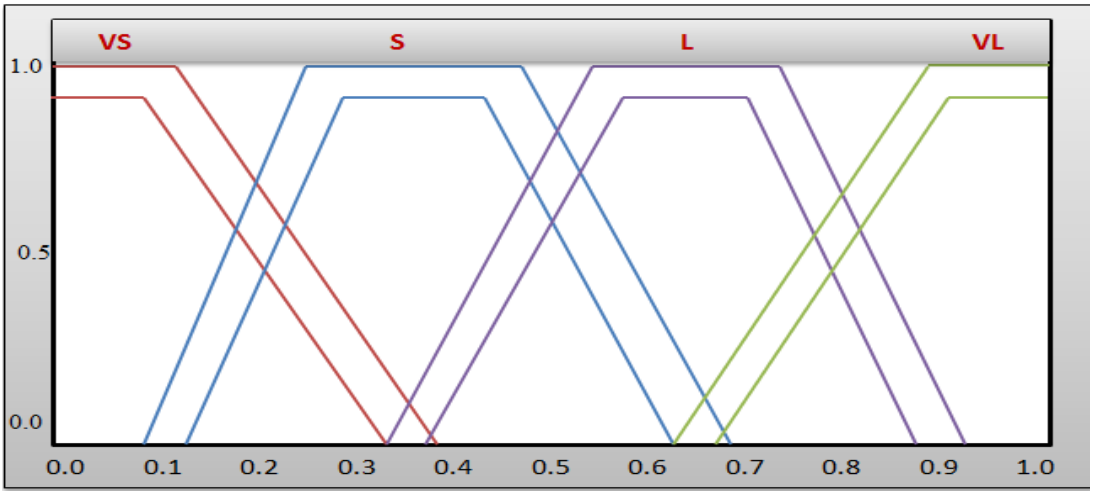

Figure 2.6: Four interval type-2 fuzzy sets for fuzzy keyword weights

The fuzzy sub-system is implemented using MATLAB where interval type-2 fuzzy sets were used. Figure 2.6 shows the interval type-2 fuzzy sets that were used to calculate the fuzzy keyword weights obtained from experts. In the figure, VS means Very Small, S represents Small, L indicates Large, and VL represents Very Large.

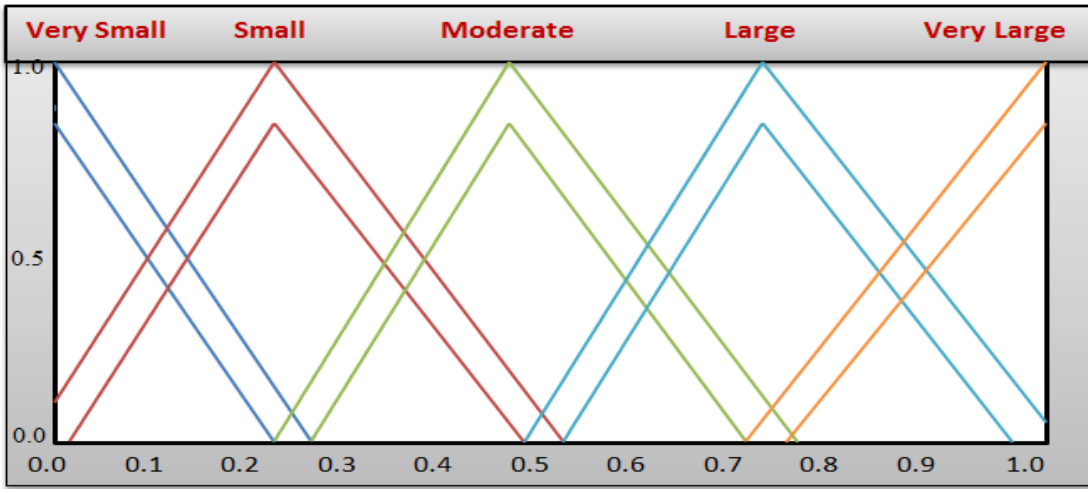

Figure 2.7: Interval type-2 fuzzy sets for relevancy ranking.

Also five type-2 fuzzy sets were used to rank the relevancy of an article- Very Small, Small, Moderate, Large, and Very Large as shown in figure 2.7. In the fuzzy feature weights part, the following eight type-2 fuzzy sets were used to calculate the weight for each feature: Extremely Small, Very Small, Small, Very Moderate, Moderate, Large, Very Large, and Extremely Large. Seventeen fuzzy rules were used in total for the fuzzy filtering sub-system - four for calculating the keyword weights, eight for calculating the feature weights, and five for ranking the relevancy of the articles. Min-Max was used for fuzzy inference, the centroid type-reducer for reducing the type-2 fuzzy set to a type-1 fuzzy set, and the centroid defuzzifier. Furthermore, the fuzzy filtering sub-system uses narrow search criteria, which helps the user to narrow the search result using text-mining techniques. Moreover, a graphic user interface was designed and implemented that assists the user to specify fuzzy weight for each feature as shown in figure 2.8.

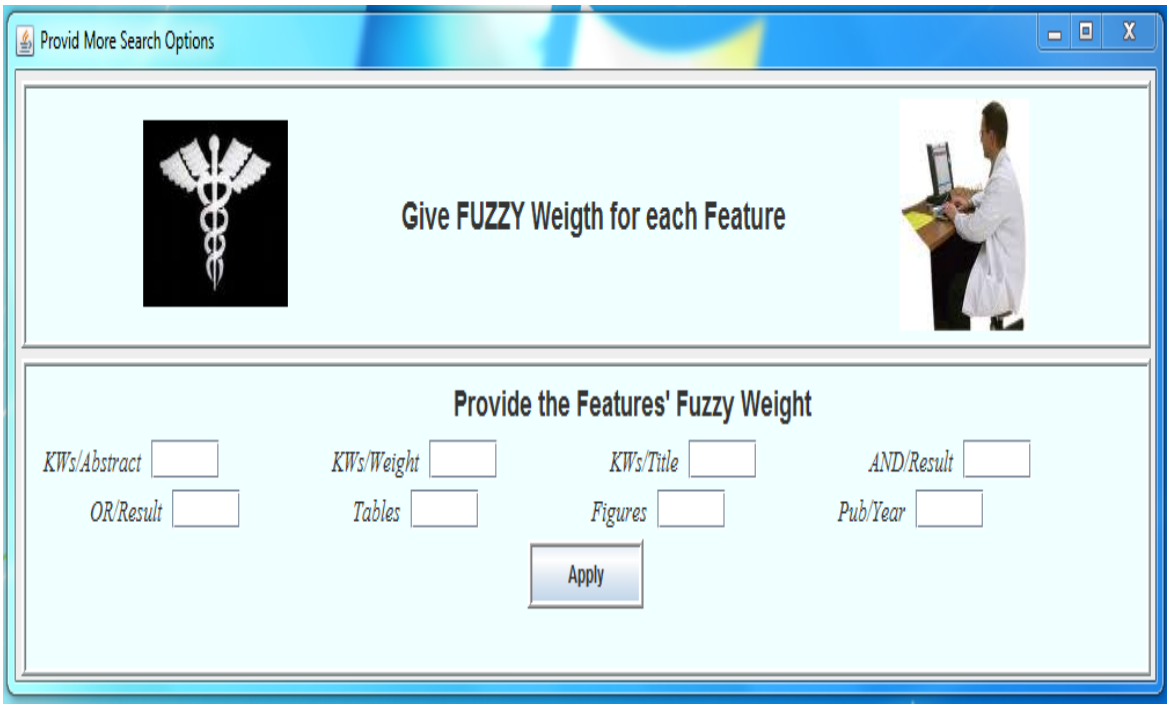

Figure 2.8: The features' fuzzy weight graphic user interface

The following fuzzy rules were used for the fuzzy relevancy ranking system:

- If cumulative weight is *Very Large*, the article is *Very Highly Relevant.*
- If cumulative weight is *Large*, the article is *highly Relevant.*
- If cumulative weight is *Moderate*, the article is *Moderately Relevant.*
- If cumulative weight is *Small*, the article is *Lowly Relevant.*
- If cumulative weight is *Very Small*, the article is *Very Lowly Relevant.*

2.3 Main Component 2 - Relevance Feedback-Based Literature Search

Relevance feedback technique is one way to improve the retrieval accuracy of a search system. Relevance feedback is a strategy of using user's feedback, implicit or explicit, on previous search result to produce a new search result that is more closely

related to what the user wants [21, 48]. This section describes the relevance feedback-based literature search component of the biomedical literature search system. This component extracts automatically semantic and meaning relations between text documents using UMLS. It conducts the search using relevance feedback mechanism, fuzzy logic, text mining techniques, and Unified Medical Language System (UMLS®) knowledge sources version (2011AA) [49]. The relevance feedback mechanism is conducted automatically with minimum user interaction; where the user needs only to provide whether the documents are relevant to his/her preference or not, he/she does not go further in details such as, why and what. Later section reports how the similarity between documents is calculated and ranked and explains how the relevant documents are ranked with the help of fuzzy logic.

2.3.1 Architecture and Methodology

This section describes the architecture and methodology of the relevance feedback-based literature search component of biomedical literature search system.

2.3.1.1 Architecture

Figure 2.9 depicts the architecture of the relevance feedback-based literature search component of the biomedical literature search system of this study and shows how the sub-components interact and connect with each other. This component consists of three units: lexical analyzer, UMLS mapping, and fuzzy & text mining relevancy representation units. In addition, it connects to several databases containing UMLS and literature document set. The UMLS database contains the UMLS knowledge sources. The lexical analyzer unit provides several natural language processing techniques for text normalization. The UMLS

mapping unit consists of two steps, ontology and semantic type mapping, which map normalized text to UMLS concepts and semantic types. The documents are ranked based on their semantic and meaning similarity to the user-selected documents by the fuzzy & text mining relevancy representation and ranking unit. A key feature of the system is that the user is allowed select some of the found documents and use them as relevance feedback.

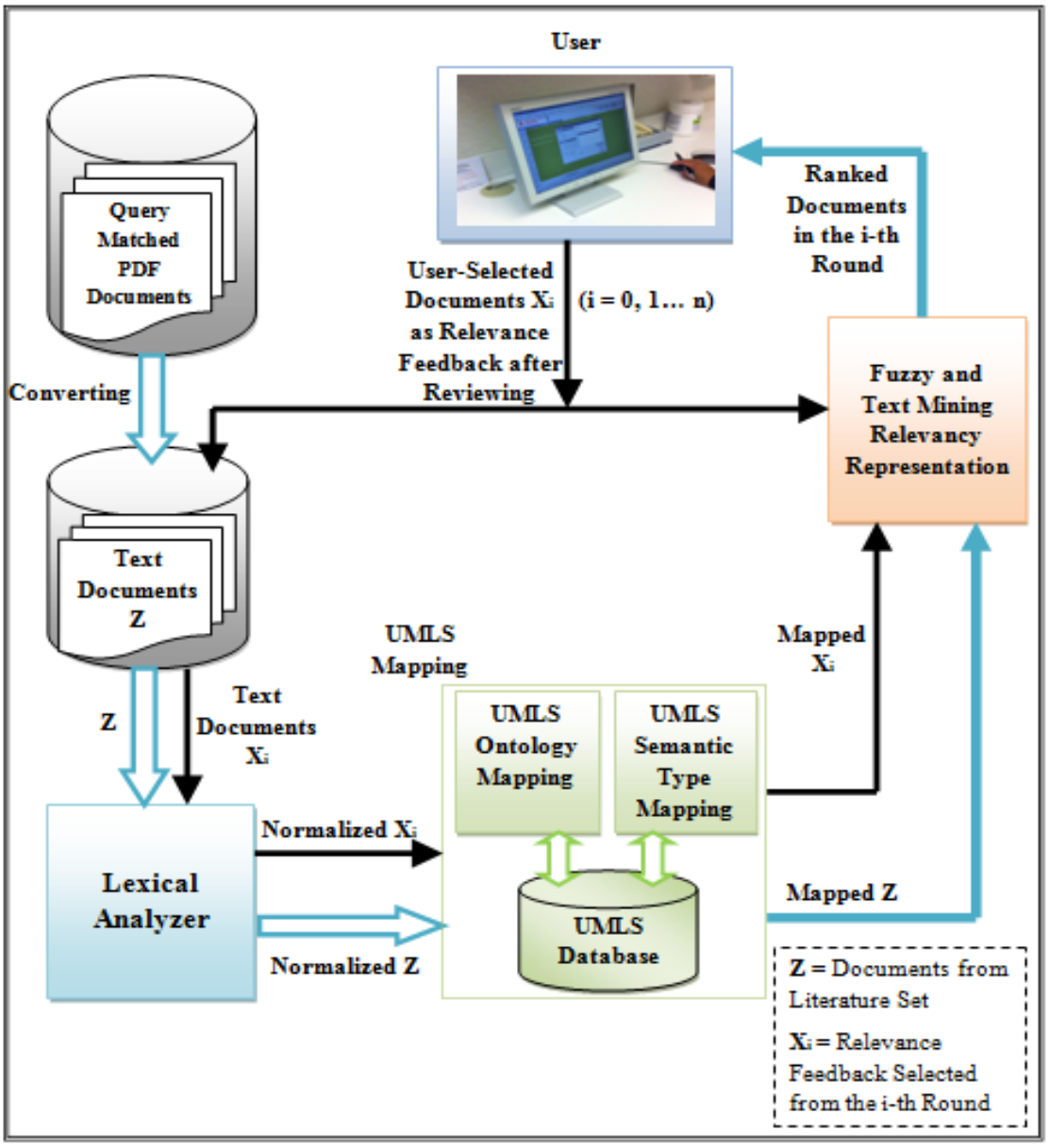

Figure 2.9: The architecture of the relevance feedback-based literature search

The user reviews and selects some documents X_i from the literature set which are used as relevance feedback. The user needs only to reveal whether or not the selected documents are relevant to his/her preference by simply indicating Yes or No. The relevance feedback with the user selected mapped documents are passed to the fuzzy & text mining relevancy representation and ranking unit. The fuzzy & text mining relevancy representation and ranking unit returns the new ranked set of documents to the user. The user reviews the new ranked set of documents. If the user does not find the information he/she is looking for, he/she selects other documents X_i as new relevance feedback documents and the process is repeated until the user find his/her needs as shown in figure 2.9.

2.3.1.2 Methodology

The following three steps are used to best rank the documents using relevance feedback. First, terms, which are words, in the articles are normalized using natural language processing techniques via lexical analyzer. Text normalization is a process by which the text in human language is analyzed and transformed in such way to make it more useful and consistent for further processing such as, removing unwanted terms, and converting a word to its base form. Then, the hidden information, which are specific pieces of information or facts in the text or relations between texts, is extracted or located for documents relevancy ranking purpose. This step is achieved by mapping the normalized terms in each article to the UMSL concepts and semantic types by UMLS mapping. Mapping is a process in which the normalized terms are linked to the best matching UMLS concepts and semantic types. Finally, calculate and measure the similarity between the

user-selected articles and the other articles in the literature set and rank the result based on the similarity of the documents in the literature set to the user-selected articles using fuzzy relevancy representation and ranking. The previously mentioned three steps are explained in more details as following:

A. Lexical Analyzer

The biomedical literature articles contain words in human language that are not important or useful for UMLS mapping. Hence, each article must be normalized, pre-processed, using the natural language processing techniques. To improve the mapping process and achieve better grasp of the content of the articles, the following seven pre-processing actions are used:

- Remove Genitive

  The genitive (‘), possessive marker, is the grammatical case that is often marks a noun as being the possessor of another noun. This sign makes the mapping process that is described later harder; hence, it must be removed before further processing.

- Remove Plural Patterns

  In English, the nouns are either single or plural. To achieve best mapping result, all normalized nouns must be in the single form. Therefore, the parenthetic plural forms of (s), (es), and (ies) should be stripped.

- Replace Punctuation with Spaces

  The English written text always contains some punctuations such as, @, (, {, [, -, and ! that indicate the structure and organization of the text. These punctuations

must be replaced with spaces to better map the words to their corresponding UMLS concepts.

- Remove Stop Word

  Stop words are the common and functional words such as the, an, a, and, or. They often do not contribute to the distinctive meaning and context of the text. Excluding the stop words can speed up the mapping process. I exclude the stop words and keep only the significant words and medical terms. I use stop word list obtained from Ranks NL – Webmaster Tools [69] that is used in their article analyzer for English text.

- Filter Proper Nouns

  A proper noun is the special name that is used for a person, place or organization. Filtering out the proper nouns can enhance and speed up the mapping process.

- Un-inflect Words

  The inflection is a modification or change in the form of a word, (typically the ending) to express different grammatical functions such as person, tense, and gender. Un-inflecting a word means changing the word to its base form. Using word un-inflection means there can be less word variation. This action can help minimizing the words variation, which makes the mapping easier. The idea is to improve the ability of detecting similarity by reducing the number of words that share a common meaning instead of mapping words as they appear in the documents.

- Canonicalize Words

  Canonicalization is the process of converting words that have more than one possible representation into their standard form. Therefore, there will be only one form for the same equivalent words. This action can speed up the mapping process.

The basic idea behind Un-inflect and Canonicalize of the words, terms, is to reduce the number of terms by: 1) Steaming the terms to their root for terms that share the same meaning, (e.g. 'educate' for education, educational, educating, etc.). 2) Converting the terms that have more than one representation to their standard form (e.g. 'teaching' for teaching, learning, tutoring, etc.). Therefore, there will be less number of terms; this process can speeds up the mapping process and makes it easier. Moreover, the ability to detect the similarity is improved regardless of the use of term variants and representations.

B. UMLS Mapping

After normalizing the words in each document, it is essential in the methodology to classify the words. The UMLS provides a mechanism for words classification using their meaning. It classifies the words that refer to the same concept by assigning them the same concept identifier (CUI). For example, "atrial fibrillation", "auricular fibrillation", and "A-Fib" all link to the same concept identifier "C0004238". This type of ontology mapping or classification is achieved by mapping each word to its concept identifier through exploring the UMLS Metathesaurus. Moreover, the UMLS provides semantic categorization to the concepts. The semantic type mapping or classification procedure is attained through exploring the UMLS semantic network. The UMLS semantic network is an upper level ontology that provides basic semantic type to each concept identifier. For instance, the

semantic type of “atrial fibrillation”, which has the concept identifier “C0004238” is “Finding” [49].

C. Fuzzy and Text Mining Relevancy Representation and Ranking

This unit consists of two processes, which are text mining relevancy ranking and fuzzy filtering ranking processes as shown in figure 2.10. Each of these processes is explained in deep details in the following section:

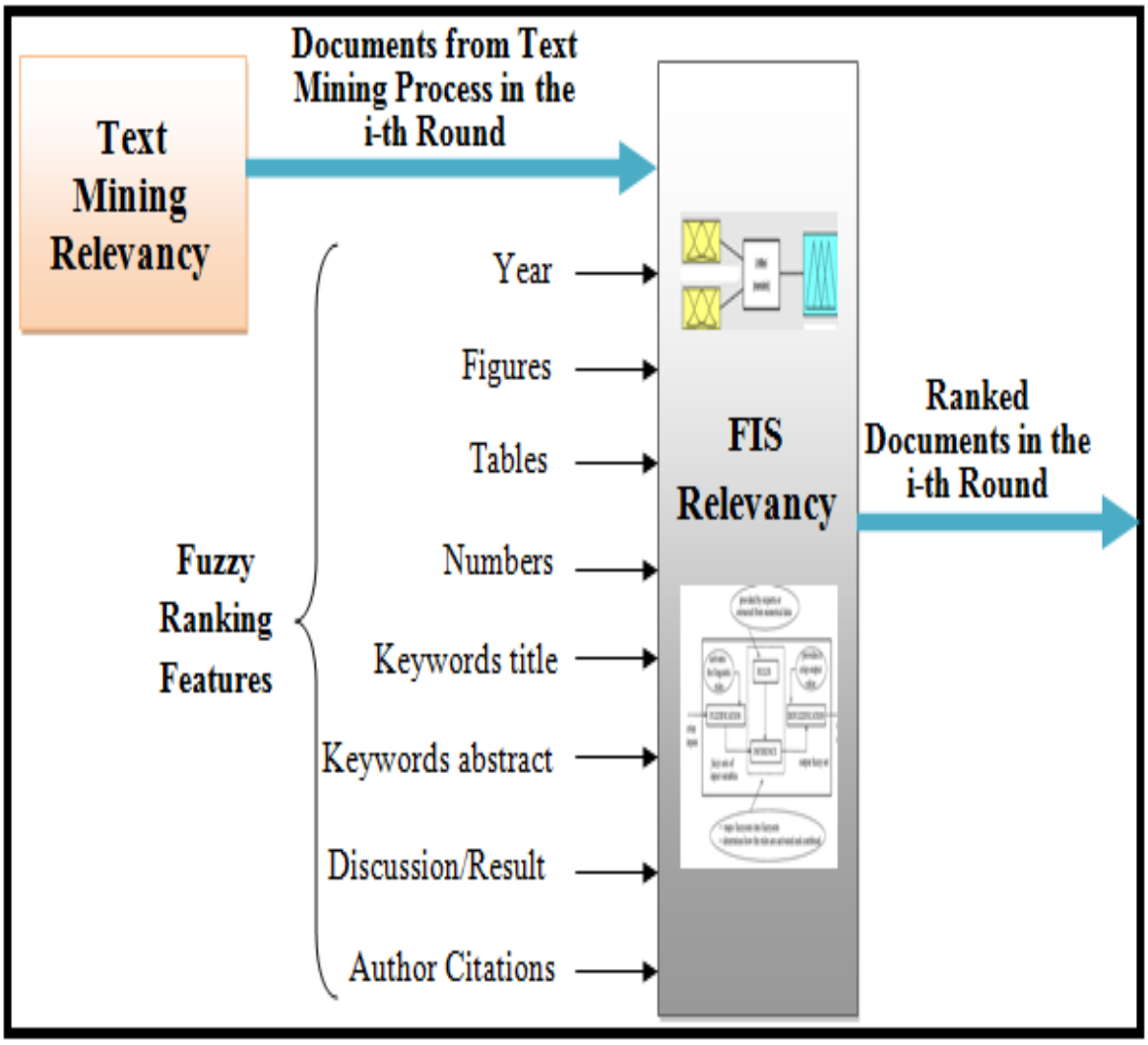

Figure 2.10: Fuzzy and Text Mining Relevancy Representation and Ranking

1. Text Mining Relevancy Ranking Process

The typical similarity measurement method used in the natural language processing applications for calculating document similarity is the vector space model in combination with cosine similarity [70, 71]. In text mining, to be able to measure the similarity between documents, it is required to represent the documents in mathematical textual data representation that describes sets of text documents. The vector space model is a typical algebraic representation of text documents used in natural language processing. In the vector space model, the vectors are constructed by representing each document as a vector containing the terms (words) weight. This can be accomplished through a commonly used weighting scheme that is Term Frequency-Inverse Document Frequency (TF-IDF) weighting scheme [72]. This weighting scheme is used to measure the importance of a term by the appearance or frequency of the term in the document multiplied by the inverse document frequency for that term. TF-IDF is an algorithm that has been widely used in information retrieval and text mining to evaluate the weight for each word in the collection documents. Term Frequency (TF) measures how many times a word appears in the document. Terms that appear many times in a document are most likely to be important within the document [72]. The TF of a term in a document is represented as:

$$tf_i = \frac{n_i}{\sum_k n_k} \tag{2.3}$$

where $n_i$represents the number of occurrences of the i-th term encountered in that document, n_k is the total number of all terms in the same document. The inverse document frequency (IDF) measures the general importance of a term in a collection. IDF of a term is calculated as:

$$idf_i = \log \frac{N}{df_i} \quad (2.4)$$

where N represents the total number of documents, and df_i is the number of documents that contain the term i. The higher value of IDF means rare term and the lower value means a common term. The TF-IDF weight of a term is achieved by multiplying TF and IDF as:

$$\text{TF-IDF} = \text{TF} * \text{IDF} = tf_i * \log \frac{N}{df_i} \quad (2.5)$$

So, each document in collection X is represented as a vector of TF-IDF weights:

$$\overrightarrow{x_i} = (w_{i,1}, w_{i,2}, w_{i,3}, \ldots \ldots w_{i,n-1}, w_{i,n}) \quad (2.6)$$

The vector space model for the whole set of documents is represented by the d x m dimensional matrix $||w_{ij}||$, where d is the number of words in all documents of the set, m is the number of documents in the set, w_{ij} is the TF-IDF weight of j-th word in the i-th document, and $\overrightarrow{x_i}$ represents a document's vector. The encoding of the documents into vectors is called indexing. During indexing, a global vocabulary is built up, assigning a unique identification to each word encountered in the entire collection. With this global vocabulary, a vector is constructed for each document with as many elements as the total number of words in the global vocabulary. For words appearing in the document at hand, the value of the respective elements is equal to its TF-IDF weight. For words not appearing in the document, the respective elements obtain a zero value [70, 72]. In my proposed system, the vector model applies for only significant words and medical terms, so stop-words are excluded.

Using vector space model, the similarity between two documents (vectors) can be computed by calculating the cosine angle between the two vectors (cosine measure

method). Specifically speaking, the cosine measure is used to quantitatively estimate the relevance of the given two documents [71]. Therefore, to compute the similarity between two documents (vectors), $x_i$and x_j, one can use the cosine similarity measure that is equal to the dot product of the vectors normalized by the product of the vector lengths. It calculates the vectors inner product as:

$$\cos\theta = S(x_i, x_j) = \frac{x_i \cdot x_j}{\|x_i\| \, \|x_j\|} \qquad (2.7)$$

where θ is the angle between the two vectors x_i and x_j. Since all vector elements are positive, a word can appear zero or more times in a document, the similarity results are values between one and zero. The similarity between two copies of the same document is one; if the two documents have no words in common, the similarity is zero [70, 71].

In this step of the search process, I explore the use of terms meaning and semantic types as two types of features to help for the search process and relevancy calculation. Therefore, each document is represented by two vectors, which are term meaning and semantic type. The text-mining relevancy step calculates the cosine similarity between two vectors of the same type for each document from the collection set against the user selected document(s) as in equation (2.7). In other words, the cosine similarity is calculated two times between the user-selected document(s) and each document from the literature set; the first between the terms meaning vectors and the second between the semantic type vectors. Since each document is represented by two vectors, it is essential to find a mechanism that facilitates the relevancy representation process. Consequently, I build this process by generating only one cosine similarity result for each document from the literature set against the user-selected document(s). I calculate the new cosine similarity by giving different

weight to each cosine similarity result, term meaning and semantic type, as shown in equation (2.8).

$$S(x_i, x_j)_{new} = (S(x_i, x_j)_m \text{ x } w_m) + (S(x_i, x_j)_{st} \text{ x } w_{st}) \quad (2.8)$$

where $S(x_i, x_j)_{new}$ is the new cosine similarity between the user-selected document(s) and a document from the literature set, $S(x_i, x_j)_m$ is the term meaning cosine similarity result between the user-selected document(s) and the document from the literature, w_m represents the weight for the term meaning cosine similarity, $S(x_i, x_j)_{st}$ is the semantic type cosine similarity result between the user-selected document(s) and the same document from the literature, and w_{st} represents the weight for the semantic type cosine similarity. Since the user selects one or more document(s) as relevance feedback, they can be relevant, or non-relevant.

To calculate the cosine similarity, the user-selected document(s) as relevance feedback must be represented by two vectors: term meaning and semantic type. For this purpose, the relevance feedback documents are represented as two vectors as the following:

- If the relevance feedback documents are relevant, one vector is constructed from the user-selected document(s) that contains the TF-IDF weight of all terms from all documents for each type. Therefore, two vectors are built from these documents, terms meaning and semantic type vectors, as relevant user feedback.
- If they are non-relevant, one vector is build from all of the user-selected document(s) that contains the TF-IDF weight of all terms from all documents for each type. Therefore, two vectors are constructed from these documents, terms meaning and semantic type vectors, as irrelevant user feedback.

2. Fuzzy Relevancy Ranking Process

The second process of the fuzzy and text mining relevancy representation is fuzzy relevancy and filtering; it extracts and decodes features embedded in the documents. One of the important roles of the fuzzy relevancy and filtering process is to decode explicit and implicit features, which are used for filtering and sifting unwanted articles and re-ranking the documents. Articles contain several features that I use to differentiate between them, and they have different levels of importance. Hence, I calculate weights for these features. Some of the features do not exist in the existing search engines. The features are: (1) medical cues and outcomes expected to be in the Results/Discussion Section of an article, (2) keywords in title, (3) keywords in abstract, (4) finding if the article contains a table and/or a figure, (5) check if the article contains numbers, (6) article's publication year, and (7) the article's author number of citations. At the end, the articles are ranked according to their relevancy to the user search criteria.

The ranking algorithm of the fuzzy relevancy process is designed in such a way that the search and ranking are performed according to features exist in the articles that indicates how each article is relevant to the user's search preference and the degree of the relevancy. The following features are extracted from articles and used to allow the user to narrow and re-rank his/her search result:

- Search the desired information in the Results/Discussion Section of an article. For a well-organized article, the Introduction Section should provide background information about the main topic, whereas the Method Section should mainly focus on the methodology of the study. The Results Section and the optional Discussion

Section should more likely contain the information that the healthcare provider is looking for - clinical findings of the study [8, 13].

- Determine whether the article contains a table or a figure. Tables and figures are features that can be used to distinguish between documents. Their existence can provide some hints concerning whether certain useful information that the user seeks exists in the article.
- Counting the author's number of citations. It is very useful to determine the number of citations that the author has published and cited. The authors who have many publications and citations in the same field are likely to have more knowledge in this field; hence, they are expected to be experts and their publications contain useful information and finding. Therefore, this can be one useful feature to distinguish between the documents in the literature search and ranking.
- Determine if the article contains numbers. Numbers is a significant feature that can provide the user with existence of important biomedical information and findings he/she is looking for.
- Check if the keyword(s) exist in the title and/or abstract. The present of the keywords in the title and/or abstract of the article, indicate that this document contains essential information and finding that the user may looking for. Thus, they can be useful features that used to filter the user's search and provide more accurate relevancy ranking result.
- Publication year; the user may be interested in finding information that published in a certain period of time. In addition, the user may prefer to find information in

recently published documents rather than old published ones. Therefore, publication year can be one feature used to distinguish between the articles.

The fuzzy relevancy step determines the degree of the articles' relevancy by calculating a cumulative weight for each article. The higher the cumulative weight is, the more relevant the article to the user preference is. The cumulative weight of a document is calculated as:

$$CW_j = \sum_{i=1}^{n} W_i * Fa_i \qquad (2.9)$$

Here CW refers to j^{th}document cumulative weight, n is the number of features, W_i represents the fuzzy weight of i^{th} feature, and Fa_i is the i^{th} feature occurrence factor. There are two types of features, which are Boolean feature and non-Boolean feature. The feature occurrence factor for Boolean feature is $\in \{0,1\}$ (it is one if the feature appears in the document and zero otherwise). While the feature occurrence factor for non-Boolean feature is computed based on its frequency range (Ff_r). The feature frequency range in a document is calculated as in (2.10).

$$Ff_r = \begin{cases} 0, & if \quad n = 0 \\ 0.25, & if \quad 0 < n \leq x_1 \\ 0.50, & if \quad x_1 < n \leq x_2 \\ 0.75, & if \quad x_2 < n \leq x_3 \\ 1.0, & if \quad x_3 < n \leq x_4 \end{cases} \qquad (2.10)$$

Where, n represents the frequency, present, of the feature in the document, and $x_1, x_2, x_3, and\ x_4$ are constants for the feature boundary rang.

The fuzzy and text-mining relevancy representation unit measures and calculates the relevancy of each article from the literature set to the user selected article(s) as relevance feedback.

2.3.2 Implementation

Since the UMLS and PubMed provide a variety of tools that are implemented using JAVA, the relevance feedback-based literature search component was built using JAVA programming language and it was tested under Linux and Windows 7 operating systems. The Fuzzy Inference system (FIS) of this component was implemented using MATLAB. First, the documents in PDF format are converted to text format for fast text search. The new text files are then normalized via the lexical analyzer unit. The lexical analyzer unit follows several natural language processing actions to filter and normalize the raw text. This unit is built with the help of Lexical and Text Tool provided by the UMLS [49]. Specifically, the Lexical Variant Generator (LVG) is used. Lexical Variant Generator is a JAVA tool that is designed to manage lexical variations, normalize, and index the raw text files. The normalization techniques provided by Lexical Variant Generator help to map terms, words, to their concepts identifier in the UMLS Metathesaurus. In this unit, the Lexical Variant Generator is used to 1) Remove Genitive; 2) Remove Plural Patterns; 3) Replace Punctuations with White Spaces; 4) Exclude Stop Words; 5) Filter out Proper Nouns; 6) Un-inflect Terms; 7) Terms Canonicalization; respectively.

The UMLS mapping unit is constructed with the help of the MetaMap application [73] to map the normalized terms to their corresponding concepts in the UMLS Metathesaurus. The MetaMap is a program providing access from normalized biomedical

literature to the concepts in the UMLS Metathesaurus. Since MetaMap assigns concept identifiers to all terms in running text, I believe that it can serve a very useful role as a generator of features, meaning and semantic type, for this application. The UMLS Mapping unit takes the normalized text and maps every term in each document to its corresponding concept identifier and semantic type. The ontology-mapping step constructs a vector for each document. This vector contains the normalized terms' concept identifier called "CUI" vector as in equation (4). The semantic type mapping step further builds another vector that holds the semantic type of the normalized terms for every document named "Semantic-Type" vector as in equation (4). Therefore, every article is represented by two vectors CUI and Semantic-Type vectors.

The text mining and fuzzy relevancy representation and ranking unit masures the articles' relevancy and rankes them based on their relevancy to the user relevance feedback, selected article(s). The text mining relevancy step uses the vectors obtained from UMLS mapping unit and calculates the cosine similarity between two vectors of the same type for each document from the literature set against the user selected document(s), which means there are two cosine similarity results. Since there are two cosine similarity results for each document compared against the user selected document(s), one cosine similarity result is constructed by applying equation (2.8) and giving different weight to each cosine similarity. Since meaning feature provides low-level classification and mapping process, it is given a weight of 0.6 and semantic type feature is given a weight of 0.4 because it is high-level medical categorization; therefore, it provides an upper level ontology mapping as have been showen in the previous wrok [52].

The fuzzy relevancy ranking process uses the result generated from previous procss to re-rank and generate new ranking result based on the articles' features. Specificly speaking, fuzzy relevancy ranking step takes the first 50 ranked articles from text mining relevancy step and ranks them using fuzzy logic and the article' features. The fuzzy relevancy ranking step is constructed with the help of fuzzy logic and sets. In this step, the following features are used to allow the user to rank the search result: 1) publication year; 2) keywords in title; 3) keywords in abstract; 4) keywords in discussion/result section; 5) tables in article; 6) figures in article; 7) numbers in the article; 8) article's author number of citations. Since the features have different level of importance, each of them is given a fuzzy weight. Therefore, a fuzzy inferance system (FIS) is constructed to rank the relevancy of the search result.

To build the FIS, five membership functions are used to represent the features weight: Very Low, Low, Medium, High, and Very High as illustrated in figure 2.11. Additionally, five fuzzy rules are used in the FIS for calculating the feature weights as shown in figure 2.12. Min-Max is used for fuzzy inference, and the centroid defuzzifier. The fuzzy relenancy ranking step determines the degree of relevancy for the articles using equation (2.9). The FIS is used to assign fuzzy weight to each feature. The feature occurance factor for Boolean features (keywords in title, keywords in abstract, keywords in disscution/result, figure, table, and year) is zero if the feature does not appear in the document and one if the feature present in the document. While it is calcualted using equation (2.10) for the non-Boolean features (numbers, author citations). The four

constants (x_1, x_2, x_3, and x_4) in this eqattion represent the boundry rang of the feature appears.

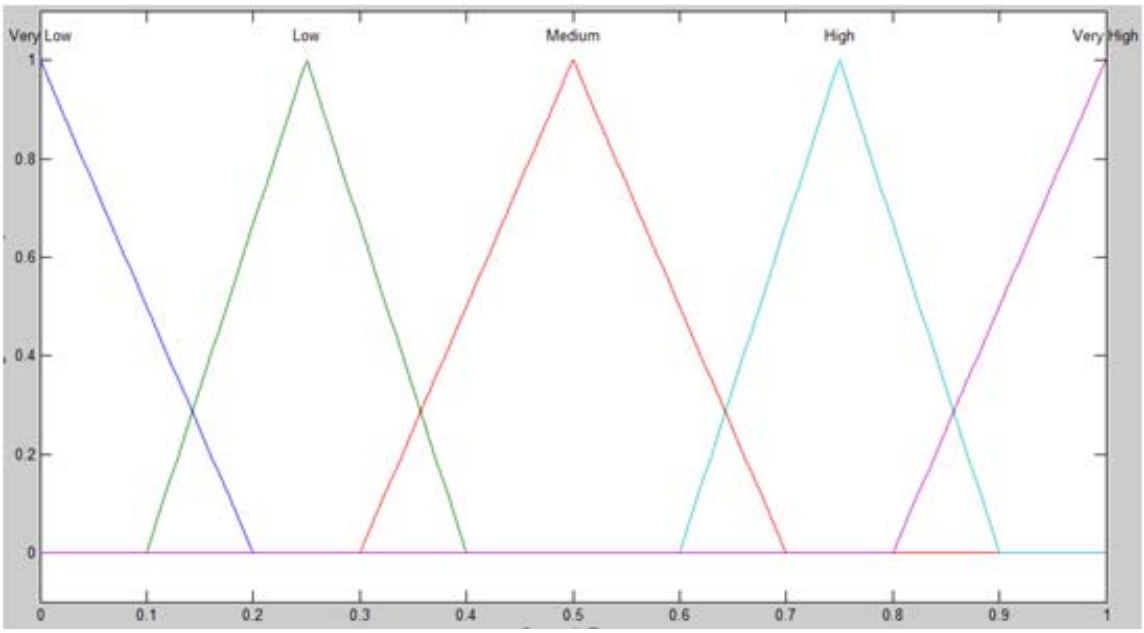

Figure 2.11: Five fuzzy sets representing the features weight

- If the feature fuzzy weight is ***Very High*** then the feature important level is ***Very High***
- If the feature fuzzy weight is ***High*** then the feature important level is ***High***
- If the feature fuzzy weight is ***Medium*** then the feature important level is ***Moderate***
- If the feature fuzzy weight is ***Low*** then the feature important level is ***Low***
- If the feature fuzzy weight is ***Very Low*** then the feature important level is ***Very Low***

Figure 2.12: Five fuzzy rules of the fuzzy inferance system

2.4 Summary

This chapter introduces the overall design and architecture of a novel relevance feedback-based biomedical literature search system for quickly narrowing biomedical literature search using UMLS knowledge sources, text mining techniques, and fuzzy sets and logic. The system extracts meaning and semantic relations from documents and uses them in the search and ranking process. Moreover, this chapter describes the design and implementation of the main system's components. Furthermore, it explains in deep details the methodology of the search process and relevancy ranking. This chapter also describes how relevance feedback is applied in the search process. It also explains how fuzzy logic and sets, UMLS, and text mining techniques are utilized and used in the search and ranking process.

CHAPTER 3

EVALUATION OF THE RELEVANCE FEEDBACK-BASED BIOMEDICAL LITERATURE SEARCH SYSTEM

This chapter provides the experiment design and setting of the implemented relevance feedback-based biomedical literature search system and its main components. Moreover, it presents and discusses the obtained results where it shows the effectiveness and usefulness of using UMLS semantic type and meaning relations between texts in the search strategy under relevance feedback mechanism.

3.1 Experiment Design and Results for Evaluating Fuzzy-Based PDF Features Extraction (Main Component 1)

The main goal of this experiment is to show the overall system design and the preliminary results involving the fuzzy part of the system. To this end, Dobutamine (a drug for treating heart failure and cardiogenic shock) is used as keyword query. The query was sent to PubMed database where the publication time period was restricted to the last 20 years; it retrieved 2,184 article citations as a result. The citations were imported to EndNote, which uses the citations information to automatically download 1,153 PDF full-texts from several online databases and repositories that either are available free or Wayne State University is subscribed to (the other 1,031 articles could not be obtained because Wayne State University library is not subscribe to the related journals).

The system is tested in terms of the search criteria, and it worked properly. For example, when the English language parameter was set, the program correctly found 1099

articles in English. Therefore, the used biomedical data set contains 1,099 original biomedical documents. The rest of the search parameters (e.g., publication year, figure, table and Result Section) all performed correctly either individually or jointly using AND and OR operators. The system is also tested using the following keywords: heart, failure, survival, cardiogenic, and shock. Two physicians on the team are asked to assign fuzzy weights to the keywords. The assigned fuzzy weight was VL, L, S, S, and VS respectively. The result shows that there are 286 articles satisfying the keyword criteria.

Moreover, the implemented system is preliminarily tested using AND operator. Fuzzy weights are given to the features as shown in Table 3.1. The fuzzy relevancy ranking system returned the following results: 245 articles with very low relevance, 188 with low relevance, 68 with moderate relevance, 27 with high relevance and no articles with very high relevance. Figure 3.1 shows sample result of the fuzzy logic-based PDF features extraction component of relevance feedback-based biomedical literature search system.

Table 3.1: Features' fuzzy weight

Feature	Fuzzy Weight
Keywords in Abstract	Very Large
Fuzzy Keywords Weight	Large
Keywords in Title	Very Large
Result Section using AND	Extremely Large
Result Section using OR	Moderate
Table	Very Moderate
Figure	Small
Publication Year	Very Small

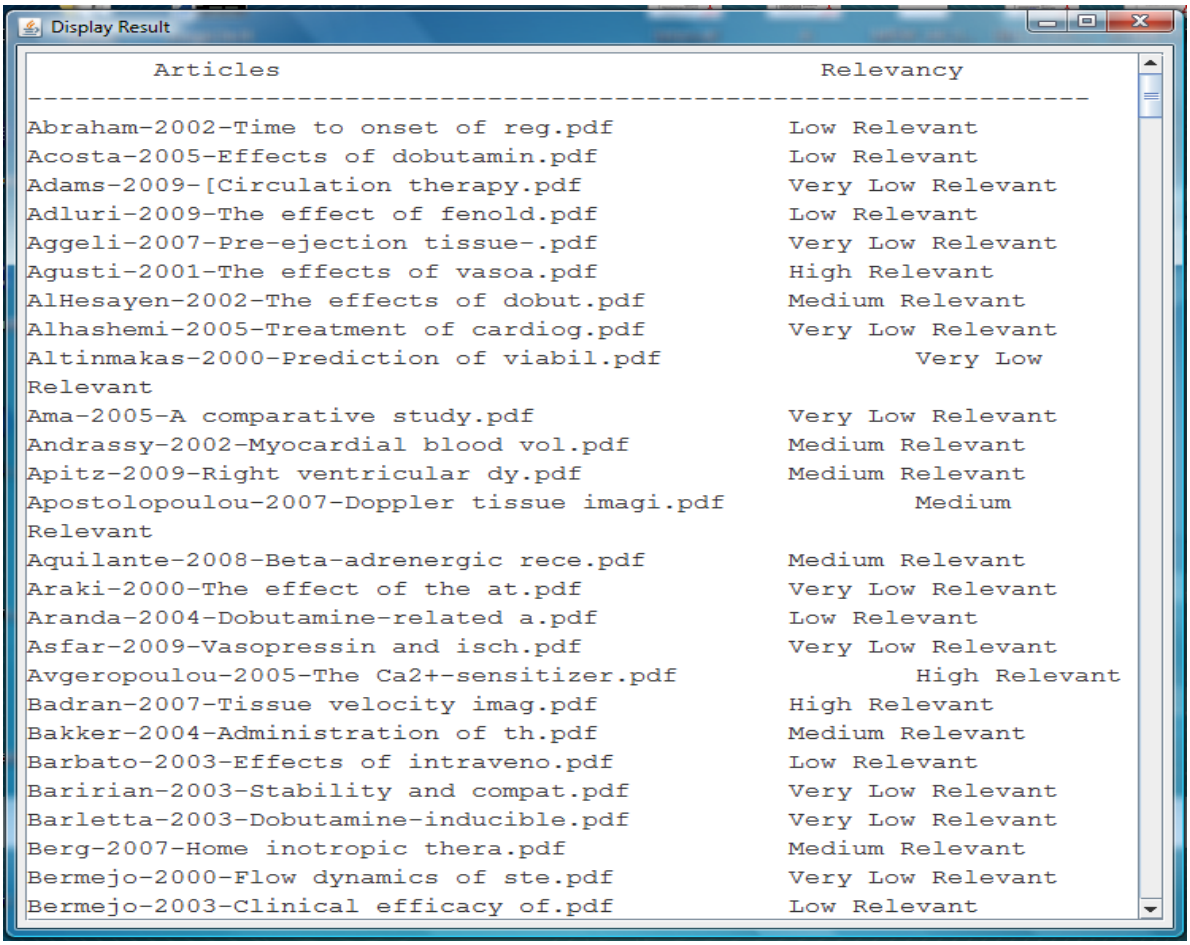

Figure 3.1: Sample result of the first component of system

3.2 Experiment Design and Results for Evaluating the Main Component 2 and the Entire System

This section describes the experiment design and shows the evaluation results for main component 2 and the overall relevance feedback-based biomedical search system. It shows the benefit of calculating the similarity between the documents using UMLS semantic type and meaning in the search and ranking process when using relevance feedback. It also shows the effectiveness of using fuzzy logic and sets and UMLS in the biomedical literature search process.

3.2.1 Experiment Design

The purpose of these experiments is to explore how to measure the relevancy, rank the search result using the fuzzy methodology, text mining technique, UMLS, and relevance feedback mechanism, and show the evaluation result; moreover, the main objective is to improve the efficiency of the relevancy of the retrieved search result. To achieve precise and coherent evaluation result, more specific keywords (30-day mortality) are used rather than the general keyword (Dobutamine). Conducting a search for the above-mentioned specific keywords results in the fact that these specific keywords exist in 327 documents out of the available data set of 1,099 documents. This result, 327 documents, is still large for a user to go through and review manually.

The difficult part to evaluate retrieval and search system is generating a ground truth. Since establishing ground truth for the specific query, 30-day mortality, among the 327 documents without human interaction is impossible, I decided to use all available data set for evaluation. While I do not have data on the number of relevant documents for the used specific query with respect to the whole data set, I am not able to compute neither the recall nor the F-measure performance evaluation metrics. This is because of the fact that computing any of them is based on knowing the number of relevant documents in the whole available data set. Hence, I only compute the precision as the main performance evaluation metric to measure the accuracy of the query result and manually decide the relevant documents of each search result round. The precision is the fraction of retrieved documents that are related to the used search query and is computed as shown in equation (3.1) [72].

For this purpose, two physicians are involved and manually evaluate the relevancy of each document in the search result for the used specific query.

$$Precision = \frac{|\{\text{relevant documents}\} \cap \{\text{retrieved documents}\}|}{|\{\text{retrieved documents}\}|} \quad (3.1)$$

To evaluate the proposed system, I designed the following experiment. In this experiment, I evaluated the performance of using relevance feedback mechanism in the search process. I used the specific keywords (30-day mortality) as the search query. Moreover, I generated two independent result sets for the query where each result set is evaluated by one physician. Each independent result set contains ten documents. Furthermore, to show the usefulness of using relevance feedback mechanism in the literature search, I use three rounds for each independent result set that is evaluated by one physician. Each round contains ten documents. Each document in each result set is manually evaluated as either relevant to the specific query or irrelevant. Finally, the precision is computed for each round of each independent result set.

To find the agreement level, correlation, between the two physicians, I designed an experiment that measures the agreement level between them. For this purpose, the two physicians are asked to evaluate whether a number of given randomly selected documents are relevant to the used specific query or not. Then, based on the evaluation result I calculate the agreement level. The physicians are given randomly selected 53 documents and asked to evaluate them based on their relevancy to the used specific query (30-day mortality) as YES (relevant) or NO (irrelevant).

3.2.2 Results

In this section, I report and discuss the result of the agreement level between the two physicians. I also evaluate and analyze the performance of the system and show the precision performance of the relevance feedback-based biomedical literature search system.

Table 3.2 shows the agreement level between the two physicians. As we can see from the table that the two physicians agree that 15 documents are relevant to the query and 27 are irrelevant to the query. While physician-1 evaluated three documents as relevant, physician-2 evaluated them as irrelevant. Physician-2 evaluated eight documents as relevant, but physician-1 evaluated them as irrelevant. Hence, the two physicians' verdict is similar in evaluating 42 out of 53 documents. Therefore, the agreement level between them is 79.25%. This agreement result reflects a good agreement level between the two evaluators, which leads to a meaning evaluation result comparison.

Table 3.2: The agreement level between the two physicians

Phyisician-2		Phyisician-1	
		YES	NO
	YES	15	8
	NO	3	27

Each feature that is decoded or extracyed from the articles is given a fuzzy weight based on its level of importance. Tables 3.3 and 3.4 show the features fuzzy weight per round per phyisician. Table 3.3 represents the fuzzy weight for the article features for the

results that are evaluated by phyicisian-1. Table 3.4 illustrates the fuzzy weight for the features decoded form documents for results that are judged by phyicision-2.

Table 3.3: Fuzzy features weight for result evaluated by physician-1

	Fuzzy Weight Round-1	Fuzzy Weight Round-2	Fuzzy Weight Round-3
Publication Year	High	Very High	High
Keywords/Title	Medium	High	High
Keywords/Abstract	Medium	High	Very High
Keywords/Discussion	Very High	Very High	Very High
Tables	Medium	Very High	Medium
Figures	Medium	High	High
Numbers	Low	Medium	Medium
Author Citations	High	High	High

Table 3.4 Fuzzy features weight for results evaluated by physician-2

	Fuzzy Weight Round-1	Fuzzy Weight Round-2	Fuzzy Weight Round-3
Publication Year	High	Very High	High
Keywords/Title	Medium	High	High
Keywords/Abstract	Medium	High	High
Keywords/Discussion	Very High	Very High	Very High
Tables	Medium	Very High	Medium
Figures	Medium	High	High
Numbers	Low	Medium	Medium
Author Citations	High	High	Very High

As meantioned early, I used the whole data set, 1099 documents, for the system performance evaluation and measuring the precision for each round. Figure 3.2 depicts the

precision evaluation that measures the performance of the proposed biomedical search system.

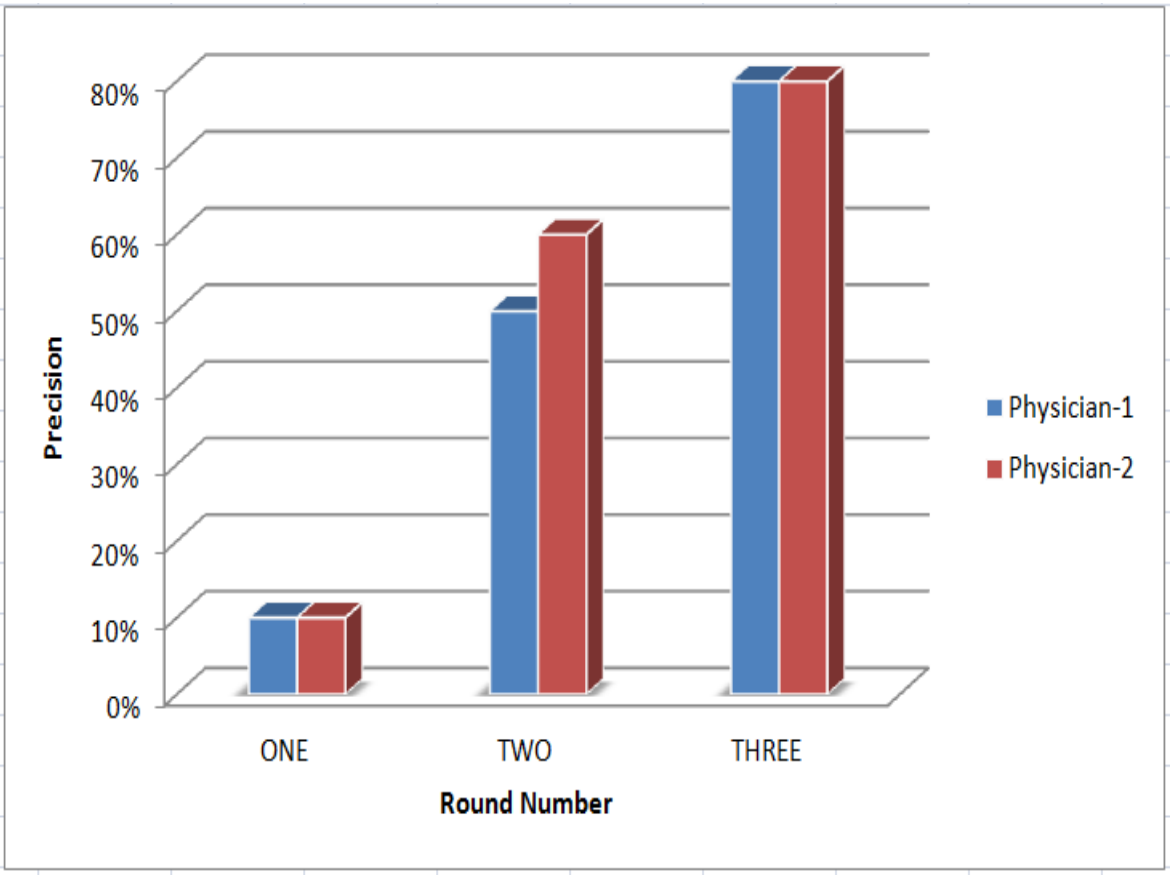

Figure 3.2: The precision result of the rounds per the physicions

It is clear from the graph that the search result improves as the rounds proceed. Figure 3.2 shows that the precision improves by 87.5% in three rounds according to results evaluated by physician-1. From figure 3.2, we can see that the results which is evaluated by phyisician-1 improves as the rounds proceed, where the precision of round one is 10%, round two is 50%, and round three is 80%. This shows the effectiveness of using relevance feedback mechanism associated with UMLS and fuzzy logic in the search process. That means the number of relevant retrieved documents increases from round one to round two,

and from round two to round three. The improvement of precision from round one to round two is 80%, and it achieves an improvement of 37.5% from round two to round three. In general the precision gains an improvement of 87.5% from round one to round three. This is a significant improvement in three round for the retrieved results.

Moreover, the graph shows that the number of retrieved related documents in results which is evaluated by phyisician-2 increases as the rounds proceeds. The precision increases from 10% in round one to 60% in round two, which is an improvement of 83.33%. The precision also increases to 80% in round-3, which means an improvement of 25%. The retrieved result improves by 87.5% based on results evaluated by phyisiacian-2 in three rounds.

In general, the result shows the effectiveness and usefulness of using relevance feedback mechanism associated with UMLS, and fuzzy logic in the search process and result ranking as evidenced by the shown experiments and results. Furthermore, the results demonstrate the efficiency and worthiness of using Unified Medical Language System knowledge sources and text mining techniques in biomedical literature search result.

3.3 Summary

This chapter reports the evaluation part of the relevance feedback-based biomedical literature search system. Using a real-world biomedical data set, I showed a result on usefulness of the fuzzy logic in extracting PDF features. The relevance feedback-based biomedical literature search system has been evaluated using a real-world biomedical data set that contains 1,099 original documents. This chapter reports the effectiveness of using relevance feedback methodology associated with UMLS, fuzzy logic, and text mining in the

search to match similar and related biomedical documents. This chapter presents several experimental results that demonstrate the usefulness of the developed system in the biomedical literature search using a real biomedical data set.

CHAPTER 4

A RANKING METHOD BASED ON FUZZY LOGIC AND UNIFIED MEDICAL LANGUAGE SYSTEM

One way to improve the retrieval accuracy is via the relevance feedback technique [21, 48]. Therefore, this chapter introduces a new fuzzy logic-based ranking mechanism involving UMLS semantic and meaning features to rank retrieved search result based on relevance feedback. This chapter reports how the similarity between the documents is calculated and how the relevant documents are ranked with the help of fuzzy logic in an experiment and showed the results [52]. Next section shows the development of the fuzzy logic-based ranking method.

4.1 Development of the Fuzzy Logic-Based Ranking Method

Figure 4.1 depicts the design of the fuzzy logic-based ranking method. It shows how the components interact and connecte with each other. This design consists of three main units: lexical analyzer, UMLS mapping, and fuzzy relevancy representation and ranking units. In addition, it has several databases containing UMLS and literature document set. The UMLS database contains the UMLS knowledge sources. The lexical analyzer unit provides several natural language processing techniges for text normalization. The UMLS mapping unit consists of two steps, ontology and semantic type mapping, which mapp normalized text to UMLS concepts and semantic types. The documents are ranked based on their semantic and meaning similarity to the user selected documents by the fuzzy relevancy representation and ranking unit.

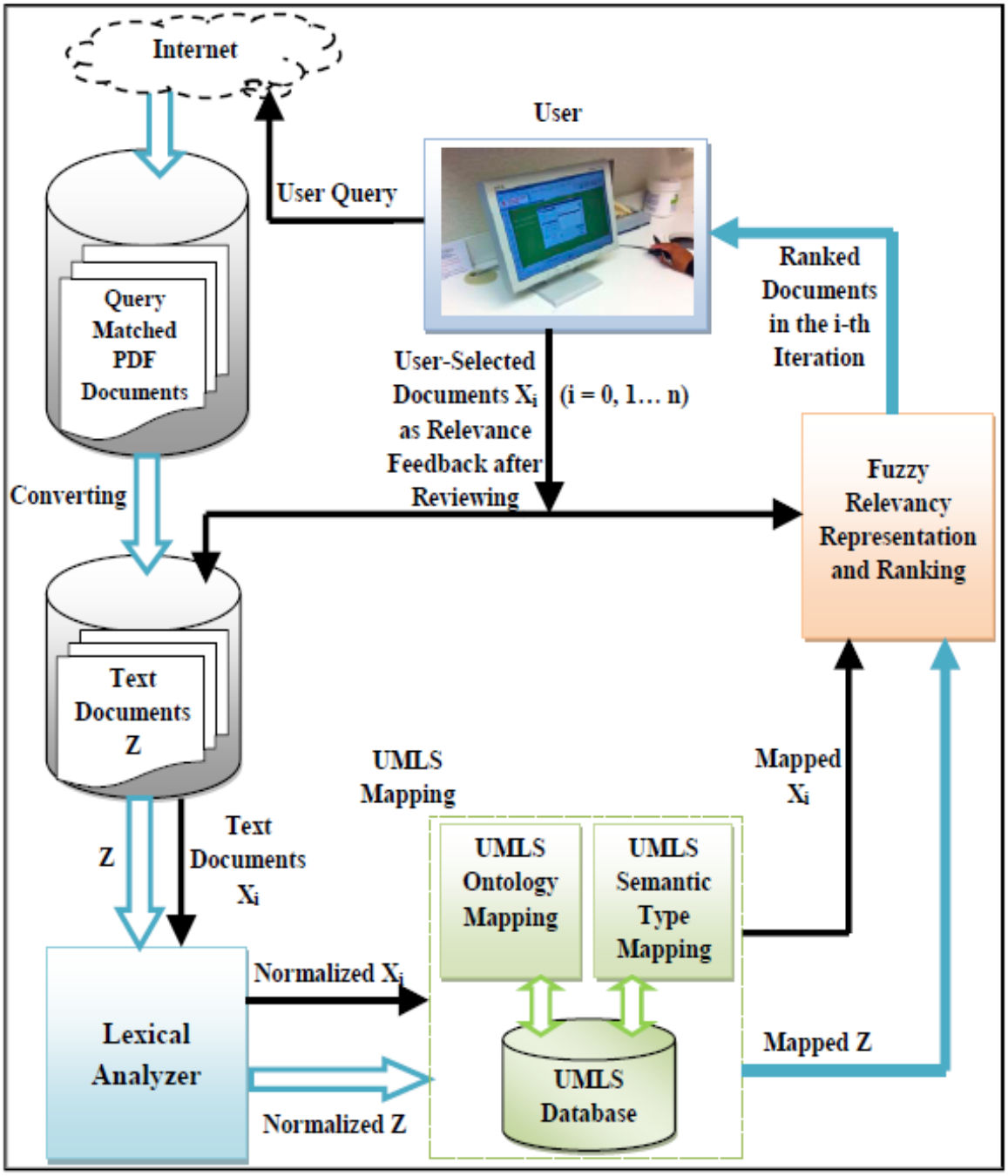

Figure 4.1: Design of the fuzzy logic-based ranking method

The user reviews and selects some documents X_i from the literature set which are used as relevance feedback. The user needs only to reveal whether or not the selected documents are relevant to his/her preference by simply indicating Yes or No. The

relevance feedback with the user selected mapped documents are passed to the fuzzy relevancy representation and ranking unit. The fuzzy relevancy representation and ranking unit returns the new ranked set of documents to the user. The user reviews the new ranked set of documents. If the user does not find the information he/she is looking for, he/she selects other documents X_i as new relevance feedback documents and the process is repeated until the user find his/her needs as shown in figure 4.1. In this work, the user is limited to select only one document, which is used as relevance feedback.

4.2 The Literature Search Result Ranking Process

The following three steps are used to best rank the documents using relevance feedback. First, terms, which are words, in the articles are normalized using natural language processing techniques via lexical analyzer. Text normalization is a process by which the text in human language is analyzed and transformed in such way to make it more useful and consistent for further processing such as, removing unwanted terms, and converting a word to its base form. Then, the hidden information, which are specific pieces of information or facts in the text or relations between texts, is extracted or located for documents relevancy ranking purpose. This step is achieved by mapping the normalized terms in each article to the UMSL concepts and semantic types by UMLS mapping. Mapping is a process in which the normalized terms are linked to the best matching UMLS concepts and semantic types. Finally, calculate and measure the similarity between the user-selected articles and the other articles in the literature and rank the result based on the similarity of the documents in the literature to the user-selected articles using fuzzy

relevancy representation and ranking. The previously mentioned three steps are explained in more details as following:

4.2.1 Lexical Analyzer

The biomedical literature articles contain words in human language that are not important or useful for UMLS mapping. Hence, each article must be normalized, pre-processed, using the natural language processing techniques. To improve the mapping process and achieve better grasp of the content of the articles, the following seven pre-processing actions are used:

- Remove Genitive

  The genitive (‘), possessive marker, is the grammatical case that is often marks a noun as being the possessor of another noun. This sign makes the mapping process that is described later harder; hence, it must be removed before further processing.

- Remove Plural Patterns

  In English, the nouns are either single or plural. To achieve best mapping result, all normalized nouns must be in the single form. Therefore, the parenthetic plural forms of (s), (es), and (ies) should be stripped.

- Replace Punctuation with Spaces

  The English written text always contains some punctuations such as, @, (, {, [, -, and ! that indicate the structure and organization of the text. These punctuations must be replaced with spaces to better map the words to their corresponding UMLS concepts.

- Remove Stop Word

  Stop words are the common and functional words such as the, an, a, and, or. They often do not contribute to the distinctive meaning and context of the text. Excluding the stop words can speed up the mapping process. The stop words are excluded and only the significant words and medical terms are kept. A stop word list obtained from Ranks NL – Webmaster Tools [69], which is used in their article analyzer for English text, is used in this step.

- Filter Proper Nouns

  A proper noun is the special name that is used for a person, place or organization. Filtering out the proper nouns can enhance and speed up the mapping process.

- Un-inflect Words

  The inflection is a modification or change in the form of a word, (typically the ending) to express different grammatical functions such as person, tense, and gender. Un-inflecting a word means changing the word to its base form. Using word un-inflection means there can be less word variation. This action can help minimizing the words variation, which makes the mapping easier. The idea is to improve the ability of detecting similarity by reducing the number of words that share a common meaning instead of mapping words as they appear in the documents.

- Canonicalize Words

  Canonicalization is the process of converting words that have more than one possible representation into their standard form. Therefore, there will be only one form for the same equivalent words. This action can speed up the mapping process.

4.2.2 UMLS Mapping

After normalizing the words in each document, it is essential in my methodology to classify the words. The UMLS provides a mechanism for words classification using their meaning. It classifies the words that refer to the same concept by assigning them the same concept identifier (CUI). For example, “atrial fibrillation”, “auricular fibrillation”, and “A-Fib” all link to the same concept identifier “C0004238”. This type of ontology mapping or classification is achieved by mapping each word to its concept identifier through exploring the UMLS Metathesaurus. Moreover, the UMLS provides semantic categorization to the concepts. The semantic type mapping or classification procedure is attained through exploring the UMLS semantic network. The UMLS semantic network is an upper level ontology that provides basic semantic type to each concept identifier. For instance, the semantic type of “atrial fibrillation”, which has the concept identifier “C0004238” is “Finding” [49].

4.2.3 Fuzzy Relevancy Representation and Ranking

The typical similarity measurement method used in the natural language processing applications for calculating document similarity is the vector space model in combination with cosine similarity [70]. In text mining, to be able to measure the similarity between documents, it is required to represent the documents in mathematical textual data

representation that describes sets of text documents. The vector space model is a typical algebraic representation of text documents used in natural language processing. In the vector space model, the vectors are constructed by representing each document as a vector containing the number of occurrences terms encountered in that document. So, each document in collection X is represented as a vector of term weights:

$$\overrightarrow{x_i} = (w_{i,1}, w_{i,2}, w_{i,3}, \ldots\ldots w_{i,n-1}, w_{i,n}) \qquad (4.1)$$

The vector space model for the whole set of documents is represented by the *d* x *m* dimensional matrix $\|w_{ij}\|$, where *d* is the number of significant words, excluding stop words, in all documents of the set, *m* is the number of documents in the set, w_{ij} is the weight of *i*-th word in the *j*-th document, and $\overrightarrow{x_i}$ represents a document's vector. The encoding of the documents into vectors is called indexing. During indexing, a global vocabulary is built up, assigning a unique identification to each word encountered in the entire collection. With this global vocabulary, a vector is constructed for each document with as many elements as the total number of words in the global vocabulary. For words appearing in the document at hand, the value of the respective elements is equal to the number of occurrences of that word in the document. For words not appearing in the document, the respective elements obtain a zero value [70]. In my proposed technique, the vector model applies for only significant words and medical terms, so stop-words are excluded.

Using vector space model, the similarity between two documents (vectors) can be computed by calculating the cosine angle between the two vectors (cosine measure

method). Specifically speaking, the cosine measure is used to quantitatively estimate the relevance of the given two documents. Therefore, to compute the similarity between two documents (vectors), $x_i$and x_j, I can use the cosine similarity measure, which calculate the vectors inner product:

$$\cos\theta = S(x_i, x_j) = \frac{x_i \cdot x_j}{\|x_i\| \|x_j\|} \quad (4.2)$$

where θ is the angle between the two vectors x_i and x_j. Since all vector elements are positive, a word can appear zero or more times in a document, the similarity results are values between one and zero. The similarity between two copies of the same document is one; if the two documents have no words in common, the similarity is zero [70].

The fuzzy relevancy representation and ranking unit measures and calculates the relevancy of each article from the literature set to the one the user selected. In this work, I explore the use of terms meaning and semantic types as two types of features to help for the ranking process, and relevancy calculation. Therefore, each document is represented by two vectors, which are term meaning and semantic type. Since each document is represented by two vectors, it is essential to find a mechanism that facilitates the relevancy representation and ranking process. Consequently, this unit is built with the help of fuzzy inference system (FIS). The fuzzy relevancy representation and ranking unit calculates the cosine similarity between two vectors of the same type for each document from the collection set against the user selected document as in equation (4.2). In other words, the cosine similarity is calculated two times between the user selected document and each document from the

literature set; the first between the terms meaning vectors and the second between the semantic type vectors.

4.3 Implementation of the Fuzzy Logic-Based Ranking Method

Since the UMLS provides a variety of tools that are implemented in JAVA, the fuzzy-logic-based ranking method is built using JAVA programming language and tested it under Linux operating system. As shown in figure 4.1 the PDF articles are converted to text format using the Java JPedal library [68]. The new text files are then normalized via the lexical analyzer unit. The lexical analyzer unit follows several natural language processing actions to filter and normalize the raw text. This unit is built with the help of Lexical and Text Tool provided by the UMLS [49]. Specifically, using the Lexical Variant Generator (LVG), which is java tool, designed to manage lexical variations, normalize, and index the raw text files. The normalization techniques provided by Lexical Variant Generator help to map terms to their concepts identifier in the UMLS Metathesaurus. In this unit, the Lexical Variant Generator is used to 1) Remove Genitive; 2) Remove Plural Patterns; 3) Replace Punctuations with White Spaces; 4) Exclude Stop Words; 5) Filter out Proper Nouns; 6) Un-inflect Terms; 7) Terms Canonicalization; respectively.

The UMLS mapping unit is constructed with the help of the MetaMap application [73] to map the normalized terms to their corresponding concepts in the UMLS Metathesaurus. The MetaMap is a program providing access from normalized biomedical literature to the concepts in the UMLS Metathesaurus. Since MetaMap assigns concept identifiers to all terms in running text, I believe that it can serve a very useful role as a

generator of features, meaning and semantic type, for the fuzzy logic-based ranking method. The UMLS Mapping unit takes the normalized text and maps every term in each article to its corresponding concept identifier and semantic type. The ontology-mapping step constructs a vector for each document. This vector contains the normalized terms' concept identifier called "CUI" vector as in equation (4.1). The semantic type mapping step future builds another vector that holds the semantic type of the normalized terms for every document named "Semantic-Type" vector as in equation (4.1). Therefore, every document is represented by two vectors CUI and Semantic-Type vectors.

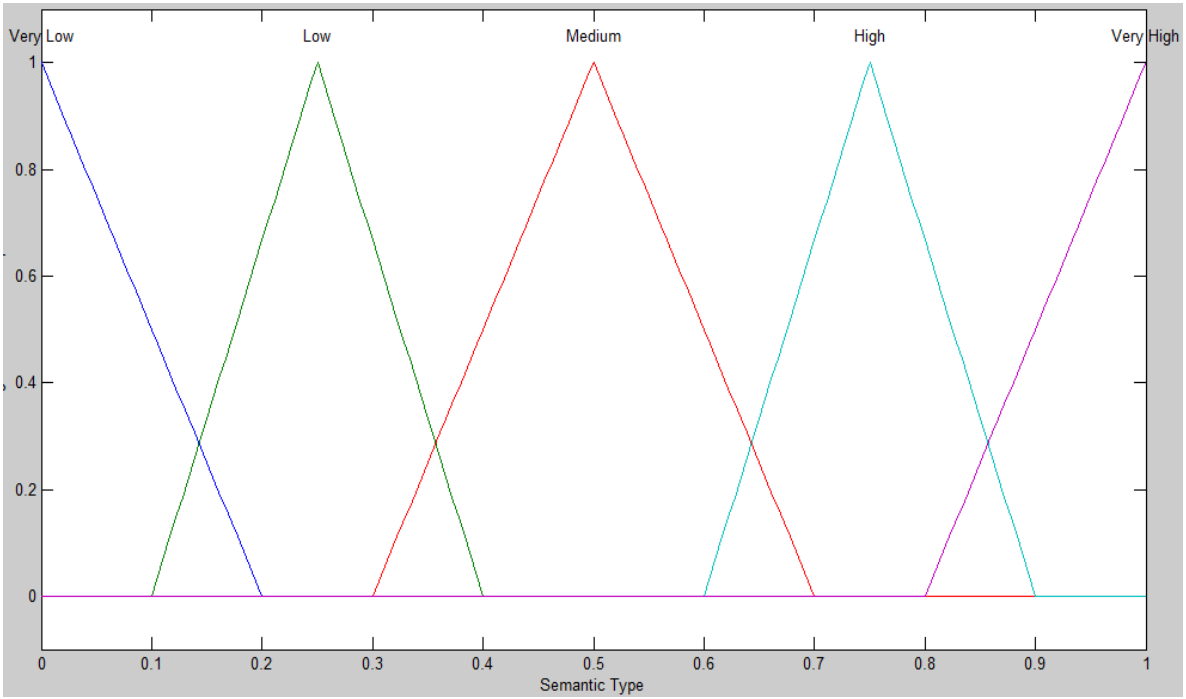

Figure 4.2: Five fuzzy sets representing semantic type membership function

The fuzzy relevancy representation and ranking unit calculates the cosine similarity between two vectors of the same type for each document from the literature set against the user selected document, which means there are two cosine similarity results. Since there

are two cosine similarity results for each document compared against the user selected document, a FIS is constructed to rank the relevancy of the literature set. To build the FIS, five membership functions are used to represent semantic type method: Very Low, Low, Medium, High, and Very High as illustrated in figure 4.2.

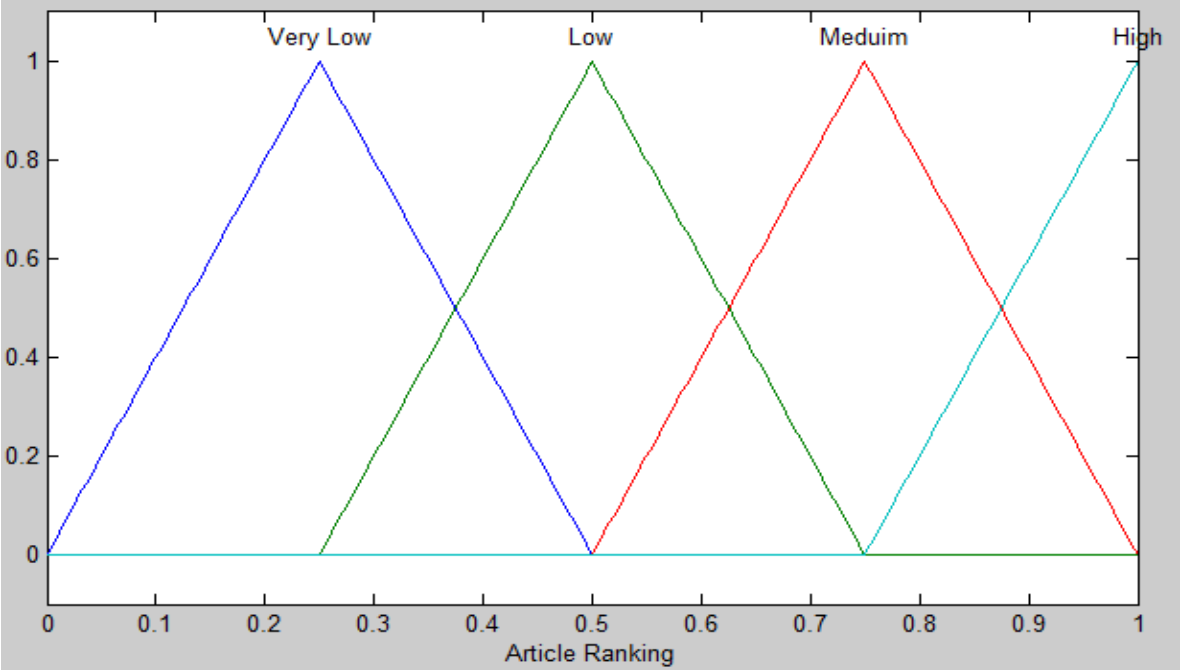

Figure 4.3: Four fuzzy sets representing article ranking membership functions

Further, four membership functions are used to characterize meaning method: Low, Medium, High, and Very High. Figure 4.3 illustrates four fuzzy membership functions that describe the article relevancy- Very Low, Low, Medium, and High. Additionally, in the FIS 20 rules are used as shown in table 4.1. Min-Max is used for fuzzy inference and the centroid defuzzifier. Finally, the relevancy ranking process to the user relevance feedback of whether the selected article is relevant or not is achieved by ranking the result according to the user feedback decision. If the user selected article is relevant, the articles with high

relevancy are those that have the high-ranking degree. Whereas if the user selected document is not relevant, the articles that have the high relevancy degree to the user selected one are considered as being irrelevant; consequently and more probably, the articles that have low relevancy degree to the user selected one are considered to be relevant to his/her preference.

Table 4.1: Rules of the fuzzy inference system

		CUI			
		Low	*Medium*	*High*	*Very High*
Semantic Type	*Very Low*	Very Low	Low	Low	Low
	Low	Very Low	Low	Low	Low
	Medium	Low	Low	Medium	Medium
	High	Low	Medium	Medium	High
	Very High	Low	Medium	High	High

4.4 Experiment Design

The main purpose of this work is to explore how to calculate the relevancy and rank the search result using the fuzzy methodology, and show the result. Therefore, a small document set containing 10 original biomedical documents and 20 synthesized documents from them is created. The number of documents is arbitrary selected. To get consistent result, I consider using documents representing two significantly different medical topics, so documents are retrieved using the keywords Dobutamine (a drug for treating heart failure and cardiogenic shock) and Cancer, which are distinctively different and arbitrary selected. Further, the 20 created documents are varied by adding text from the original 10 documents; for example, add some paragraphs from Dobutamine documents to Cancer documents,

combine several paragraphs from Dobutamine documents to create other Dobutamine documents, and include number of paragraphs from Cancer documents to other Cancer documents. The new documents are crated because the original Cancer and Dobutamine documents include their own distinct terms, such as cancer and dobutamine. Hence, there are no shared significant and medical terms among them. Therefore, without mixing the documents there is no impact of the ranking result. By mixing some paragraphs together to create new documents, I can achieve a document set that provides a comparable ranking result. The added text is selected from the core document sections excluding introduction and reference sections. The document set contains three categories: Dobutamine, Cancer, and Dobutamine-Cancer. The number of words per document in the created documents ranges from 758 to 2,127 words with an average of 1,332 words per document, which are randomly selected. The number of words per document in the original documents ranges from 1,891 to 2,536 words. The average number of paragraphs added to the new created documents is six, which is randomly chosen. Table 4.2 shows the arrangement of documents in the document set; where the numbers in each cell are the document numbers. The documents in each column belong to the same category.

Table 4.2: The arangment of documents in the document set

	Categories		
	Dobutamine	*Cancer*	*Dobutamine-Cancer*
Original	1,2,3,17,18,19,20	4,5,16	----
Synthesized	7,11,12,13,15,24,26, 27,28.29	10,14,25,30	6,8,9,21,22,23

To evaluate the proposed method, two types of experiments are designed: 1) to compare the performance of fuzzy logic-based ranking method against the UMLS meaning and semantic type methods, and 2) to explore the effectiveness of using relevance feedback mechanism in the search process. In the first experiment, the performance of each ranking method is evaluated against gold standard result in order to allow comparison for the methods. Gold standard result is a pre-defined and previously known result. Several experiments are designed using selected documents from the created document set. The performance of the fuzzy logic-based ranking method is evaluated over the UMLS methods by determining the ranking order accuracy of each ranking method. For this experiment, each document in the document set is provided as user selected document and compare the results of the three ranking methods based on their ranking order accuracy against the gold standard result. For the second type of experiments, the relevance feedback mechanism is examined by a) feeding single one related document as relevance feedback, and b) providing unrelated document as user selected choice and explore the retrieved results. In other words, feeding Dobutamine document to the method as related document, the method should provide the user with Dobutamine documents whereas giving Dobutamine document to the method as an unrelated document, it should retrieve Cancer documents.

4.5 Experiment Results

Figure 4.4 depicts the performance of the ranking methods. It is clear from the graph that the fuzzy ranking method performance is the best among the three methods. The ranking order of the fuzzy and UMLS meaning methods are much better than of the UMLS semantic type method. Figure 4.4 shows that the fuzzy ranking method achieves better

performance than UMLS meaning method 17 times, whereas the UMLS meaning method performs better 8 times, and the methods perform equally well 5 times. In some cases, fuzzy ranking method gains as high as 23% better performance than the meaning method and up to 60% than semantic type approach. The fuzzy ranking method achieves an average performance of ranking order accuracy of 3.35% and 29.55% more than UMLS meaning and semantic type methods respectively.

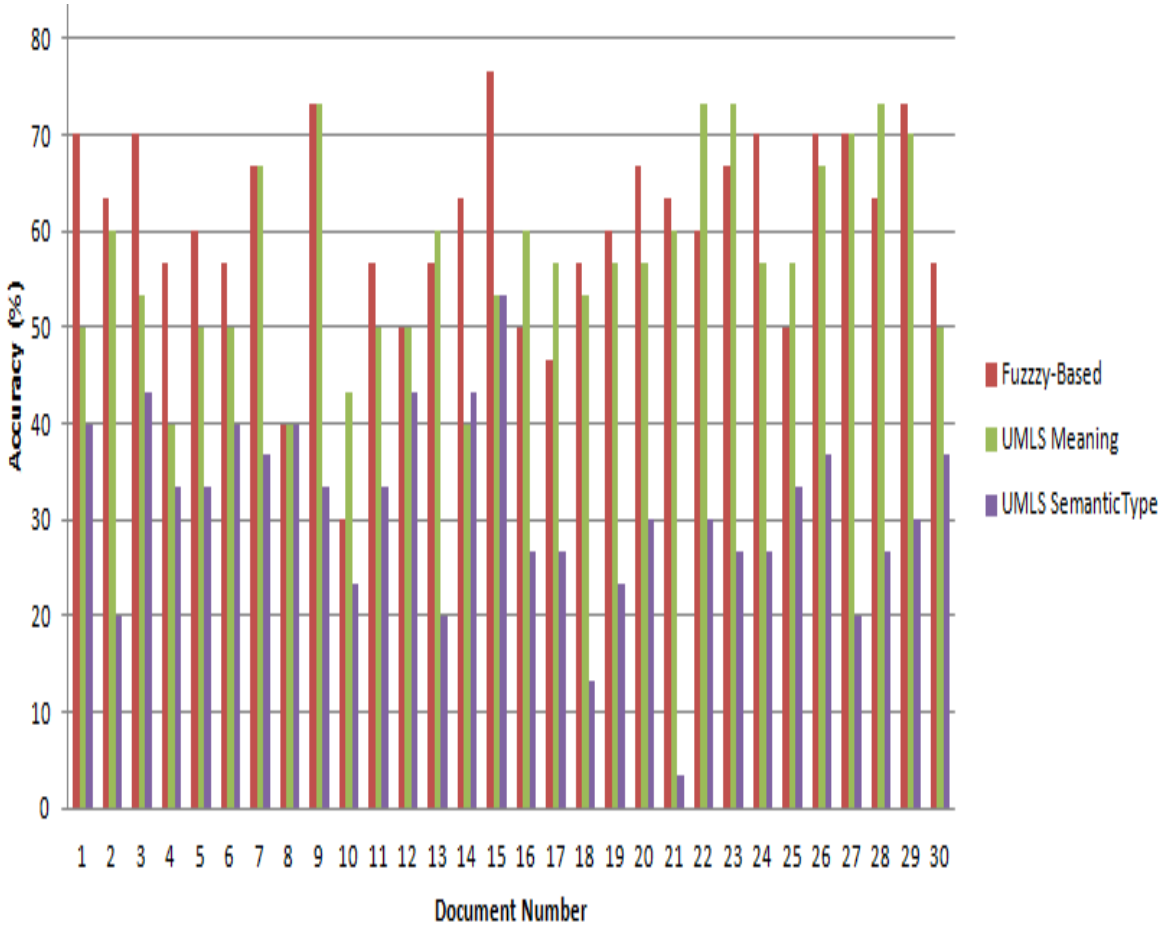

Figure 4.4: The performance comparison of the three ranking methods

I find that the UMLS meaning method ranks some documents as unrelated documents, but in reality they are related, while UMLS semantic type method ranks some

documents as related whereas they should be unrelated. For instance, when providing to the system one Cancer document and examine the results of the three ranking methods. The gold standard order contains the Cancer and some Dobutamine-Cancer documents as relevant documents. The UMLS meaning ranking method result is less than expected where some expected Dobutamine-Cancer documents are marked as less related than they should be, while the UMLS semantic type ranking method result provide some Dobutamine-Cancer document as high related where they should not be. I notice that both fuzzy and UMLS meaning ranking methods rank most of the first five hits correctly whereas UMLS semantic type method ranks only the most first three hits correctly. I believe this is because the fact that the UMLS meaning method provides low-level classification and mapping process while UMLS semantic type method is high-level medical categorization. Therefore, it provides an upper level ontology mapping.

The proposed ranking method was tested in term of relevance feedback. Document number 17 was randomly selected from the created document set, which is Dobutamine document, and was used as relevance feedback. The selected document was marked as "related" to the user's preference. Table 4.3 shows the result of the fuzzy ranking method compared with the gold standard result. The ranking result is measured using the following scale: High, Medium, Low, and Very Low; where High means highly related and Very Low indicates very low relevancy. The numbers in each cell of table 4.3 are the document numbers. The documents in each cell are ranked based on their relevancy score to the user selected document, from high to low. For example, document 26 has a higher relevancy score than document 28. Table 4.3 shows that all the Dobutamine documents are retrieved

as either high or medium related to the user-selected document and Dobutamine-Cancer documents are retrieved with low ranking score; whereas all Cancer documents are retrieved with very low ranking score.

Furthermore, the fuzzy logic-based ranking method was assessed by providing a Dobutamine document as "unrelated" to the user needs. Dobutamine document number 20 was randomly selected from the created document set and was marked as unrelated to the user preference. Table 4.4 shows the result of the proposed approach compared with the gold standard result using the selected Dobutamine document. The result shows that all the Cancer documents are retrieved as highly related, Dobutamine-Cancer documents with less ranking scores and Dobutamine documents with very low ranking scores.

Table 4.3: Ranking result when document 17 is selected as "relevant" by the user via relevance feedback

	High	**Medium**	**Low**	**Very Low**
Gold Standard	26,28,18,29,24,7,11	27,13,15,19,1,12,2,20	3,9,23,21,22,6,8	5,30,25,14,10,16,4
Fuzzy Approach	26,28,18,29,7,24,11	27,13,15,12,1,19,6,20	23,2,9,3,10,14,8	5,21,16,30,25,22,4

Table 4.4: Ranking result when document 20 is selected As "irrelevant" by the user via relevance feedback

	High	**Medium**	**Low**	**Very Low**
Gold Standard	4,25,30,5,14,16,10	21,9,8,6,13	7,1,2,11,17,19,29,15,26	28,12,3,24,18,22,23,27
Fuzzy Approach	4,25,30,16,21,6,5	14,13,1,10,8	7,9,2,11,17,19,29,15,26	28,12,3,24,18,22,23,27

4.6 Summary

In this chapter, a fuzzy logic-based ranking method is developed under UMLS knowledge sources to rank biomedical literature search result. The proposed ranking method is tested using a small biomedical document set that was created. The document set contains 10 original documents and 20 synthesized documents from them. The performance of the proposed ranking method is compared with the UMLS meaning and semantic type methods. Furthermore, the effectiveness of using relevance feedback methodology to match similar and related biomedical documents is investigated. The results have demonstrated the effectiveness of fuzzy logic and UMLS knowledge sources in support for ranking the documents. By experiments, I showed that the fuzzy ranking method provides more accurate result compared with the other methods. Finally, several experimental results were presented that demonstrate the effectiveness and usefulness of the proposed fuzzy logic-based ranking method and mechanisms using the created biomedical document set.

CHAPTER 5

CONCLUSIONS AND FUTURE DIRECTIONS

5.1 Conclusions

In this dissertation, I developed a biomedical literature search system that uses relevance feedback mechanism, fuzzy logic, text mining techniques and Unified Medical Language System. The system is developed to assist healthcare providers to find more related documents using relevance feedback mechanism. The system extracts and decodes information from the documents and uses the extracted information to filter unwanted documents and ranks the related ones based on the user preference. The system has the ability to search large document set and find the most related articles based on the user preference. I used text mining techniques to extract PDF features and use these features to filter unwanted documents with the help of fuzzy logic. The extracted internal features can provide an appropriate way that filters unwanted and unrelated documents and then ranks the documents in a great manner that is more close to what the user wants. The system extracts meaning and semantic relations between texts and calculates the similarity between documents using these relations.

Moreover, I designed and developed a fuzzy logic-base literature ranking method, which can work either with the above-mentioned system or function independently. The ranking mechanism uses fuzzy logic, text mining techniques and Unified Medical Language System. The ranking process is utilized based on fuzzy logic and Unified

Medical Language System knowledge resources. The fuzzy logic-based ranking method uses semantic type and meaning concepts to map the relations between texts in documents.

5.2 Future Directions

The system can be extended and enhanced by using more PDF features that are extracted from the documents such as Journal Impact Factor (JIF), which is an important feature that can improve the retrieval and ranking process of the system. The system can also be improved by filtering the documents based on their type such as review, case study, and randomized CRT.

The system can also be improved by using scanned image documents. The scanned image documents must be converted to a text-searchable format. There exists a method for converting scanned image into text, which is optical character recognition (OCR). The OCR output is not 100% accurate. Therefore, fuzzy logic can be used to handle this issue and then use scanned image documents in the search.

BIBLIOGRAPHY

[1] M. Lee, *et al.*, "Beyond information retrieval--medical question answering," *AMIA Annu Symp Proc*, pp. 469-73, 2006.

[2] S. J. Athenikos and H. Han, "Biomedical question answering: A survey," *Comput Methods Programs Biomed*, Nov 12 2009.

[3] J. P. Harrison and K. Radcliffe, "Evidence based medicine as a strategy for quality improvement," *International Journal of Public Policy*, vol. 5, pp. 133 - 142, 2010.

[4] PubMed Dtabase [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/

[5] L. Hunter and K. B. Cohen, "Biomedical language processing: what's beyond PubMed?," *Mol. Cell*, vol. 21, pp. 589–594, 2006.

[6] Z. Lu, "PubMed and beyond: a survey of web tools for searching biomedical literature," *Database, doi:10.1093/database/baq036*, 2011.

[7] A. D. Corlan. (2004, 2013-07-26). *Medline trend: automated yearly statistics of PubMed results for any query*. Available: Web resource at URL:http://dan.corlan.net/medline-trend.html. (Archived by WebCite at http://www.webcitation.org/65RkD48SV)

[8] K. R. McKeown, *et al.*, "Leveraging a Common Representation for Personalized Search and Summarization in a Medical Digital Library," presented at the Third ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL'03), 2003.

[9] E. A. Mendonca and J. J. Cimino, "Building a knowledge base to support a digital library," *Stud Health Technol Inform*, vol. 84, pp. 221-5, 2001.

[10] D. Demner-Fushman and J. Lin, "Situated Question Answering in the Clinical Domain: Selecting the Best Drug Treatment for Diseases," presented at the In Proceedings of COLING/ACL 2006 Workshop on Task-Focused Summarization and Question Answering, Sydney, Australia, 2006.

[11] D. Islamaj, R.*, et al.*, "Understanding PubMed user search behavior through log analysis," *Database doi:10..1093/database/bap018.*, 2009.

[12] S. L. De Groote and J. L. Dorsch, "Measuring Use Patterns of Online Journals and Databases," *Journal of the Medical Library Association,* vol. 91, pp. 231-240 2003.

[13] Y. NIU. and G. Hirst., "Analysis of Semantic Classes in Medical Text for Question Answering," in *Proceedings of the ACL 2004 Workshop on Question Answering in Restricted Domains*, 2004, pp. 54–61.

[14] J. Dismukes, "How can medical libraries become more relevant in the age of digital information?," *Library Student Journal,* 2009.

[15] A. Hoogendam*, et al.*, "Analysis of queries sent to PubMed at the point of care: observation of search behaviour in a medical teaching hospital," *BMC Med Inform Decis Mak,* vol. 8, p. 42, 2008.

[16] R. Hoffmann and A. Valencia, "Implementing the iHOP concept for navigation of biomedical literature," *Bioinformatics,* vol. 21 Suppl 2, pp. ii252-8, Sep 1 2005.

[17] M. Ongenaert*, et al.*, "PubMeth: a cancer methylation database combining text-mining and expert annotation," *Nucleic Acids Res,* vol. 36, pp. D842-6, Jan 2008.

[18] M. He*, et al.*, "PPI finder: a mining tool for human protein-protein interactions," *PLoS One,* vol. 4, p. e4554, 2009.

[19] L. Hunter and K. B. Cohen, "Biomedical language processing: what's beyond PubMed?," *Mol Cell,* vol. 21, pp. 589-94, Mar 3 2006.

[20] M. L. Chambliss and J. Conley, "Answering clinical questions," *J Fam Pract,* vol. 43, pp. 140-4, Aug 1996.

[21] G. Salton and C. Buckley, "Improving Retrieval Performance by Relevance Feedback," *Journal of the American Society for Information Science and Technology,* vol. 41, pp. 288-297, 7 JAN 1990.

[22] W. B. Croft*, et al.*, "Relevance feedback and personalization: A language modeling perspective.," in *Second DELOS Workshop: Personalisation and Recommender Systems in Digital Libraries*, 2001.

[23] K. Shin*, et al.*, "Advanced Relevance Feedback Query Expansion Strategy for Information Retrieval in MEDLINE," presented at the CIARP, Springer-Verlag, 2004.

[24] N. V. NGUYEN*, et al.*, "Text Retrieval Relevance Feedback Techniques for Bag of Words Model in CBIR," presented at the International Conference on Machine Learning and Pattern Recognition (ICMLPR), Paris (France), 2009.

[25] Chen Chen*, et al.*, "Relevance Feedback Fusion via Query Expansion," presented at the IEEE/ACM International Conference on Web Intelligence and Intelligent Agent technology, Macau, 2012.

[26] Z. Minjuan, "Selecting Good Expansion Terms for Improving XML Retrieval Performance," presented at the 2012 International Conference on Control Engineering and Communication Technology, Liaoning, 2012.

[27] ZHANG Zhi-yong and Y. Bai-lin, "A Log-Based Relevance Feedback Scheme For 3D Model Retrieval," presented at the 2010 WASE International Conference on Information Engineering, Beidaihe, Hebei, 2010.

[28] Y.-L. Chang and J.-T. Chien, "Language Model Adaptation for Relevance Feedback in Information Retrieval," presented at the 2008. ISCSLP '08. 6th International Symposium on Chinese Spoken Language Processing, Kunming, 2008.

[29] WANG Xiao-Gang and L. Yue, "Relevance Feedback on Keyword Space for Interactive Information Retrieval," in *2009 IITA International Conference on Services Science, Management and Engineering*, Zhangjiajie, China, 2009.

[30] X. Wang, *et al.*, "Improve retrieval accuracy for difficult queries using negative feedback," in *CIKM '07 Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, 2007, pp. 991-994.

[31] C. Li and J.-y. Wang, "A Clustering Approach to Improving Pseudo-Relevance Feedback Improving Retrieval Effectiveness by Removing Noisy Documents," presented at the 2012 International Symposium on Information Science and Engineering (ISISE), Shanghai, 2012.

[32] Jun-yi Wang and X.-m. Ye, "The Study of Methods for Language Model Based Positive and Negative Relevance Feedback in Information Retrieval," presented at the 2010 IEEE International Conference on Intelligent Computing and Intelligent Systems (ICIS) Xiamen, 2010.

[33] W.-j. Zhang and J.-y. Wang, "The Study of Methods for Language Model Based Positive and Negative Relevance Feedback in Information Retrieval," in *ISISE '12 Proceedings of the 2012 Fourth International Symposium on Information Science and Engineering*, 2012, pp. 39-43.

[34] S. M. Bidok and S. M. R. Moosavi, "IDUF: an Active Learning Based Scenario for Relevance Feedback Query Expansion," presented at the 2012 International Conference on Information Retrieval & Knowledge Management (CAMP), Kuala Lumpur, 2012.

[35] I-Chin Wu*, et al.*, "On Learning Researchers' Dynamic Information Needs: An Entropy-based Query Expansion Approach," presented at the 2012 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology, acau, China China, 2012.

[36] Chun-Bo Liu*, et al.*, "A Relevance Feedback Retrieval system based on indri toolkit," presented at the 2011 IEEE 3rd International Conference on Communication Software and Networks (ICCSN), Xi'an, 2011.

[37] C. von Mering*, et al.*, "STRING: known and predicted protein-protein associations, integrated and transferred across organisms," *Nucleic Acids Res,* vol. 33, pp. D433-7, Jan 1 2005.

[38] T. C. Rindflesch*, et al.*, "EDGAR: extraction of drugs, genes and relations from the biomedical literature," *Pac Symp Biocomput,* pp. 517-28, 2000.

[39] T. K. Jenssen*, et al.*, "A literature network of human genes for high-throughput analysis of gene expression," *Nat Genet,* vol. 28, pp. 21-8, May 2001.

[40] Y. C. Fang, *et al.*, "MeInfoText: associated gene methylation and cancer information from text mining," *BMC Bioinformatics,* vol. 9, p. 22, 2008.

[41] F. M. Couto, *et al.*, "GOAnnotator: linking protein GO annotations to evidence text," *J Biomed Discov Collab,* vol. 1, p. 19, 2006.

[42] D. Rebholz-Schuhmann, *et al.*, "EBIMed--text crunching to gather facts for proteins from Medline," *Bioinformatics,* vol. 23, pp. e237-44, Jan 15 2007.

[43] J. R. Hobbs and E. Riloff, "Information Extraction," in *Handbook of Natural Language Processing*, N. Indurkhya and F. J. Damerau, Eds., Second Edition ed Boca Raton, Florida: CRC Press, 2010, pp. 511-532.

[44] M. A. Harris, *et al.*, "The Gene Ontology (GO) database and informatics resource," *Nucleic Acids Res,* vol. 32, pp. D258-61, Jan 1 2004.

[45] J. Choi, *et al.*, "BOSS: A Biomedical Object Search System," in *DTMBIO 11*, 2011.

[46] D. J. States, *et al.*, "MiSearch adaptive pubMed search tool," *Bioinformatics,* vol. 25, pp. 974–976, 2009.

[47] H. Yu, *et al.*, "2010," *BMC Bioinformatics,* vol. 11 (Suppl. 2), S6, Enabling multi-level relevance feedback on PubMed by integrating rank learning into DBMS.

[48] Y. Lv and C. Zhai, "Adaptive Relevance Feedback in Information Retrieval," in *Proceedings of the 18th ACM International Conference on Information and Knowledge Management (CIKM'09)*, 2009, pp. 255-264.

[49] Unified Medical Language System® (UMLS®) [Online]. Available: http://www.nlm.nih.gov/research/umls/

[50] D. E. Appelt, "Introduction to Information Extraction," *AI Communications,* vol. 12, pp. 161-172, 1999.

[51] Massuod Alatrash*, et al.*, "Application of Type-2 Fuzzy Logic to Healthcare Literature Search at Point of Care," in *North American Fuzzy Information Processing Society*, El Paso, TX, 2011, pp. 1-5.

[52] Massuod Alatrash*, et al.*, "Ranking Biomedical Literature Search Result Based on Relevance Feedback Using Fuzzy Logic and Unified Medical Language System," in *North American Fuzzy Information Processing Society*, Berkley, CA, USA, 2012, pp. 1-6.

[53] L. A. Zadeh, "Fuzzy sets," *Information and Control,* vol. no. 8, pp. 338–353, 1965.

[54] L. A. Zadeh, "The concept of a linguistic variable and its application to approximate reasoning, Parts 1, 2, and 3," *Information Sciences,* vol. 8, 9, pp. 199-249, 301-357,43-80, 1975.

[55] C. Moraga, "Introduction to Fuzzy Logic," *SER.: ELEC. ENERG,* vol. 18, pp. 319-328, 2005.

[56] J. M. Mendel. (FEBRUARY 2007) Type-2 Fuzzy Sets and Systems: An Overview. *IEEE COMPUTATIONAL INTELLIGENCE MAGAZINE* 20-29.

[57] N. N. Karnik and J. M. Mendel, "Introduction to Type-2 Fuzzy Logic Systems," in *Fuzzy Systems Proceedings, 1998. IEEE World Congress on Computational Intelligence.*, 1998, pp. 915 - 920.

[58] j. M. Mendel, *Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions*, 1 ed.: Prentice Hall PTR, 2001.

[59] J. M. Mendel*, et al.*, "Interval Type-2 Fuzzy Logic Systems Made Simple," *IEEE TRANSACTIONS ON FUZZY SYSTEMS,* vol. 14, pp. 808-821, DECEMBER 2006.

[60] J. R. Castro*, et al.*, "Interval type-2 fuzzy logic toolbox," *Journal of Engineering Letters,* vol. 15, pp. 89 - 98 online version., August 2007.

[61] A. R. Aronson and T. C. Rindflesch, "Query expansion using the UMLS Metathesaurus," *Proc AMIA Annu Fall Symp,* pp. 485-9, 1997.

[62] A. Myosho*, et al.*, "Semantic Classification of Nouns in UMLS Using Google Web 1T 5-gram," presented at the 20th International Conference on Genome Informatics, Yokohama Pacifico, Japan, 2009.

[63] O. Bodenreider, "Using UMLS semantics for classification purposes," *Proc AMIA Symp,* pp. 86-90, 2000.

[64] W. Pratt, "Dynamic organization of search results using the UMLS," in *Proc AMIA Annu Fall Symp*, 1997, pp. 480-4.

[65] D. Demner-Fushman and J. Lin, "Answer Extraction, Semantic Clustering, and Extractive Summarization for Clinical Question Answering," in *the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, Sydney, Australia, 2006, pp. 841 - 848.

[66] Q. Liang and J. M. Mendel, "Interval Type-2 Fuzzy Logic Systems: Theory and Design," *IEEE TRANSACTIONS ON FUZZY SYSTEMS,* vol. 8, pp. 535-550, October 2000.

[67] T. Reuters. (2010, April/2010). *EndNote, Bibliographies Made Easy*. Available: http://www.endnote.com/enhome.asp

[68] IDRsolutions. (1999, 05/2010). *Java PDF Extraction Decoding Access Library (JPEDAL)*. Available: http://www.jpedal.org

[69] (2011, *RANKS NL - WEBMASTER TOOLS*. Available: http://www.ranks.nl/resources/stopwords.html

[70] T. Magerman, *et al.*, "Exploring the feasibility and accuracy of Latent Semantic Analysis based text mining techniques to detect similarity between patent documents and scientific publications," *Scientometrics*, vol. 82, pp. 289-306, 2010.

[71] M. W. Beery and M. Brown, "Understanding Search Engines: mathematical Modeling and Text Retrieval," presented at the Society for Industrial and Applied Mathematics, Philadelphia, 1999.

[72] G. Salton and M. J. McGill, *Introduction to Modern Information Retrieval*. New York: McGraw-Hill, 1983.

[73] A. R. Aronson, "Effective mapping of biomedical text to the UMLS Metathesaurus: the MetaMap program," *Proc AMIA Symp*, pp. 17-21, 2001.

ABSTRACT

A RELEVANCE FEEDBACK-BASED SYSTEM FOR QUICKLY NARROWING BIOMEDICAL LITERATURE SEARCH RESULT

by

MASSUOD HASSAN ALATRASH

December 2013

Advisor: Prof. Hao Ying

Major: Computer Engineering

Degree: Doctor of Philosophy

The online literature is an important source that helps people find the information. The quick increase of online literature makes the manual search process for the most relevant information a very time-consuming task and leads to sifting through many results to find the relevant ones. The existing search engines and online databases return a list of results that satisfy the user's search criteria. The list is often too long for the user to go through every hit if he/she does not exactly know what he/she wants or/and does not have time to review them one by one. My focus is on how to find biomedical literature in a fastest way. In this dissertation, I developed a biomedical literature search system that uses relevance feedback mechanism, fuzzy logic, text mining techniques and Unified Medical Language System. The system extracts and decodes information from the online biomedical documents and uses the extracted information to first filter unwanted documents and then ranks the related ones based on the user preferences. I used text

mining techniques to extract PDF document features and used these features to filter unwanted documents with the help of fuzzy logic. The system extracts meaning and semantic relations between texts and calculates the similarity between documents using these relations. Moreover, I developed a fuzzy literature ranking method that uses fuzzy logic, text mining techniques and Unified Medical Language System. The ranking process is utilized based on fuzzy logic and Unified Medical Language System knowledge resources. The fuzzy ranking method uses semantic type and meaning concepts to map the relations between texts in documents. The relevance feedback-based biomedical literature search system is evaluated using a real biomedical data that created using dobutamine (drug name). The data set contains 1,099 original documents. To obtain coherent and reliable evaluation results, two physicians are involved in the system evaluation. Using (30-day mortality) as specific query, the retrieved result precision improves by 87.7% in three rounds, which shows the effectiveness of using relevance feedback, fuzzy logic and UMLS in the search process. Moreover, the fuzzy-based ranking method is evaluated in term of ranking the biomedical search result. Experiments show that the fuzzy-based ranking method improves the average ranking order accuracy by 3.35% and 29.55% as compared with UMLS meaning and semantic type methods respectively.

AUTOBIOGRAPHICAL STATEMENT
MASSUOD HASSAN ALATRASH

EDUCATION

- Doctor of Philosophy in Computer Engineering, December 2013, Wayne State University, Detroit, Michigan, United States
- Master of Science in Computer Science, 2002, University of Newcastle, Newcastle Upon Tyne, UK
- Bachelor of Science in Electrical and Computer Engineering, 1996, University of Sebha, Sebha, Libya

RESEARCH INTERESTS

Massuod H. Alatrash's main research interest areas are fuzzy systems and fuzzy logic, intelligent systems, machine learning, natural language processing, data mining, and biomedical text mining.

AWARDS AND HONORS

- Golden Key Honor Society, 2011
- Thomas C. Rumble University Graduate Fellowships, Wayne State University, 2012
- Graduate Student Professional Travel Award, Wayne State University, 2011 and 2012
- Summer University Graduate Fellowships, Wayne State University, 2013