DEVELOPING A PERSONALIZED ARTICLE RETRIEVAL SYSTEM FOR PUBMED

By

Sachintha Pitigala

A dissertation submitted in partial fulfillment

of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

Computational Science

Middle Tennessee State University

August 2016

Dissertation Committee:

Dr. Cen Li, Committee Chair

Dr. Suk Seo

Dr. John Wallin

Dr. Qiang Wu

ACKNOWLEDGEMENTS

Firstly, I thank my advisor Dr. Cen Li, Professor of Computer Science, for her unrelenting advice and support throughout the past six years. I would like to thank Dr. John Wallin, Director of the Computational Science Program, for his invaluable advice and support throughout the years. My special gratitude also goes to Dr. Suk Seo and Dr. Qiang Wu for their advice in presenting this thesis. Further, I would like to take the opportunity to thank all the professors in the computational science program for teaching me new and valuable materials to successfully finish my PhD studies. My heartfelt gratitude also goes to all who participated in the study of the PARS system by volunteering their time to evaluate the search outputs. I would also like to thank all the academic and nonacademic staff of the Department of Computer Science, College of Basic and Applied Sciences and the Graduate School for supporting me in various ways with regards to my studies throughout the past six years. Finally, I thank my parents for their support and encouragement over the years.

ABSTRACT

PubMed keyword based search often results in many citations not directly relevant to the user information need. Personalized Information Retrieval (PIR) systems aim to improve the quality of the retrieval results by letting the users supply information other than keywords. Two main problems have been identified for the current PIR systems developed for PubMed: (1) requiring the user to supply a large number of citations directly relevant to a search topic, and (2) producing too many search results, with a high percentage being false positives. This study developed a Personalized Article Retrieval System (PARS) for PubMed to address these problems. PARS uses two main approaches to find the relevant citations to the given information need: (1) Extending the PubMed Related Article (PMRA) feature and (2) Text classification based Multi Stage Filtering (MSF) method. Both approaches require only a small set of citations from the user, and reduce the search output size by eliminating the false-positive citations in the search output. PARS has been experimentally evaluated using the TREC 2005 dataset, and empirically evaluated by subject experts from the biomedicine field. Results show the PARS system is able to produce retrieval results of better quality than the existing PIR systems for PubMed.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Scientific literature databases had an exponential growth over the past decade. Google Scholar [1], PubMed [2], The SAO/NASA Astrophysics Data System [3] and CiteSeerX [4] are some of the popular citation databases on the internet. These online databases open a new way of accessing and searching for the information for the scientific community. Before this era, scientists need to go to the library and search through the library catalogs to find the relevant citation manually. Now, researchers routinely find literature and information by providing a few keywords to the online databases.

PubMed is the largest and most comprehensive literature database in the field of biomedicine. Biomedicine researchers use PubMed as their primary tool to search for published articles matching their research interests. PubMed is developed and maintained by the National Center for Biotechnology Information (NCBI) at the National Library of Medicine (NLM) [5]. NLM started to index biomedical and life science journal articles in 1960's. The indexed citations were kept in the Medline citation database. Currently, Medline provides access to over 21 million citations dating back to 1946 [6]. In 1996, NCBI at NLM introduced PubMed citation database. PubMed provides access to over 24 million citations in the field of biomedicine [2]. Primarily, it allows free access to Medline citation database via its web interface. PubMed contains more citations than the Medline database, covering the in-progress Medline citations, out of scope citations, "Ahead of Print" citations, and NCBI bookshelf citations.

Typical users of PubMed search for relevant articles to their specific research interests by entering query terms at the PubMed web interface. Often times, too many citations were returned as a result of the query with many of the returned citations not directly relevant to the information need. For example, over one-third of PubMed queries returned 100 or more citations [7]. Sifting through these citations to locate the ones that represent the

most relevant articles for one's query can be a tedious and time-consuming process. This problem is becoming more challenging due to the rapid growth of the size of the PubMed database. To improve the quality of information retrieval from PubMed, NCBI and other academic and industry groups have recently developed many tools to enhance the traditional keyword-based information retrieval method [8, 10, 15, 16].

Two main approaches have been used to enhance the information retrieval in PubMed. The first approach builds supplementary tools for the original PubMed search interface. These supplementary tools are focused on different aspects of the PubMed web search interface. For example, PubMed Advanced Search Feature [8, 9] PubMed Auto Query Suggestions [10], PubMed Automatic Term Mapping (ATM) [11] and Medical Subject Headings (MeSH) [12] are some of the PubMed supplementary tools developed to assist the users to express their information need accurately and wisely. PubMed filters sidebar in the PubMed web interface provides varies types of filters to narrow down the PubMed search output [13]. PubMed Related Article (PMRA) [14] feature is another supplementary tools can be found in the PubMed web interface. It helps the users to find additional citations relevant to those they have selected from the search results. Even though NCBI has been continually improving the PubMed web interface and its features, still, the majority of users found it difficult to find information they really look for from the PubMed search output due to information overload [7, 10].

A number of search tools complementary to the PubMed web interface have been built to enhance the information retrieval task. Some popular approaches in building complementary tools are based on text classification methods [15, 16, 17], semantic based methods [18, 19] and special input (set of genes or set of protein names) based methods [20, 21]. MedlineRanker [15], MScanner [16], PubFinder [17], Caipirini [20] and MedEvi [18] are examples in this category. These complementary search tools have shown tremendous improvement in PubMed information retrieval as opposed to the PubMed web interface.

Most of the complementary search tools for PubMed captures the user information need using an extensive input other than keywords. For example, a text paragraph, a set of documents, a set of relevant gene names and organism names are some of the extensive inputs. This type of input method will feed additional information to the search tools. Also, the PubMed complementary tools are capable of delivering a relatively small search output compared to the PubMed interface. These types of complementary search tools are referred as Personalized Information Retrieval (PIR) tools within the information retrieval community [78, 79].

PIR tools are capable of capturing each user's unique research interest and returning a smaller set of citations of the truly relevant articles from large literature databases such as PubMed. PIR systems consist of two main tasks. The first task is to capture and understand a user's information need accurately and more thoroughly. The second task is to find the citations closest to the user information need and present these citations in a meaningful manner.

For the traditional Information Retrieval (IR) systems, user information needs are provided as user queries consisting of keyword terms. However, it is often difficult to fully capture and identify user information need precisely using only the query terms. Additional information about the intention of the user query is required for the PIR systems to deliver more personalized results. There are two main approaches to gather the individual user interest for the PIR systems. The first approach is to capture the user interest explicitly [20, 15, 16]. The second approach collects information about the intention of the user query implicitly; for example, in terms of the click-through links in the search history [23, 24, 25]. Either explicit or implicit information provides a deeper understanding about the user information need to the PIR systems. The second stage of the PIR system carefully filters and sorts the relevant information to the user information need using more sophisticated text mining algorithms. Text classification [15, 16], text clustering [26, 27], relevance ranking

[22, 28] and semantic based methods [18, 19] are some of the text mining techniques used in the PIR systems.

PubMed's Advanced Search Feature [8] can be considered as the simplest PIR system in PubMed towards personalized retrieval. With PubMed's Advanced Search Feature, user may explicitly enter the area of interest, publication period, journals or authors of interest, Medical Subject Heading (MeSH) [12] terms along with the query terms. This additional information about the user query helps to state the information need more clearly in the search tools. Also it helps further filter the search output, thus reduces the size of the search output. The assumption for this approach is that the user has proficient knowledge and vocabulary in the search area to state the advanced search accurately. Otherwise, it may lead to search results less relevant than those from the basic PubMed search.

It is possible to get more explicit information about the user's query intent in an attempt to deliver more personalized results. For example, explicit PIR systems allow the users to enter a text paragraph or the abstract of a research article, instead of few keywords, to explain his or her information need. eTBLAST [22] is an explicit PIR tool for PubMed based on free text inputs. This form of inputs leads to better results compared to the traditional keyword based method and advanced search method. However, eTBLAST is mainly focused on ranking the search output according to the text similarity to the input. It is not directly addressing the problem of reducing the size of search results in the context of personalized information retrieval.

Free text input such as a couple of sentences may not be the optimal way to gather the user information need for a PIR system. Instead, user input in the form of a number of abstracts is a more efficient way to get an extensive input for a PIR system. In practice, most researchers, even beginners, in the biomedicine field know a few or couple dozen articles directly pertinent to their current study. Therefore, for both expert and non-expert users, it is easier to form the information need using these articles when using PubMed. Currently,

a few systems have been built based on this approach, namely, PubFinder [17], MScanner [16] and MedlineRanker [15].

PubFinder [17] is one of the earliest PIR systems that uses text mining and ranking approach. First, a reference dictionary is created which contains the most frequent 100,000 words in the PubMed abstracts. PubFinder takes a set of representative PubMed abstracts from the user as an input. Then a word-list is obtained from the abstracts the user provides. Next, a likelihood value for each abstract in PubMed is calculated using the mutual words in the reference dictionary and the derived word-list. Finally, it sorts the search output according to the likelihood values. PubFinder can be used without expert knowledge and it is not required to specify any special keyword to explain the information need. However, it needs the user to provide a sufficiently large set of abstracts to produce accurate results. Also, it is not focused on filtering or refining the search output by incorporating the user's personalized information. PubFinder mainly aimed at providing a ranking of the search output. Moreover, PubFinder needs a lot of computations to produce an output and its reference dictionary has to be updated frequently.

MScanner [16] is PIR tool that takes a set of user supplied abstracts as input. Figure 1 shows the web interface of MScanner. It first extracts the Medical Subject Heading (MeSH) [12] annotations attached to the input abstracts. Next, a set of abstracts is selected using the cross validation technique and labeled as the background set – Naive Bayes classifier is trained using the input set and the background set. Then it finds more relevant abstracts from PubMed using the trained Naive Bayes classifier. Finally, it sorts the output according to the likelihood probabilities. MScanner uses MeSH annotations to represent the abstracts rather than words in the abstracts. Because of this, MScanner is much faster than PubFinder [17]. However, it cannot handle abstracts with missing or incomplete MeSH annotations. MScanner also requires a large amount of user-defined abstracts in order to produce accurate results.

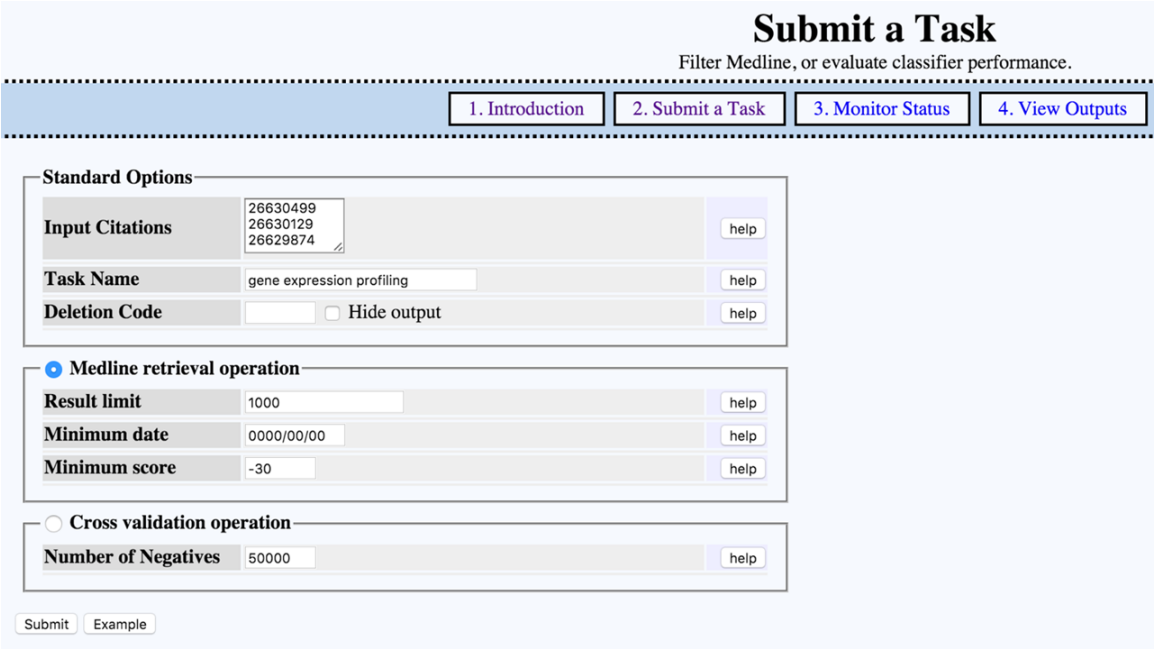

Figure 1: The web interface of MScanner. The user is required to provide PubMed IDs of relevant citations in the first text box. In the next option, the user can select the number of citations in the search output.

MedlineRanker [15] was built based on the MScanner [16] approach, but it uses words appearing in the abstract rather than the MeSH annotations. Therefore, MedlineRanker is able to handle abstracts with missing or incomplete MeSH annotations. Figure 2 shows the web interface of MedlineRanker. It also needs more than 100 abstracts to produce reasonable results. Moreover, MedlineRanker is slower than the MScanner search tool.

1.1 Motivation for the Thesis Work

Most of the current PIR systems require a large set of user-defined abstracts as input [15, 16]. However, in practical situations, it may not be feasible for a user to find hundreds of abstracts directly relevant to his or her information need. It is desirable for a researcher to work with a Personalized Information Retrieval (PIR) system that can learn from a small set of input citations. Also, it should limit the search output size by eliminating the false-positives in the search output using the small set of input citations.

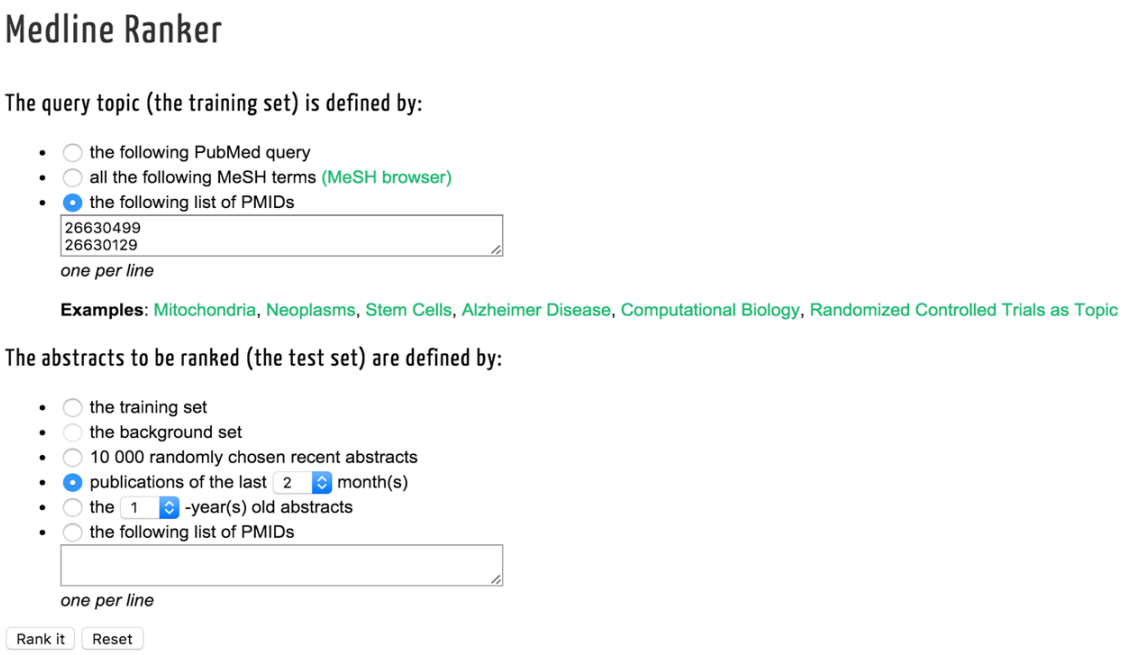

Figure 2: The web interface of MedlineRanker. A user needs to provide a list of relevant PMIDs and the test set as the input.

Few of the current PIR systems for PubMed; for example, Anne O'Tate [26], McSyBi [27] and GOPubMed [29] focus on reducing the search output size, by clustering the search outputs into meaningful categories. Clustering the search output provides a quick way for users to navigate through the search output. However, finding and locating the correct cluster of citation is not a trivial process. It is time consuming and requires proficient domain knowledge. Therefore, PubMed can be greatly enhanced with a PIR tool that directly focuses on reducing the search output size solely based on the initial user input.

In addition to the explicit PIR systems for PubMed discussed above, there are few implicit systems developed for PubMed personalized information retrieval [30, 31]. Implicit PIR tools need a lot of implicit data about the user profile to produce an accurate result. These systems infer the user query intention through browsing history, click-through links or by previous search entries. In order for the user profile data to be useful, it often requires the collection of one or two month of user behavioral data, which may not be readily available.

1.2 Objectives

The goal of this study is to build an efficient Personalized Article Retrieval System (PARS) for PubMed that can be used by both expert and non-expert users:

- It learns from user input with a small number of relevant articles.
- It enhances search output quality by removing the irrelevant citations from the search output purely based on the small set of citations and no additional information from the user.

The system is designed with the following considerations:

1. Uniquely infer and represent user information need in terms of a small set of user input citations.
2. Expand training data based on the user input citation set automatically.
3. Find the relevant citations to the user information need by applying text-mining algorithms.
4. Reduce and refine the relevant citations derived using the additional filtering processes.
5. Present the final search output ranked according to each citation's similarity value to the user information need.

Figure 3 shows a high-level architectural view of the system developed based on these considerations.

The first consideration of the proposed PARS system is that it can be trained based on a small set of citations such as between 5 and 20 citations – provided by the user. We refer to this set as "the *seed* documents." The PARS system gathers the information about the user query intention based on this small set of *seed* documents. This process is critical for the overall performance of PARS. Typically, a user will enter the PubMed IDs of the

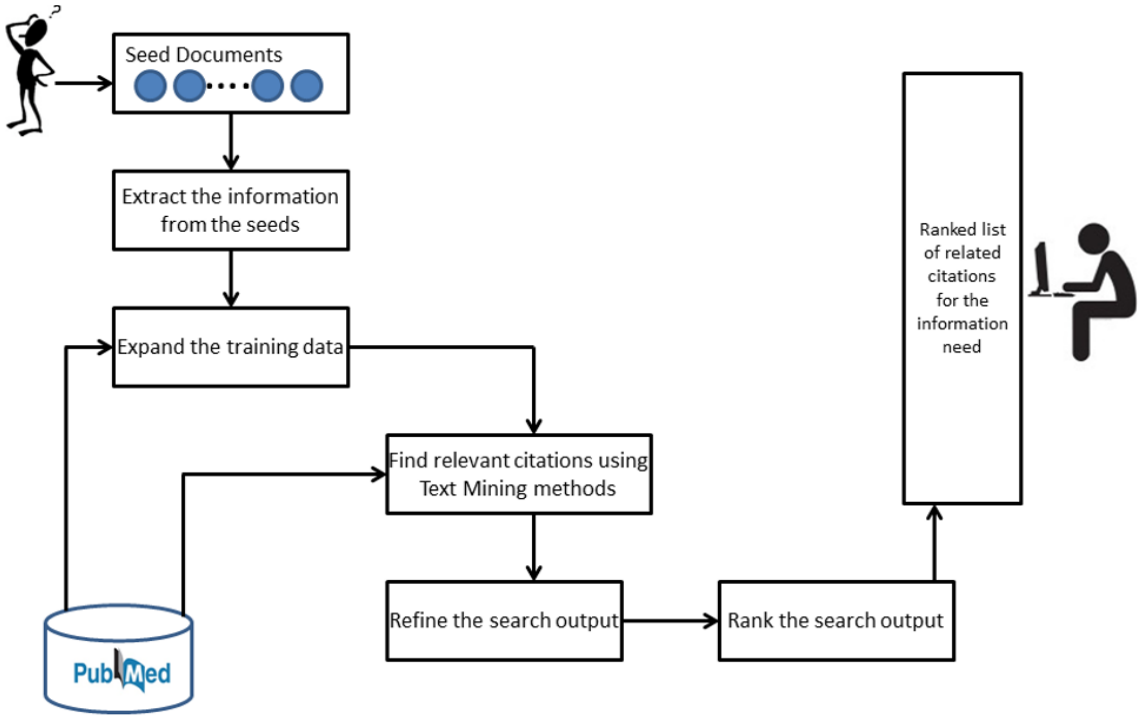

Figure 3: Abstract view of the proposed Personalized Article Retrieval System (PARS).

seed documents in the PARS interface. Based on the *seed* PubMed IDs, PARS extracts the citation title, abstract and all the meta-data attached to the PubMed citation such as MeSH terms. Then PARS uses this information collectively to represent the information need.

In the next step, PARS trains its text mining and text similarity algorithms to find the relevant documents to the information need. It is well known to the text mining community, it is difficult to train text-mining algorithms with good accuracy based on small data set. Therefore, the second consideration of the proposed system is to develop an automated procedure to expand the training data based on user *seed* citations. The current approach to this problem is to allow for interactive user inputs. That is, system will initially present some citations to the user. Then the user is asked to pick relevant citations to his or her information need from the presented citations. The user selected citations are added to the training set. This process is repeated several iterations until PIR tool get sufficient training data. The search tool RefMed [32] was developed based on this multi-level relevance feedback

with the RankSVM [33] learning method. However, this approach is time consuming and tiresome and requires proficient knowledge about the topic being searched. In addition, users need to be cautious about the feedback inputs. Less relevant abstracts admitted into the data set may decrease the accuracy of the classifier learned. After multiple iterations of inaccurate feedbacks, the classifier may produce final results irrelevant of the initial information need. When designing the proposed PARS system, we decided that this process is to be automated using the text mining algorithms without any intermediate user input.

The third consideration of the proposed PARS system is to adapt and develop text classifiers or other text mining methods to find the relevant citations from PubMed. When the PARS system finds new articles from PubMed, a properly trained text classifier can differentiate the relevant articles from the irrelevant ones. Currently, MedlineRanker [15] and MScanner [16] use Naive Bayes text classifier in their search tool.

Typically, a single text classifier produces a large set of relevant citations as a result of the large volume of citations in PubMed. Therefore, the fourth consideration for the proposed PARS system is to develop a process to refine and reduce the search output by removing false-positives in the search output. A multi-stage filtering procedure is developed for this purpose.

The search output from the proposed multi-stage filtering process is a much-improved set of highly relevant citations to the user information need. However, it may still contain some of the false-positives. Therefore, the final and fifth consideration of the proposed system is to develop an efficient ranking procedure to rank the search output. This process ranks the search output citations according to relevance value to the user initial information need. Finally, the top ranked citations are presented as the final retrieval results.

1.3 Roadmap

Chapter 2 covers the related work in medical information retrieval and describes background theories and methods used in this study. First, it introduces different information retrieval models. Then it explains the medical information retrieval using PubMed. PubMed Web Interface, PubMed Query Suggestions and Automatic Term Mapping (ATM), PubMed Advanced Search Feature, Medical Subject Headings (MeSH), PubMed Filters, PubMed Relevance Sort and PubMed Related Article (PMRA) feature are presented in the first section in Chapter 2. Then PIR concepts are introduced in Chapter 2. Next, it presents the different text classification methods namely; Naive Bayes, k- Nearest Neighbor (kNN) and Support Vector Machines (SVM). Document representation, feature selection and similarity measures are also presented in Chapter 2.

Chapter 3 presents the PARS methodology of finding the relevant citations for the user information need by extending the PMRA method in the PIR context. First it presents the user input for the PARS system. Then it presents the methodology of extending the PMRA method in the personalized retrieval context.

Chapter 4 discusses the classification based filtering approach in the PARS system. It presents the user input for the method, procedure of expanding the training set and Multi-Stage Filtering method of finding new unseen citations for the given information need.

Chapter 5 presents the evaluation procedure in this study. It discusses two approach to evaluate the proposed PARS. First it presents the procedure of evaluating PARS using the TREC (Text REtrieval Conference) 2005 Genomics track ad-hoc retrieval task data. Then it presents an empirical study with real PARS users.

Chapter 6 presents the results of this study. It first presents the results of both methods in PARS with the small TREC dataset. Then it presents the results of the real world users empirical study.

Chapter 7 presents the conclusions and future directions of this study.

CHAPTER 2

BACKGROUND

With the rapid advances in internet technology and developments, more and more electronic documents, images and videos have been produced and added to World Wide Web (WWW) and made available online. For example, currently Google web index contains over a trillion web pages [34]. This makes finding the relevant information from a large collection of information source, such as Web, a challenging task. Information Retrieval (IR) is relatively new field of Computer Science addressing this problem.

Formally, Information Retrieval (IR) can be defined as "finding materials from an unstructured nature that satisfies an information need within large collections" [35]. Therefore, the goal of IR can be expressed as finding the relevant resources to a user's information need and helps the user to complete a task. The web search engines are the most common and visible information retrieval systems. IR community has developed new set of algorithms and methods to build and enhance information retrieval from the Web and other domains. The Boolean Information Retrieval (BIR) model and the Vector Space Model were the earliest information retrieval model developed by the IR community.

2.1 Boolean Information Retrieval (BIR) Model

The interaction with an information retrieval systems starts by the user stating his or her information need. The most common way an information need is stated is through keywords. The keywords a user provides form a query, which is then submitted to the information retrieval system. IR system finds the relevant information to the user query from its defined domain. This domain could be a collection of web pages, scientific literature, audio files, video files, images, etc. Early IR systems focused on retrieving information to user query from domain of relative small size. The Boolean Information Retrieval (BIR) model [35] was developed to find documents or audio/video files having text or description

containing the exact matches of the query terms within its domain. In the early days of computing, when the search domain is small, the BIR model was adequate in finding the relevant information. However, with the rapid growth of information and domains, the BIR model is no longer a viable Information Retrieval tool in terms of accuracy and efficiency. As the size of the information search domain scales up, a more efficient methodology is needed. The vector space based information retrieval methods were developed to efficiently retrieve relevant information from large data source [35].

2.2 Vector Space Based Information Retrieval

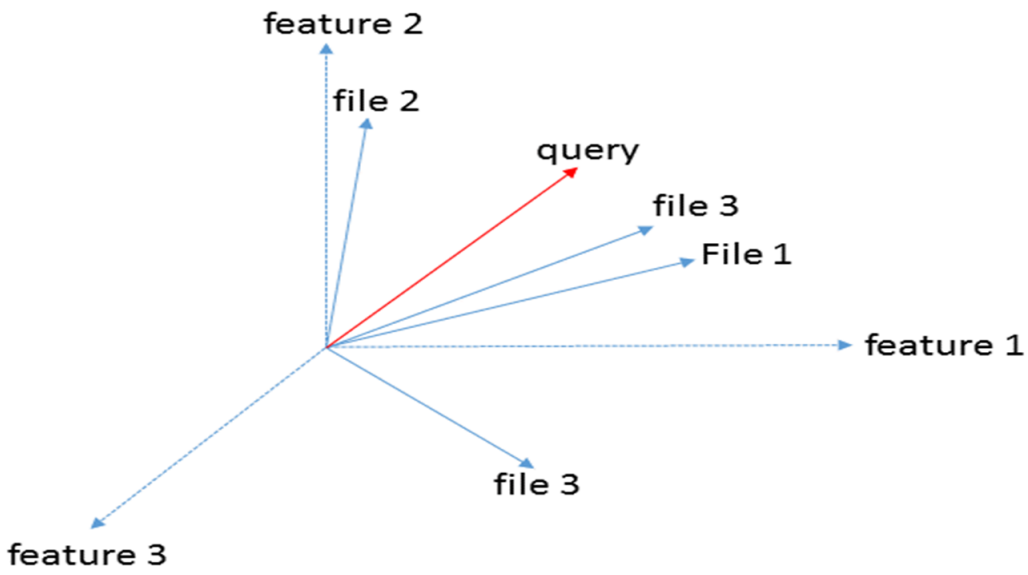

Figure 4: Three-dimensional feature space. The features could be terms or phrases extracted from a document set. All the documents (files) in the domain are represented in this feature space as vectors. The user query is also plotted in the same feature space.

In the vector space model, each data file is represented as a vector in a multidimensional vector space [35]. All the possible terms in the information collection form the dimensions of this space. Once a user submits a query to a vector space based IR system, it transformed the query into a multidimensional vector. Then, it plots the query into the same vector space as showing in Figure 4. The IR system finds the similar files to the query vector from the

multidimensional vector space using similarity measures such as the Cosine similarity [36], Jaccard Model [36] or Overlap model [36]. Euclidean distance [37], Manhattan distance [38] and Hamming distance [39] measures can be used inversely to calculate the similarity between the query vector and the file vectors. Finally, the vector space model produces a search output containing files that are the most similar to the user query. Figure 5 shows an abstract view of information retrieval using the vector space model. With introduction of vector space models, relevancy is introduced to the information retrieval. Because of this, search output of an IR system is a ranked list of relevant entities to the query. Therefore, IR systems based on vector space model are able to produce efficient search output from medium size datasets compared to those from the BIR model. Another group of computation models used by the IR systems is the probabilistic models [40, 41], inference networks [42, 43] and language models [44, 45]. However, IR systems developed based on these models were unable to show significant success in information retrieval from large datasets.

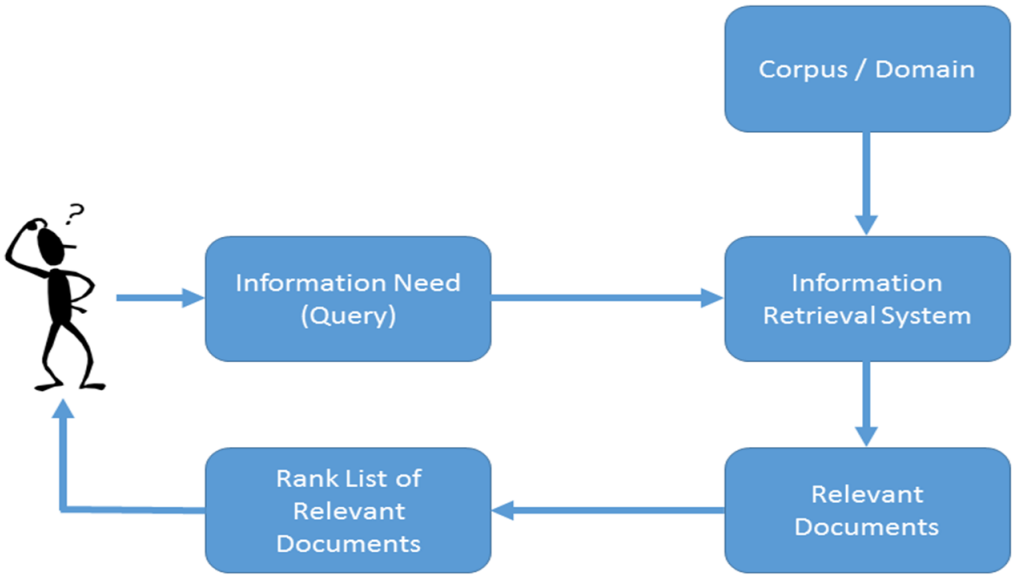

Figure 5: An overview of the information retrieval process based on vector space model. This model produces a ranked list of relevant documents or files (audio/video) as the search result for a query.

With the introduction of the PageRank algorithm [46] in the late 1990s, information retrieval has redefined and evolved dramatically. The PageRank algorithm aimed to improve the information retrieval efficiency from the Web. It captures the relevancy and importance of each document in the Web. It was the first algorithm used by the Google search engine [47, 48] and led to huge success in the information retrieval from the Web. The PageRank algorithm and its further development have enhanced the information retrieval from unstructured text.

The most common IR application is the web search engines, but there are many other applications in information retrieval for different domains. For example, E-mail search [49], searching the content in a personal computer [50, 51], searching information in a corporate knowledge bases [52], legal information retrieval [53] and medical information retrieval [54] are some of the applications of information retrieval. This thesis work mainly focuses on medical information retrieval.

2.3 Medical Information Retrieval

With the advancement of scientific fields, the scientific research community produced tremendous amount of research articles. Initially, scientific literature was kept in libraries and standard alone computer systems in universities, research organizations and national laboratories. Due to popularity and ease of access, many government and industry publishers opened their citation databases to general public via internet. Google Scholar [1], PubMed [2], CiteSeerX [4], Science.gov [55] and Astrophysics Data System (ADS) [3] are some of the popular scientific literature databases over the internet.

PubMed is the largest and most comprehensive citation database in the biomedicine field. It is developed and maintained by the National Center for Biotechnology Information (NCBI) at the National Library of Medicine (NLM) [5]. Primarily, PubMed provides free access to NLM's Medline citation database. Medline contains citations from many different

fields of medicine, health care systems, nursing, podiatry, veterinary, etc. In addition to Medline database, PubMed provides many other life science and out-of scope citations. Currently, PubMed contains more than 24 million citations [2]. Therefore, most of the practitioners and researchers in the field of medicine and biomedicine use PubMed as their primary information source.

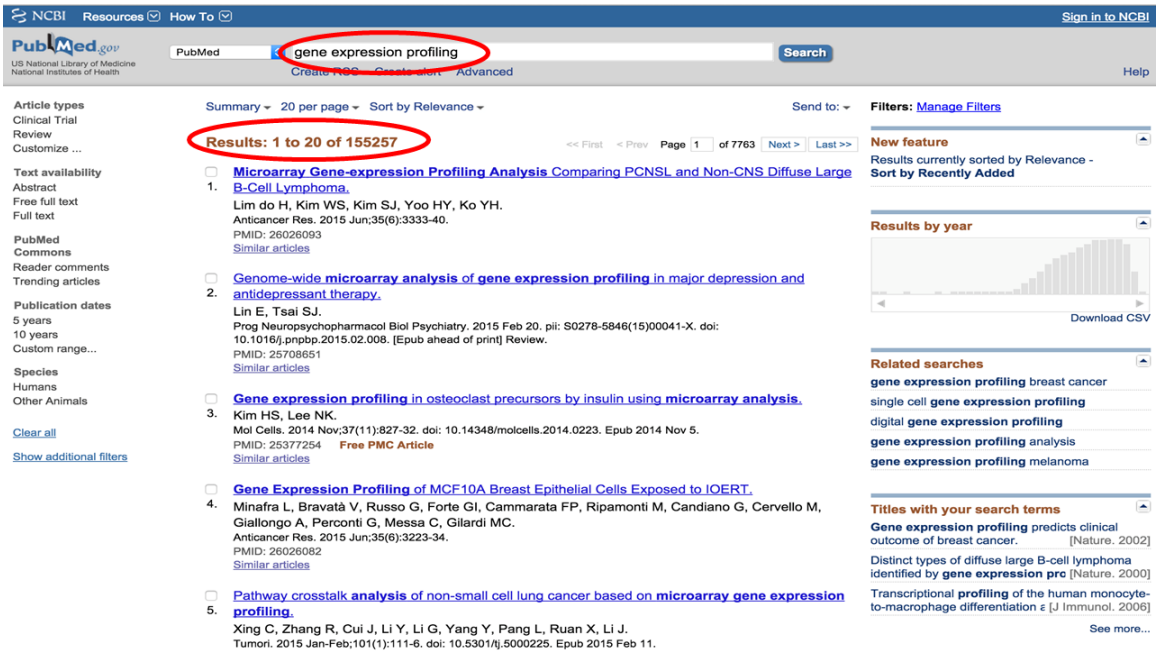

Figure 6: PubMed search interface. For the query gene expression profiling it returns 155,257 citations in the search results.

PubMed web interface is the primary way of accessing and searching literatures in the PubMed database. It provides a traditional keyword based search interface for its users as shown in Figure 6. Due to continued growth of the size of the medical citations, retrieving the relevant citations from PubMed for one's information need is becoming a challenge. The average search output size for a PubMed query is about 10,000 citations [56]. The PubMed search suffered from an information overload due to its volume. There is an increasing need to improve the information retrieval system for PubMed. In response to this, NCBI has introduced many features to enhance the information retrieval for PubMed. PubMed Advanced Search [8, 9], PubMed Filters [13], PubMed Auto Query Suggestions

[56], PubMed Automatic Term Mapping (ATM) [11] and Medical Subject Headings (MeSH) [12] are some of the features introduced by NCBI to perform a more efficient and accurate retrieval from PubMed.

2.3.1 PubMed Query Suggestions and Automatic Term Mapping (ATM)

PubMed Query Suggestions help users in formulating their information need in the PubMed interface [2]. PubMed automatically suggests queries to the users by using the previous popular queries in PubMed logs and the initial user query terms [56]. Then, users can modify their queries to better formulate their information needs. PubMed Automatic Term Mapping (ATM) [11] is another feature developed by the NCBI to automatically find the related terms to the user input keywords. It maps the Medical Subject Headings (MeSH) terms [12], journals and authors to the user input query terms to build a robust query without additional user information. This tool also helps to retrieve the highest relevant citations to the user by automatically enhancing the query. Both PubMed Query Suggestions and ATM are developed to assist the users to express their information need precisely. Both tools have achieved some success, but many users are still having difficulties in succinctly and accurately expressing their information need in PubMed, which leads to less than desirable search results.

2.3.2 PubMed Advanced Search

To further improve the quality of the search results of the original PubMed, NCBI developed an Advanced Search Builder [8, 9] to refine and limit the search output. The PubMed Advanced Search Builder allows the users to specify additional information about their information need along with the query terms. For example, a user can specify the interested journal, authors, and MeSH terms as shown in Figure 7. Additional information from the users help to reduce the search output size from the PubMed. Therefore, this

answers the information overload issue in PubMed up to certain extent. However, to compose an advanced search query in PubMed requires expert knowledge in the biomedicine field and PubMed search strategies. The expertise may not be present to a novice in the field, or to experts exploring a new research area in biomedicine.

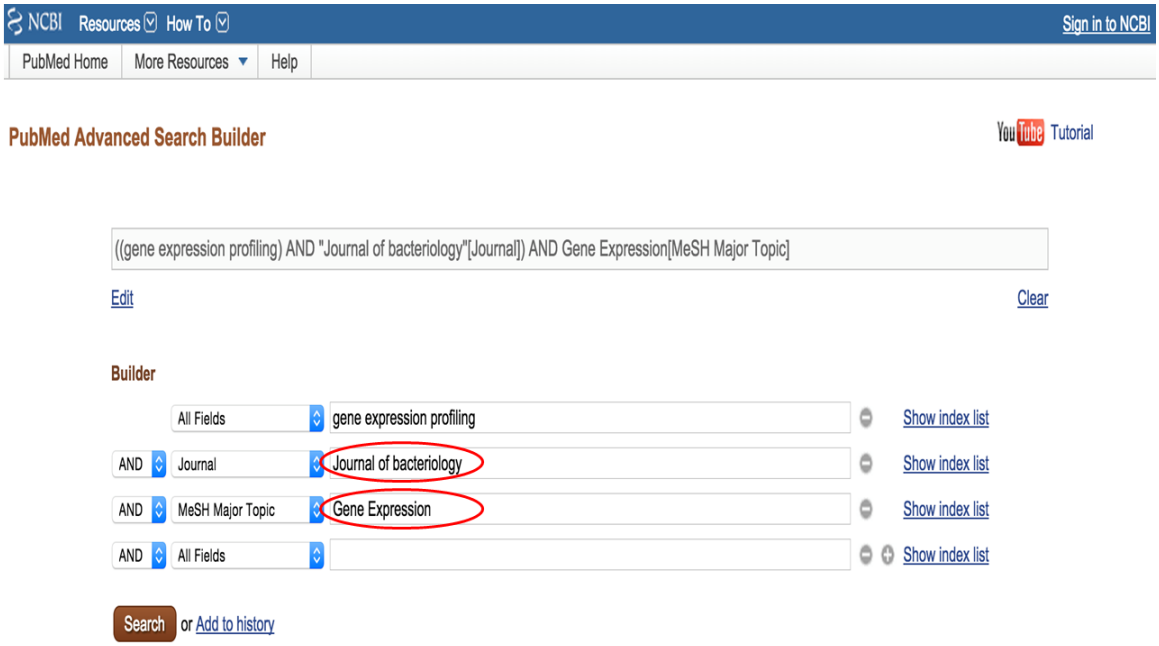

Figure 7: PubMed Advanced Search Builder. In this query, in addition to the keyword "gene expression profiling", it specified journal and MeSH Major topic fields to limit the search output.

2.3.3 Medical Subject Headings (MeSH)

Medical Subject Headings (MeSH) [12] terms are the important feature in the PubMed Advanced Search Builder. The Medical Subject Headings (MeSH) was introduced by the United States National Library of Medicine (NLM) to help in searching and indexing the Medline citations. Basically, it is a comprehensive controlled vocabulary thesaurus. MeSH vocabulary includes four different types of terms namely, MeSH Headings (MeSH Descriptors), MeSH subheadings (qualifiers), Supplementary Concept Records and Publication Characteristics [57]. MeSH descriptors are organized in a hierarchical structure called MeSH tree. In MeSH tree, MeSH descriptors are organized in 16 categories. In each category,

MeSH descriptors are arranged from most general to most specific using 12 hierarchical levels. Figure 8 shows an example view of the organization of MeSH terms in the tree. Subject experts in the NLM updated this MeSH tree annually.

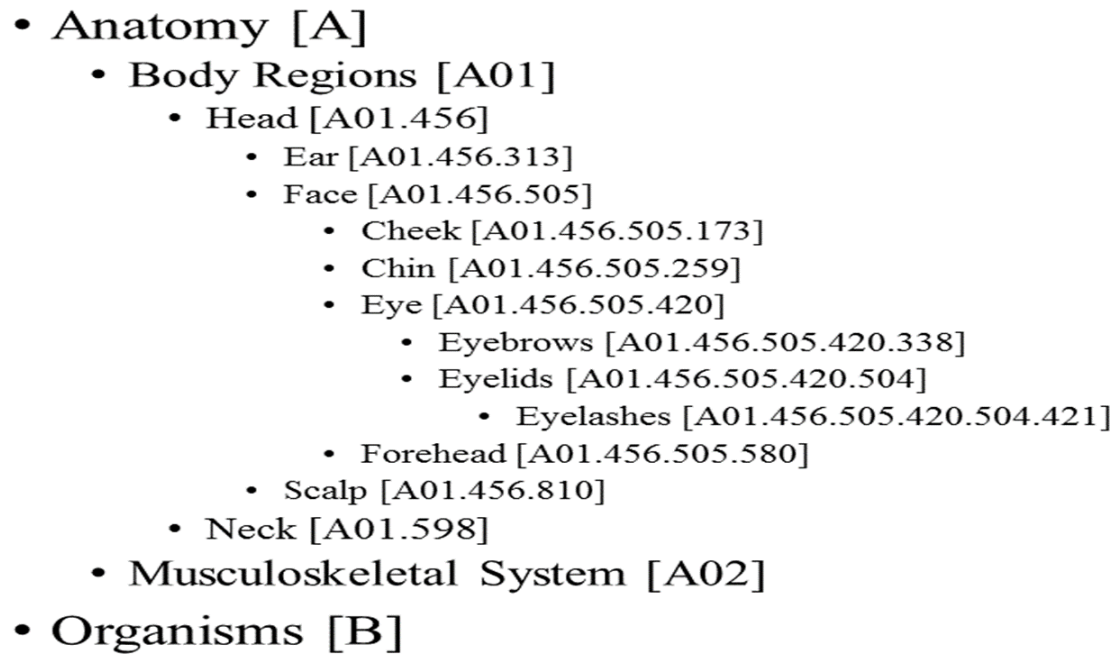

Figure 8: A slice of the MeSH tree. It shows how MeSH terms are arranged from most general to more specific. At the top, it has the most general Body Regions and Head MeSH terms. At the bottom of the tree more specific MeSH terms namely Eyelids and Eyelashes, are listed.

When a user expresses information need in the PubMed Advanced Search Builder [8], MeSH terms can be used to limit the search output in a systematic way. However, even the experienced researchers in the biomedicine field are reported having difficulties in using the MeSH terms effectively in their PubMed searches [80]. Because of this, NCBI has introduced PubMed filters [13] to the PubMed web interface as a different approach to refine and limit the original PubMed search output.

2.3.4 PubMed Filters

PubMed search results page includes the PubMed filters [13] sidebar on the left as shown in Figure 9. These filters can be used to narrow down the search results. A wide variety of

filters have been made available in the sidebar. For example, article type filters, publication date filters, species filters and text availability filters are some of the PubMed filters available in the PubMed filters sidebar. Users can select one or more filters from the sidebar to narrow down their search results. Because of this, users can find the relevant citations easily from the narrowed search output, however applying the right filter to refine the search output requires expert knowledge in the field of biomedicine. In addition to these features, there are two other important features provided in the PubMed interface, namely the PubMed Relevance Sort [58] and the PubMed Related Article (PMRA) [14].

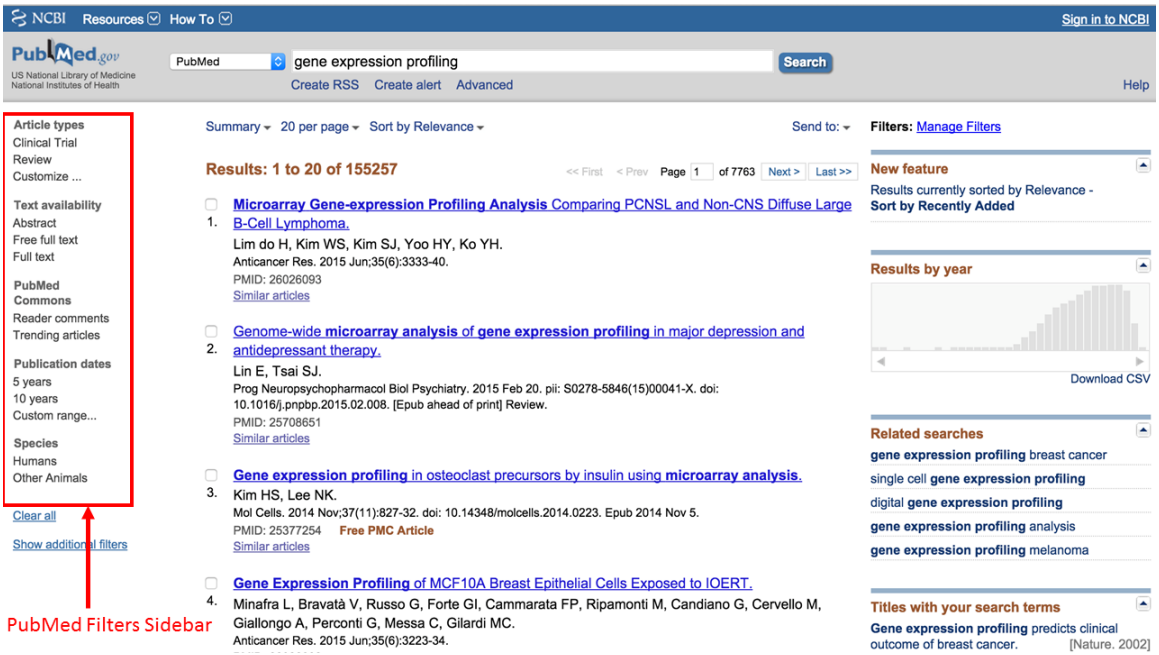

Figure 9: PubMed Filters sidebar on the PubMed search results page.

2.3.5 PubMed Relevance Sort

Originally, PubMed search output is sorted according to the reverse chronological order. Recently PubMed introduced the Relevance sort [58] as a new feature in the web interface. Relevance sort calculates the relevancy as a weight for each citation in the search output according to the common words between the query and the citation [58]. Then, it sorts the search output according to the relevancy value. This puts the most relevant citations on top

of the search output and is a more desirable search result.

2.3.6 PubMed Related Article (PMRA) Method

The purpose of the PMRA feature [14] is to find the articles that are similar to a chosen PubMed citation from the entire PubMed database. When a user selects a citation from the PubMed search results, PubMed also displays the citations having the highest PMRA similarity values, e.g., the closest matching, to the chosen citation in the similar article panel of PubMed web interface as shown in Figure 10. The list of the most similar citations forms the related citation list for the selected citation. The related citation list for each article in PubMed is pre-calculated, and pre-sorted according to the PMRA values [59]. The calculation and sorting of PMRA lists are done at the back-end and PubMed is updated periodically with the new PMRA scores. The calculation of PMRA documents for a given citation is performed with the following procedure.

Figure 10: PubMed Related Articles for the citation “Dissecting the polysaccharide-rich grape cell wall changes during winemaking using combined high-throughput and fractionation methods” are showing in the right hand panel of PubMed web interface.

Given that document d is deemed related to one's information need, PMRA computes the relatedness of document q in terms of the posterior probability $P(q|d)$, where q can be any document in PubMed. Assuming a document can be decomposed into a set of N mutually exclusive and exhaustive "topics" $\{s_1, s_2, ., s_N\}$. $P(q|d)$ is computed as shown in Equation 1.

$$P(q|d) = \sum_{j=1}^{N} P(q|s_j)P(s_j|d) \tag{1}$$

Expanding $P(s_j|d)$ using the Bayes theorem, we obtained Equation 2.

$$P(q|d) = \frac{\sum_{j=1}^{N} P(q|s_j)P(d|s_j)P(s_j)}{\sum_{j=1}^{N} P(d|s_j)P(s_j)} \tag{2}$$

For a user selected document d, the denominator of Equation 2 remains constant for any document q. Therefore, for comparative purposes, the denominator of Equation 2 can be ignored and the following criteria can be used to rank documents based on their relatedness or similarity.

$$P(q|d) \propto \sum_{j=1}^{N} P(q|s_j)P(d|s_j)P(s_j) \tag{3}$$

Here $P(q|s_j)$ is the probability that the user finds an interest in document q, given an interest in topic s_j. Similarly, $P(d|s_j)$ is the probability that the user finds an interest in document d, given an interest in topic s_j. $P(s_j)$ is the prior probability of the topic s_j i.e., the fraction of all documents that discuss the topic s_j. Therefore, relevance of a document q to the given document d can be computed by summing up the product of $P(q|s_j)$, $P(d|s_j)$ and $P(s_j)$ across all the topics [14].

In order to estimate $P(q|s_j)$, $P(d|s_j)$ and $P(s_j)$, PMRA introduced a concept called *eliteness* [14]. *Eliteness* explains whether a given document d is about a particular topic s_j or not. The original PMRA method assumes that each word in the PubMed citation (title, abstract and MeSH term list) represents a topic (s_j). Moreover, each word (term) in the PubMed citation represents an idea or concept in the document. A term t_i is *elite*

for document d if it represents the topic s_j. Otherwise, term t_i is *non-elite* for document d. Equation 4 can be derived using the *eliteness* concept and Bayes theorem [14]. Let E represent the *eliteness* of a term in document d, and $\bar{E}$ represents the *non-eliteness* of a term in document d. The probability a term is *elite* in a document is conditioned on the number of times, k, that term appears in the document:

$$P(E|k) = \frac{P(k|E)P(E)}{P(k|E)P(E) + P(k|\bar{E})P(\bar{E})} = \left(1 + \frac{P(k|\bar{E})P(\bar{E})}{P(k|E)P(E)}\right)^{-1}. \tag{4}$$

$P(k|E)$ and $P(k|\bar{E})$ are calculated using the Poisson distributions as shown in Equations 5 and 6.

$$P(k|E) = \frac{\lambda^k e^{-\lambda}}{k!} \tag{5}$$

$$P(k|\bar{E}) = \frac{\mu^k e^{-\mu}}{k!} \tag{6}$$

where λ is the mean of the Poisson distribution of the *elite* case for the given term, and μ is the mean of the Poisson distribution of the *non-elite* case for the given term. First, substitute Equation 5 and 6 values into Equation 4. Then applying document length normalization and algebraic manipulations to Equation 4, we derived Equation 7.

$$P(E|k) = \left(1 + \frac{\mu^k e^{-\mu} P(\bar{E})}{\lambda^k e^{-\lambda} P(E)}\right)^{-1} = \left[1 + \eta \left(\frac{\mu}{\lambda}\right)^k e^{-(\mu-\lambda)l}\right]^{-1} \tag{7}$$

where l is the length of the document and $\eta = P(\bar{E})/P(E)$.

Then, we combine the concept of *eliteness* with the relatedness concept of two documents. $P(E|k)$ is used to estimate $P(q|s_j)$ and $P(d|s_j)$ in the $P(q|d)$ model. To efficiently calculate the similarity values, $P(s_j)$ is estimated using the inverse document frequency of a term (topic) t_j, idf_{t_j}. Inverse document frequency (idf) calculates the important of a term within a document collection. Then, the following weighting function and the similarity function

are derived to calculate the similarity of the two documents.

$$P(q|d) \propto sim(q,d) = \sum_{j=1}^{N} [P(E|k)]_{t_j,q} \cdot [P(E|k)]_{t_j,d} \cdot idf_{t_j} \tag{8}$$

$$sim(q,d) = \sum_{j=1}^{N} \left[1+\eta\left(\frac{\mu}{\lambda}\right)^{k} e^{-(\mu-\lambda)l}\right]^{-1}_{t_j,q} \cdot \sqrt{idf_{t_j}} \cdot \sum_{j=1}^{N} \left[1+\eta\left(\frac{\mu}{\lambda}\right)^{k} e^{-(\mu-\lambda)l}\right]^{-1}_{t_j,d} \cdot \sqrt{idf_{t_j}} \tag{9}$$

$$w_t = \left[1+\eta\left(\frac{\mu}{\lambda}\right)^{k} e^{-(\mu-\lambda)l}\right]^{-1} \cdot \sqrt{idf_t} \tag{10}$$

$$sim(q,d) = \sum_{j=1}^{N} w_{t_j,q} \cdot w_{t_j,d} \tag{11}$$

where w_t calculates the term weight for a given document. Similarity between the two documents is computed with an inner product of the term weights as in Equation 11.

Parameter Estimation in PMRA

PMRA similarity calculation requires that a number of parameters, λ, μ, η be estimated. A simplifying assumption has been made for the *elite* and *non-elite* Poisson distributions: half of the terms in the document are *elite* and the other half of the terms are *non-elite*. This assumption leads to Equation 12, a model similar to the maximum entropy models used in Natural Language Processing [14, 60].

$$\eta\left(\frac{\mu}{\lambda}\right) = \frac{P(\bar{E})\mu}{P(E)\lambda} = 1 \tag{12}$$

The weighting scheme expressed in Equation 10 can then be re-written as:

$$w_t = \left[1+\left(\frac{\mu}{\lambda}\right)^{k-1} e^{-(\mu-\lambda)l}\right]^{-1} \cdot \sqrt{idf_t} \tag{13}$$

This way, PMRA reduces the number of parameters to be estimated from three to two. Medical Subject Heading (MeSH) [12] information in Medline was used to estimate λ and μ. MeSH descriptors to each PubMed indexed citation are assigned manually by experts

in the field of biomedicine. Therefore, terms in the MeSH descriptors can be considered as *elite* terms for the citations. The terms in the citation that do not appear in the MeSH descriptors are considered *non-elite* terms for the citation. The average appearance of a given *elite* term (λ) or a given *non-elite* term (μ) can be calculated based on a collection of PubMed citations.

To give developers and researchers full advantage of information stored at PubMed, NCBI has opened up their database and programming utilities to the public by allowing them to develop application specific and enhanced information retrieval system for PubMed. As a result, many academic and industry groups have developed IR systems to enhance different aspects of PubMed; for example, ranking the search output [15, 16], clustering the search output [26, 29], finding relevant materials using semantic based method [18, 19] and visualize the retrieval results [61].

The main goals of these enhanced IR systems is to capture a user's information need uniquely and broadly and return a small set of truly relevant citations as the search output from PubMed. This type of IR systems is more broadly referred to as Personalized Information Retrieval (PIR) systems.

2.4 Personalized Information Retrieval

PIR systems consist of two main tasks:

- The first task is to capture and understand a user's information need accurately and efficiently.
- The second task is to find the citations closest to the user information need and present these citations in a meaningful manner.

For the traditional IR systems, user information needs are provided as user queries consisting of keyword terms. However, it is often difficult to fully specify and capture the user information need precisely using only the query terms. Additional information about

the user query intention is required for the PIR systems to deliver more personalized results. There are two main approaches to gather individual user interest for the PIR systems. The first approach is to capture the user interest explicitly [22, 15, 16]. The second approach collects information about the user query intention implicitly; for example, in terms of the click-through links in the search history [23, 24, 25]. Either explicit or implicit information provides a deeper understanding about the user information need to the PIR systems. The second stage of the PIR system carefully filters and sorts the relevant information to the user information need using more sophisticated text mining algorithms. Text classification [15, 16], text clustering [26, 27], relevance ranking [22, 28] and semantic based methods [18, 19, 20] are some of the text mining techniques used in the PIR systems. This study focuses on building a PIR system based on text classification approach. The next subsection discusses the text classification based PIR systems.

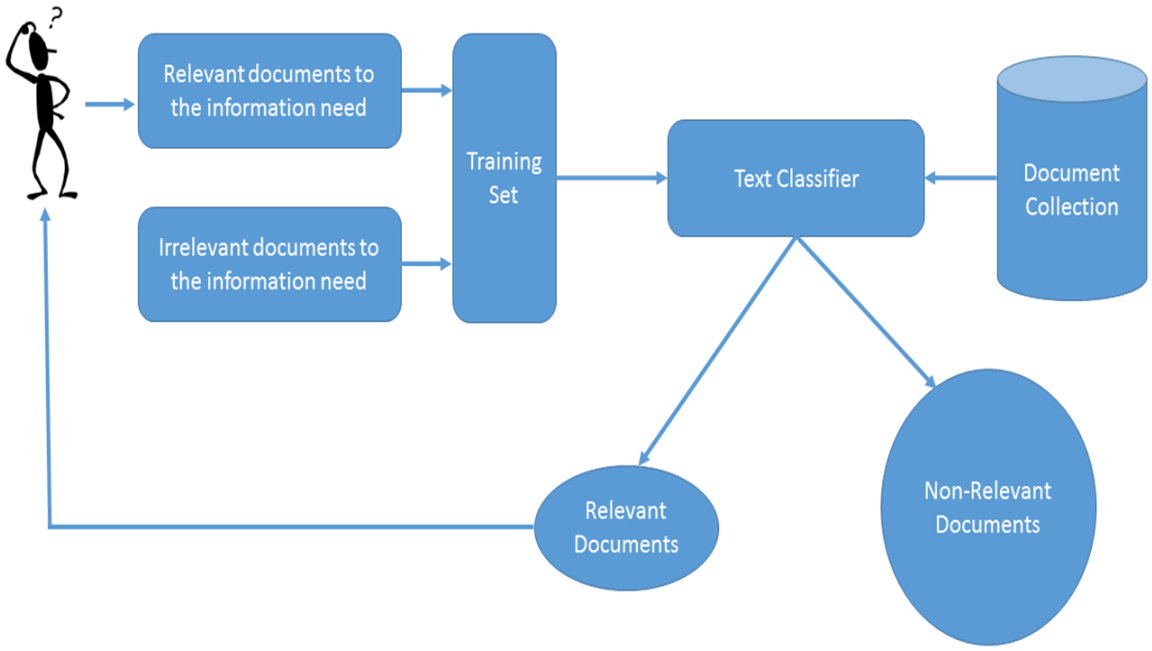

Figure 11: Overview of text classification based Personalized Information Retrieval (PIR) systems.

2.4.1 Information Retrieval as Classification

Data required to train the text classifiers in a PIR system consists of a set of relevant citations and non-relevant citations. The relevant set consists of citations deemed relevant by the user to his or her information need. The non-relevant set consists of irrelevant citations to the user information need. In many cases, the relevant set is provided by the user and the irrelevant set is randomly selected from the entire dataset by the PIR system. Figure 11 presents an abstract view of a text classification based PIR systems.

Both relevant and non-relevant citation sets are used to build the training set for the text classifiers in the PIR system. Then, PIR system trains the text classifiers in the system using the training set of data. The trained text classifiers are used to classify the documents in the collection into two classes: documents relevant to the user information need and documents non-relevant to the user information need. Finally, the PIR system presents a partial or full list of relevant citations as the search result. There are many different text classification algorithms available to employ for this type of PIR system [15, 16, 35]. The following section presents an overview of the different methodologies developed in text classification.

2.5 Text Classification

Over the past two decades, many different text classification approaches have been developed including the Naive Bayes (NB) method [35], the Support Vector Machines (SVM) [35], the Rocchio method [35], the regression based models [62], the k-Nearest Neighbour (kNN) method [35], and the Neural Networks [62]. Text classification consists of two major phases, the learning or training phase and the classification phase. The objective of the learning phase is to derive the classification model from the training examples provided. Once the training (learning) phase has finished, the classification phase starts where the learned classifier is used to determine the class label for new, unseen documents. The following sections briefly explain the theory behind the Naive Bayes classification, the kNN

text classification and the Support Vector Machine (SVM) approaches, which have been used as base classifiers in the PIR system developed in this thesis work.

2.5.1 Naive Bayes (NB) Text Classifier

The Naive Bayes classification method is fast, robust and easy to implement. It is based on the *posterior* probability model derived using the Bayes theorem [35]. Given a document d, its probability of belonging to a class c is $P(c|d)$. The goal of the Naive Bayes classification is to find the optimal class for a given document, i.e., the class that gives the maximum posterior probability, $\hat{P}(c|d)$ [35]. This is expressed as:

$$C_{map} = argmax_{c \in C} \hat{P}(c|d) \tag{14}$$

where, C_{map} is the class with the maximum posterior probability, $c \in \{c_1, c_2, c_3, ..., c_n\} = C$ is the set of class labels and d is the given document. Applying the Bayes theorem, Equation 14 can be re-written as:

$$C_{map} = argmax_{c \in C} \frac{\hat{P}(d|c)\hat{P}(c)}{P(d)} \tag{15}$$

The term $P(d)$ has the same value for all the classes and it does not affect the *argmax* decision. Therefore, $P(d)$ can be dropped from the above Equation 15. Then, assuming $\{t_1, t_2, t_3, ..., t_{nd}\}$ is the set of terms in document d, and nd is the total number of terms in the document, Equation 15 can be written as:

$$C_{map} = argmax_{c \in C} \hat{P}(t_1, t_2, t_3, ..., t_{nd}|c)\hat{P}(c) \tag{16}$$

To efficiently compute $\hat{P}(t_1, t_2, t_3, ..., t_{nd}|c)$, the Naive Bayes Conditional Independence assumption is made: the terms in the documents are conditionally independent from each

other given the class information. Using this assumption Equation 16 becomes:

$$C_{map} = argmax_{c \in C} \hat{P}(t_1|c) \cdot \hat{P}(t_2|c) \cdots \hat{P}(t_{nd}|c) \hat{P}(c) \tag{17}$$

$$C_{map} = argmax_{c \in C} \hat{P}(c) \prod_{i=1}^{nd} \hat{P}(t_i|c) \tag{18}$$

During the training stage, the probabilities, $\hat{P}(c)$ and $\hat{P}(t_i|c)$, are estimated from the training data.

$$\hat{P}(c) = \frac{N_c}{N} \tag{19}$$

where, N_c is the number of documents in class c and N is the total number of documents in the training set.

To calculate $\hat{P}(t_i|c)$, the Naive Bayes Positional Independence assumption is made where the conditional probability of a term is independent of the position of the term in the document. Therefore, $\hat{P}(t_i|c)$ can be computed as:

$$\hat{P}(t_i|c) = \frac{T_{ct_i}}{N_{ct}} \tag{20}$$

where, T_{ct_i} is the total number of occurrence of term t_i in class c including multiple occurrences, N_{ct} is the total number of terms in class c including multiple occurrences.

At the classification stage, given terms $\{t_1, t_2, t_3, ..., t_{nd}\}$ for a document d, Equation 18 is used to compute the *posterior* probability of the document for each possible class, $c \in C$. The class assigned to the document is the one having the highest *posterior* probability.

2.5.2 k-Nearest Neighbor (k-NN) Text Classifier

k-Nearest Neighbor (kNN) [35] algorithm is also known as instance-based learning or lazy learning. The kNN algorithm does not have an explicit training step. During classification, it examines the class labels of k nearest neighbors that are the most similar to the test

object, and classifies the test object with the majority label from its k neighbors. A similarity measure is used to find the k nearest neighbors from the training set. Cosine Similarity [36] is used here to find the nearest neighbors. In this study, kNN training set consists of equal number of positive (relevant) and negative (non-relevant) training examples. We need to predefine a value for k in the kNN text classifier. In order to break ties in majority vote, an odd integer for k such as 1,3,5,7... is often used. The best value of k depends on the dataset.

2.5.3 Support Vector Machine (SVM) Text Classifier

Support Vector Machines is another very popular and powerful machine learning algorithm for text classification. It is originally a binary classification system developed by Vapnik and his colleagues [63]. The goal of the Support Vector Machine is to find N dimensional hyperplane that optimally separates data into regions. Let us consider the example in Figure 12. It has two categories of data and the data can be represented in a two dimensional space. A line can be easily drawn to separate the data into two non-overlapping regions. In fact, an infinite number of lines can be drawn to separate those data, as shown in Figure 12. Support Vector Machines method gives a formal explanation to find the optimal hyperplane to separate data. According to Figure 13, Support Vector Machines find the optimal hyperplane which maximize the margin between two data regions. The data points lie on the margins are called *support vectors*.

According to Figure 14, the optimal separating hyperplane H can be defined as:

$$w^T \cdot x + b = 0 \tag{21}$$

and marginal (canonical) hyperplanes H_1 and H_2 as follows;

$$w^T \cdot x + b = 1 \tag{22}$$

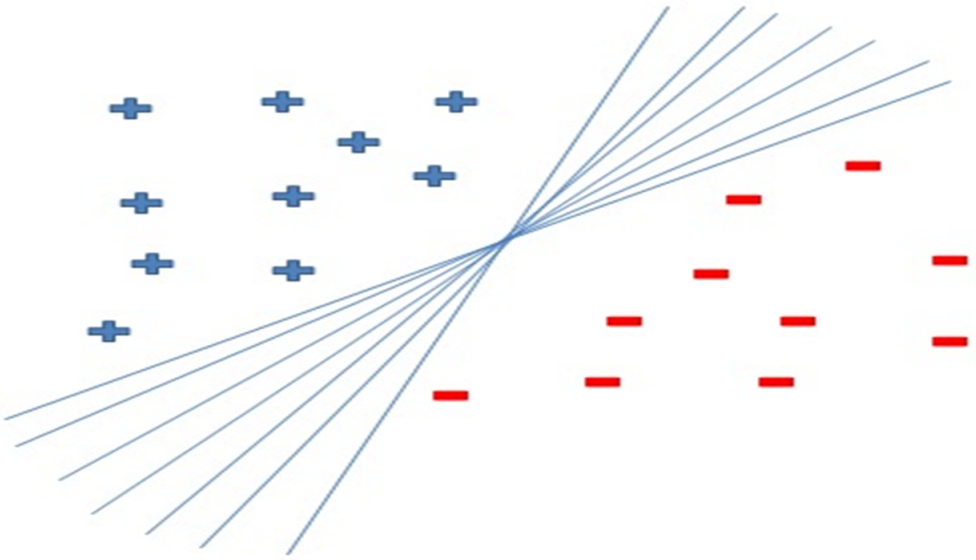

Figure 12: Two categories of data in a two dimensional space. An Infinite number of lines can be drawn to separate + and − in the given 2D space.

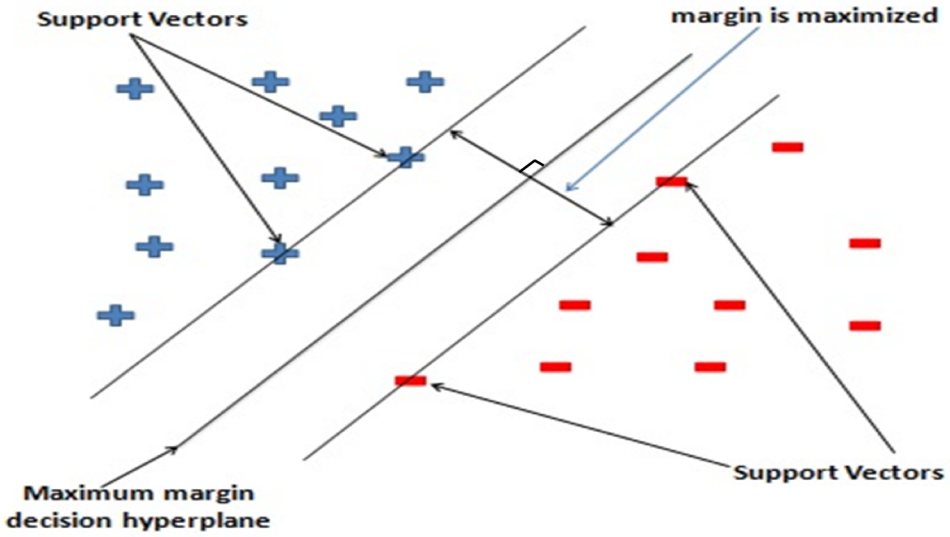

Figure 13: Graphical representation of Support Vector Machines classification for the two class problem.

$$w^T \cdot x + b = -1 \tag{23}$$

where, w is a weight coefficient vector perpendicular to the hyperplanes and b is a bias term.

Once w and b are computed, we can use the following classification criteria to classify data:

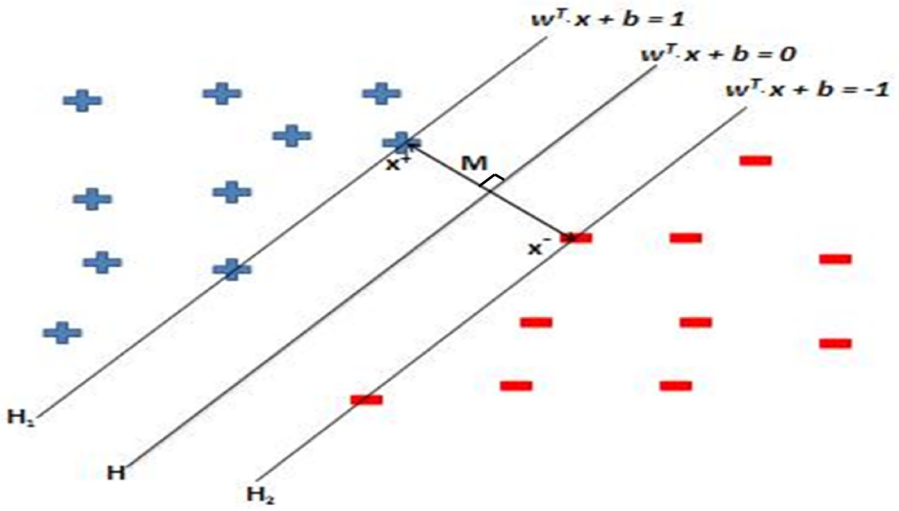

Figure 14: SVM binary classification. H represents the optimal separating hyperplane and H_1 and H_2 represents marginal hyperplanes.

$$\textit{If } w^T \cdot x + b \geq 1 \textit{ then classify as +1}$$

$$\textit{Else } w^T \cdot x + b \leq -1 \textit{ then classify as -1}$$

Take x^+ and x^- the points which are on the H_1 and H_2 hyperplanes respectively, Equations 22 and 23 can be written as:

$$w^T \cdot x^+ + b = 1 \tag{24}$$

$$w^T \cdot x^- + b = -1 \tag{25}$$

The x^+ and x^- points are called as support vectors for the optimal separating hyperplane. Equation 24 and 25 lead to:

$$w^T \cdot (x^+ - x^-) = 2 \tag{26}$$

Since, w is a vector which is perpendicular to the $w^T \cdot x + b = 0$ plane, we can write $(x^+ - x^-)$ vector as follows:

$$(x^+ - x^-) = \lambda w \tag{27}$$

Combining Equations 26 and 27, λ can be computed as:

$$\lambda = \frac{2}{w^T \cdot w} \tag{28}$$

and the margin M can be computed as:

$$M = |x^+ - x^-| = \lambda \cdot |w| = \frac{2}{w^T \cdot w}\sqrt{w^T \cdot w} \tag{29}$$

where, $|w| = \sqrt{w^T \cdot w}$

$$M = \frac{2}{\sqrt{w^T \cdot w}} \tag{30}$$

Maximizing M can be achieved by minimizing $\frac{\sqrt{w^T \cdot w}}{2}$. The minimization function $g(w)$ is given as:

$$g(w) = \frac{1}{2}(w^T \cdot w) \tag{31}$$

subject to the constraints: $y_i[w^T \cdot x_i + b] \geq 1$

where y_i is the class label for x_i. Now, the problem is reduced to optimization of a quadratic function subject to linear constraints. This type of problem can be solved by constructing a dual problem where α_i is associated with each constraint $y_i[w^T \cdot x_i + b] \geq 1$ in the primal problem [35].

Find $\alpha_1, \alpha_2, \alpha_3, ..., \alpha_N$ such that

$$\sum \alpha_i - \frac{1}{2}\sum_i \sum_j \alpha_i \alpha_j y_i y_j (x_i^T \cdot x_j) \tag{32}$$

is maximized and $\sum_i \alpha_i y_i = 0$, $\alpha_i \geq 0$ for all $1 \leq i \leq N$.

Then the final solution can be expressed as:

$$w = \sum \alpha_i y_i x_i \tag{33}$$

$b = y_k - w^T \cdot x_k$ for any x_k such that $\alpha_k \neq 0$.

Finally, the classification function for new unseen data 'z' can be written as:

$$D(z) = sign(\sum_i \alpha_i y_i x_i^T z + b) \tag{34}$$

The SVM method works well for linearly separable data. Figure 15 shows an example where data is not linearly separable. SVM uses kernel trick for data which is not linearly separable. The main task of a kernel function is to transform the data from the input space into a higher dimensional feature space where a hyperplane can be used to do the separation.

Figure 16 gives the visual explanation of the task of a kernel function. The kernel trick

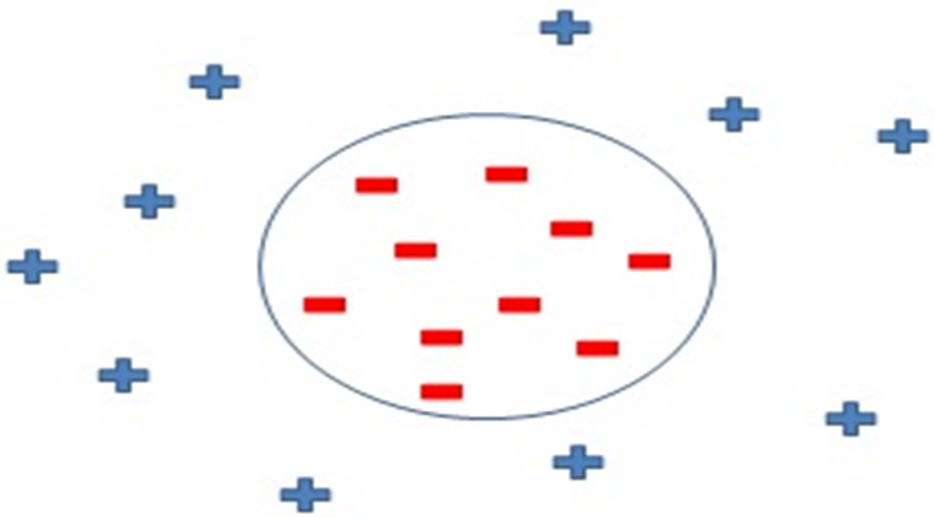

Figure 15: Data is not linearly separable in the two dimensional input space. Therefore, SVM classifier is not directly applicable to this type of data.

of the SVM replaces the $(x_i^T \cdot x_j)$ dot product in Equation 32 using some well known kernel functions. With g as the transformation function, $(x_i^T \cdot x_j)$ can be written as:

$$x_i^T \cdot x_j \rightarrow g(x_i^T) \cdot g(x_j) \tag{35}$$

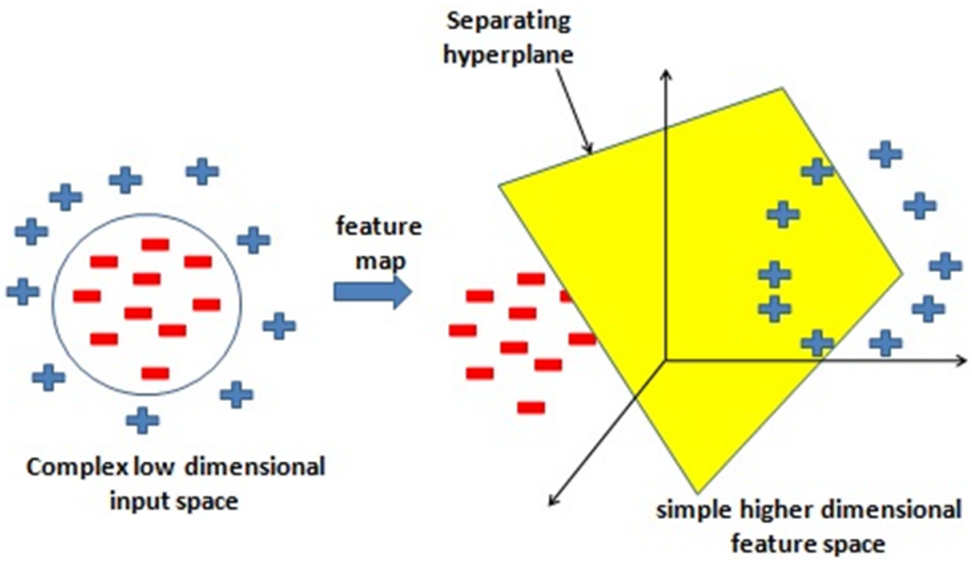

Figure 16: Process of the kernel function. It transforms low dimensional input data into higher dimensional feature space.

Then, a kernel function $K(x_i, x_j)$ can be defined as:

$$g(x_i^T) \cdot g(x_j) = K(x_i, x_j) \tag{36}$$

Here are some common kernel functions:

$K(x_i, x_j) = (x_i \cdot x_j + 1)^d$ - Polynomial,

$K(x_i, x_j) = exp(-\gamma \|x_i - x_j\|^2)$ - Radial Basis,

$K(x_i, x_j) = tanh(k(x_i \cdot x_j) + c)$ - Sigmoid.

2.6 Document Preprocessing

Different ways of representing the documents in the text classification system may lead to significantly different performance in any text classifier. Therefore, it is important to carefully select a scheme to represent the documents in a PIR system. A vector based representation is the most common and standard scheme of representing document in an information retrieval system. With this scheme, each document is represented as a multi-

dimensional vector. The following section presents the steps used to converting a text document into a vector representation.

Most of the documents on the web or in scientific citation databases are in the form of HTML or XML. Therefore, Data preprocessing steps were used to obtain a meaningful wordlist from the HTML/XML documents. First, the document title and the content can be extracted from the HTML/XML file. Then, the document title and the content are tokenized into a list of words. From this list of words, stop words, words containing only numbers, and words containing only a single character are removed. After that, stemming is applied to normalize the words. This process converts a HTML/XML document into a list of normalized words. Next, HTML/XML documents need to transform into a vector-based representation using this list of words [81].

2.6.1 Document Representation

To further prepare the document into a representation suitable for classification, the list of words obtained in each document is converted into a feature vector of dimension K, where K is the size of the dictionary built based on words appeared in the training set. The set of unique words in the dictionary is referred to as terms. Feature vector values are represented using the Term Frequency (TF) or Term Frequency Inverse Document Frequency (TF-IDF) [35] techniques. Term frequency (TF) method counts the number of appearances of dictionary term t in the document word list of document d as $tf_{t,d}$, and stores it in the corresponding dimension of the feature vector. In the TFIDF method, feature vector values were represented using the TF-IDF weights [35]. TF-IDF weight is computed for each term, t, in each document, d, as:

$$w_{t,d} = (1 + log(tf_{t,d})) \times log\left(\frac{N}{df_t}\right) \tag{37}$$

where, N is the total number of documents in the training set, and df_t is the frequency of documents where term t appears.

Each document is transformed into a K dimensional vector. K is the total number of unique words from the training set documents. Usually K is a large number for even a small training set. For N documents, a $N \times K$ matrix may be used to store all the vector representations of the N documents. This matrix is often a very sparse matrix, since only a small percentage of words from the dictionary appear in each document. One way to reduce the sparsity of the matrix is to reduce the dimension size by only retaining the informative words in the dictionary.

2.7 Feature Selection

Feature selection is the process of finding the most discriminative set of features from the whole feature set. Mutual Information [35], Chi-square method [35] and Correlation coefficient [64] are common feature selection measures. Best first [35], Greedy forward selection [65], Genetic algorithms [66] and simulated annealing [67] have been developed for text mining problems. However, applying a sophisticated feature selection procedure in a PIR system would lead to longer response time. Also, most of the time, a training set in a PIR system is a small set of citations. Applying an advanced feature selection procedure to a small set is not efficient. Therefore, in this study we proposed a simple feature selection procedure based on document frequency values in the training set.

2.7.1 Feature Selection Procedure

The following list of steps are used to select features in this study:

1. The dictionary is created by collecting all the unique terms in the training set documents.

2. Calculate the document frequency (df) values for each term in the dictionary. That is,

for each term in the dictionary, count how many documents in the training set contain that term.

3. Select all terms appearing in two or more documents from the dictionary and stored as a Feature Dictionary (FD). That is, selecting df ≥ 2 terms from the original dictionary.

4. Represent the document-to-vector conversion scheme in the PIR system using the reduced feature vector.

Using this procedure, the size of the new Feature Dictionary (FD) is smaller than the original dictionary built based on all the terms appearing in training set documents. The sparse representation of document vectors can be eliminated. This leads to a significant improvement in the text classification performance and the execution time of the classification.

2.8 Similarity Measures

As a final step, the PIR system ranks the search output according to the similarity of each citation to the user information need. Therefore, a similarity measure is required to rank the search output in the PIR system. The following section describes the similarity measure used in this study.

Similarity measures are used to find how close two instances are in the data space. The closer two instances are to each other, the larger the similarity value between them. Similarity measures are inversely related to the distance measures. Similarity measures can be divided into two main categories: probability based and vector based similarity measures. Fidelity similarity [68], Hellinger affinity [68] and bm25 [69, 70] are some of probability based similarity measures. Person correlation coefficient [68], Jaccard coefficient [68] and cosine similarity [36] are some of the distance based similarity measures.

This study uses cosine similarity algorithm to find the similarities between the documents. Cosine similarity is heavily used in the information retrieval and text mining community.

A previous study showed that the cosine similarity and the overlap model out-performed many other similarity measures in the TREC dataset [71]. The following section provides a theoretical explanation about the cosine similarity measure.

2.8.1. Cosine Similarity

Cosine Similarity provides a simple and effective method to compute the similarity between two vectors. It measures the angle between two vectors as the similarity value. Let's assume A and B are n dimensional vectors in the data space. The dot product of the A and B vectors can be written as in equation 38.

$$A \cdot B = \|A\|\|B\|cos(\theta) \tag{38}$$

Then, angle between A and B vector (similarity) is expressed in the equation 39.

$$similarity = cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2}\sqrt{\sum_{i=1}^{n} (B_i)^2}} \tag{39}$$

where, $A = \{A_1, A_2, A_3, ..., A_n\}$ and $B = \{B_1, B_2, B_3, ..., B_n\}$.

CHAPTER 3

EXTENDING PMRA FOR PERSONALIZED RETRIEVAL

To produce reasonably accurate results, most of the current PIR systems for PubMed require that, at the beginning of an IR process, a user needs to provide 100 or more citations that have been screened to be relevant to the information need [15, 16]. This set of citations play a central role in these PIR systems where they form the basis of the training data to learn the classification models. However, in real applications, this is not a requirement that can be easily satisfied. In fact, it is not easy for a user to provide just 50 citations that are directly relevant to his information need. To eliminate the need for this unrealistic requirement in a PIR system, in this chapter, we propose our PARS system for PubMed that is capable of learning (training) from a much smaller set of citations provided by user.

Two different approaches to develop PARS for PubMed are discussed:

- The first approach adapts from the PubMed Related Article (PMRA) feature [14] developed in PubMed, and
- The second approach implements a text classification based filtering method.

Chapter 3 discusses the PARS using the extended PMRA feature and Chapter 4 discusses the PARS with a text classification based filtering method.

PMRA [14] is a useful tool in PubMed. Given a PubMed citation, PMRA finds all its relevant citations from the PubMed database. Due to the amount of computation involved, the PMRA lists for individual citations in PubMed are calculated and updated on the back-end, at the time when each new citation is indexed and added to the database [59]. We would like to know:

1. "Is it possible to extend the existing PMRA feature in the personalized information retrieval context?"

2. "What is the best way to extend PMRA feature in the personalized information retrieval setting?"

We answer these questions by developing a PARS system with an extended PMRA feature [14]. Figure 17 gives an overview of the system. It consists of the input module, and PMRA tree crawling module, and the EPMRA method. Details of how PMRA was extended to work with the PARS system and the methodologies developed in this PARS system will be discussed in detail.

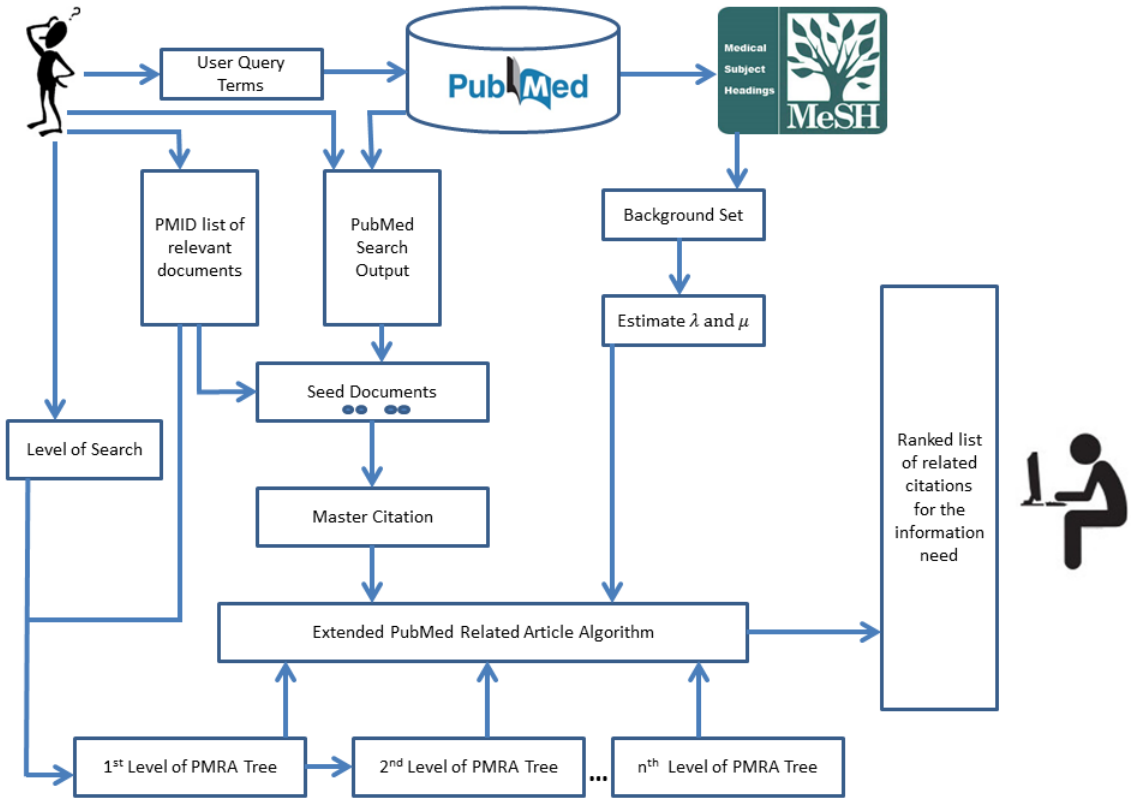

Figure 17: The overall system architecture of the PARS system with EPMRA.

3.1 User Input

There are different ways for a user to express his information need in an IR system. The most common way of expressing an information need is using a query in the form of multiple keywords or terms [1, 2]. Other ways of expressing information needs include using a text paragraph [22], using multiple documents [15, 16], and using multimedia input

(image or voice input) [72, 73]. In the proposed PARS system, user information need is expressed by providing a small set of citations, for example 5-10 PubMed citations that discuss a specific brain cancer treatment procedure. These citations are considered highly relevant to one's information need. These citations provide a more detailed representation of the user's information need than that expressed using keywords. It also provides more information to the PARS system.

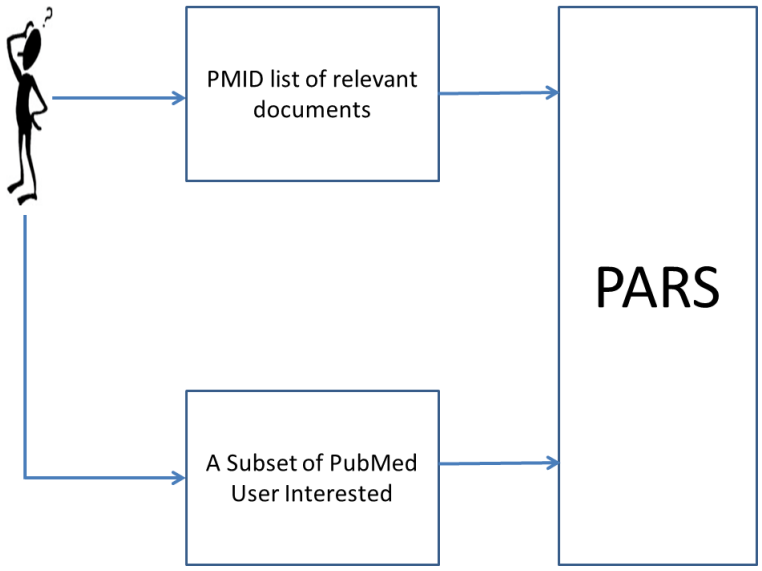

Figure 18: User input to the PARS consists of a set of relevant PubMed IDs and a subset of PubMed for which the user is interested in finding citations. This subset is defined using the existing PMRA feature in PubMed.

Figure 18 shows the two types of input required from the users in PARS. In addition to the citations, a user is required to select a subset of the PubMed database from which to find new and unseen relevant citations. Two different ways have been provided for a user to enter the relevant PMRA citations in the PARS system. Figure 19 shows the web interface of the first method of composing an information need in PARS. The PubMed ID (PMID) of the relevant citations can be entered through the text box in PARS web interface as shown in Figure 19. PMID is a unique identifier for individual citation in the PubMed database. The relevant citations entered by the user are referred to as the *seed* documents in PARS.

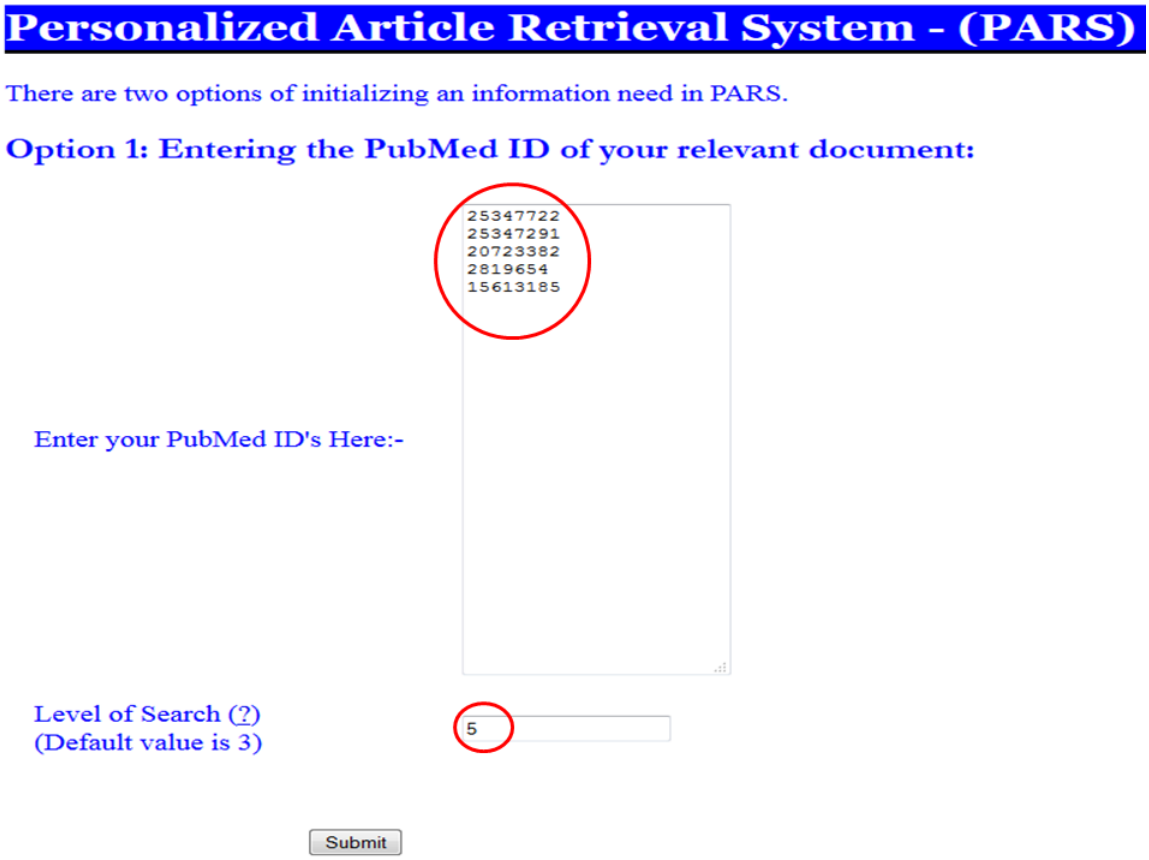

Figure 19: The user interface of the proposed PARS system. This web page shows the first method of entering citations in PARS. It will take a list of PubMed citation IDs and the Level of Search (LOS) value. The default LOS value is set to 3.

Then, the user needs to specify the Level of Search (LOS) value in PARS. LOS defines how deep in the PubMed database the user chooses to crawl. A tree crawling module is developed to systematically search through all the citations in PubMed.

3.2 Tree Crawling Module

The original PubMed search output is sorted according to the chronological order of published date [74]. However, PARS ranks the search output according to the relevance of the citations to the user's information need. In order to do that efficiently, PARS is designed to crawl the PubMed database using a tree structure based on the PMRA related citation lists as shown in Figure 20. To build the PMRA tree, first, the system retrieves the PMRA list for

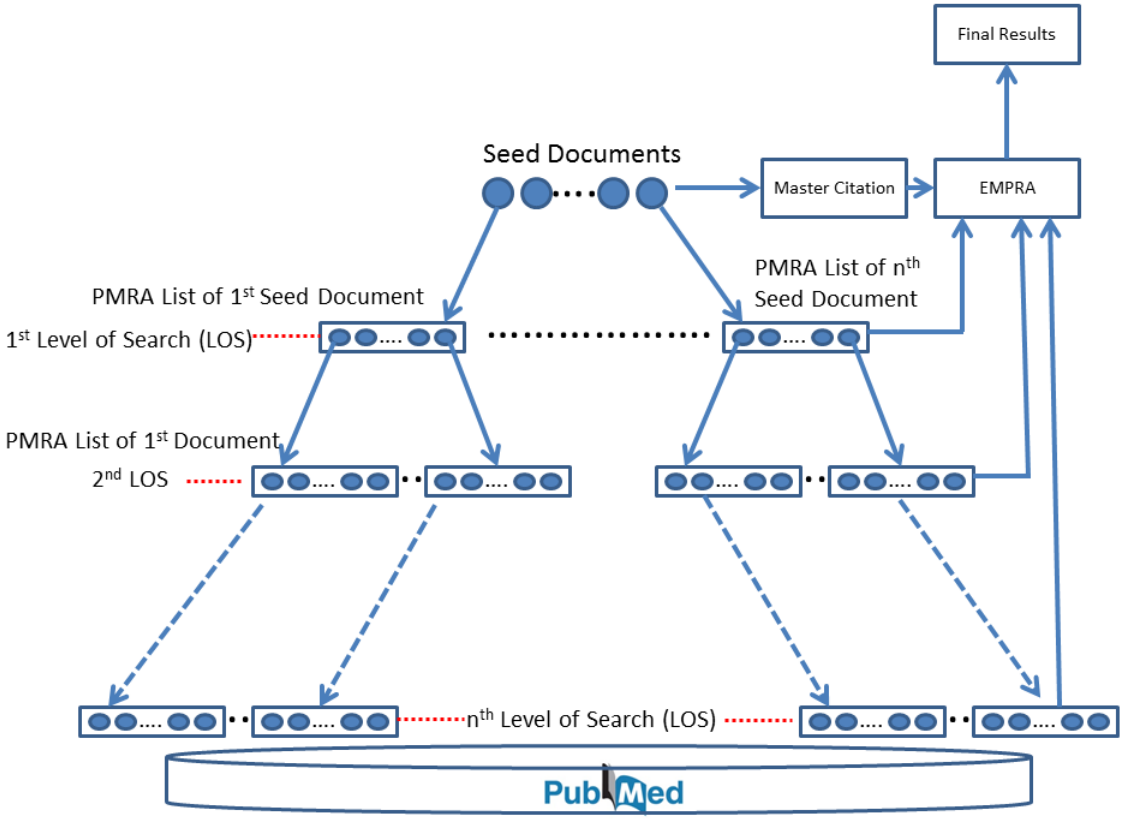

Figure 20: The PMRA Tree structure is used for crawling the PubMed database in PARS. First level includes citations from the PMRA lists of the *seed* documents. The second level consists of citations of all the PMRA lists of citations used in the first level crawling. The n^{th} level covers the entire PubMed database.

each *seed* citation. The first level of the PMRA tree consists of all the PMRA documents for the user *seed* citations. If a user needs to retrieve only a small set of citations from the system, he or she can specify a small LOS value for the information need. For example, when the LOS value is 1, PARS finds the relevant documents from the entire *seed* document PMRA lists. When the LOS value is set to 2, PARS finds the new relevant citations from PMRA list of *seed* documents and PMRA list of each citation in the first level. If a user want to find citations from the entire PubMed databse, then user can specify the wildcard value "*n*". Then, PARS finds the relevant citations from the entire PubMed database. The default value for the LOS is set to 3. This LOS value indirectly specifies a subset of the PubMed

database the user is interested in. After specifying the LOS value in the web interface, the user can submit his or her information need to the PARS system.

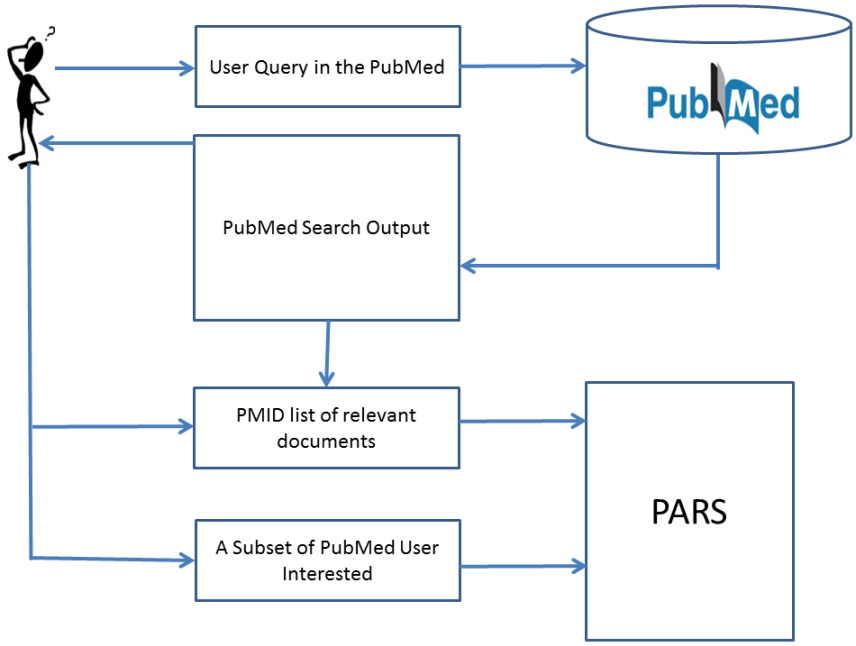

Figure 21: The procedure of expressing the information need in PARS using the existing PubMed Search interface.

In the second method of expressing the information need, the existing PubMed search feature is adapted to allow the users to express their information need. Figure 21 illustrates the steps in this method. This method is mostly suitable for beginners or the researchers who start a new branch in their research. Because, in these situations, the user of PARS does not have a pre-defined set of relevant articles for their information need. Therefore, a relevant article set may be found using the standard PubMed search based on keywords.

With this method, a user expresses his information need using multiple keywords in the search box as shown in Figure 22. The standard PubMed search is performed and a list of citations is output to the user. The user can examine the citations in the search output and form a *seed* set by selecting the most relevant citations from the search output. In the next step, the user can limit the document set from which to search for the relevant articles by specifying the Level of Search (LOS) value as explained in the previous section.

Figure 22: The second option of composing an information need in PARS. First, user does a traditional key-word search using original PubMed search. Then, from the search output user can choose some of the citations that are relevant to his or her study.

3.3 Extending the PMRA Similarity Measure

With the user input entered using the methods discussed above, PARS proceeds to find the unseen relevant documents from the subset of PubMed specified by the user using the Extended PMRA (EPMRA) method. The following sections explain the EPMRA method and related methodologies on finding more relevant citations to a given information need.

The original PMRA was developed to find the relevant citations for a given citation. Currently, PMRA method is not directly applicable to finding the relevant citations for multiple user selected citations.

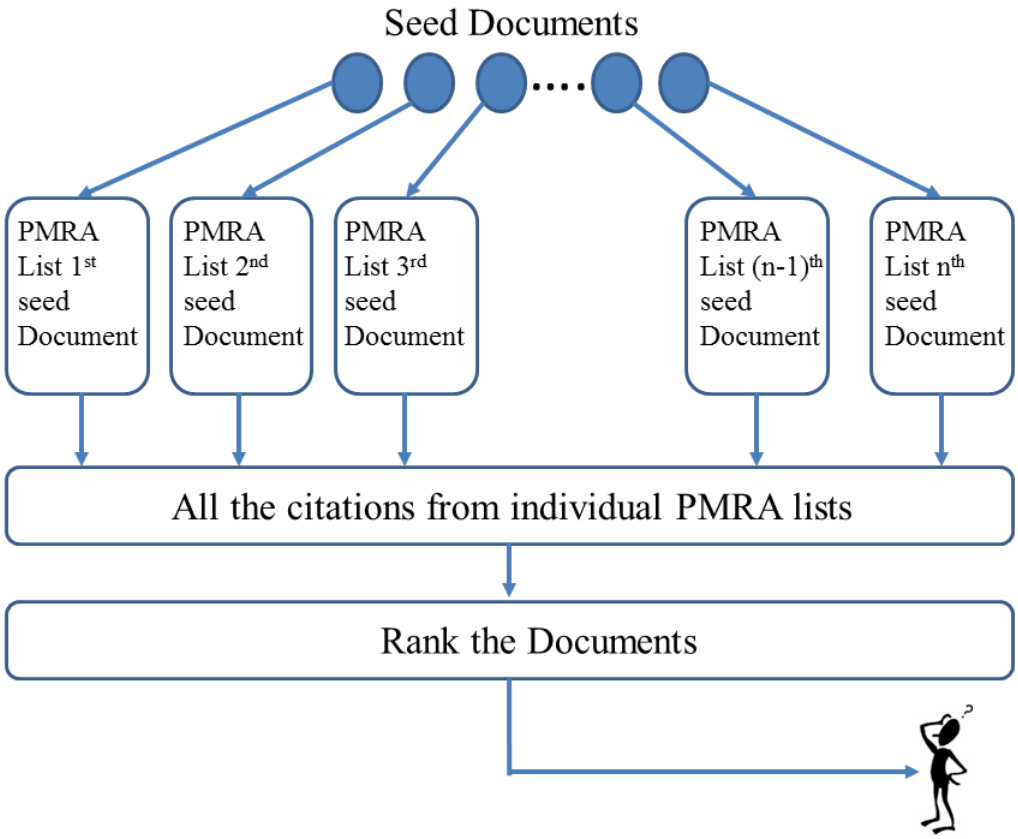

Figure 23: Building PARS using only the existing PMRA lists. All the PubMed related citations from the *seed* documents are collected to form a potentially relevant citation set. Then, the citations are ranked according to the relatedness to the information need. Finally, the citations most relevant to the information need are presented to the user. we refer this method as the Basic Method.

The straightforward way of extending PMRA for multiple citations is to combine the PubMed Related Article (PMRA) lists obtained from the individual *seed* citations, and sort all the derived articles according to their PMRA similarity values. We refer to this method as the Basic method. Figure 23 shows the procedure of building the PIR system based on this Basic methodology. The PMRA related article list for individual citations is pre-calculated in PubMed. Therefore, the Basic method can be completed in a very short time. However, this method is not good at capturing the overall user concept or idea of the information need expressed through multiple citations, i.e., the individual citations are

considered independently. Also, the citations that are not indexed in PubMed do not have a PMRA list. Retrieval results using this method are not accurate.

The second approach is to combine multiple citations into a single citation and to find the relevant citations to this newly-formed citation. This method is slower than the first approach because the newly-formed citation is not present in the PubMed database, therefore no pre-computed list is available. But, this approach gives a better representation of the particular user information need by taking into account information present in all the user-defined citations. This is the approach taken by the proposed PARS system.

The combined citation is referred to as the *Master Citation*. There are multiple ways of combining the set of *seed* articles into a single citation (*Master Citation*):

- The first method, the **All-Inclusive** method, simply combines the terms from all the *seed* citations;
- The second method, the **Intersection** method, forms the new citation by only including terms that simultaneously appeared in every single *seed* citation, i.e., intersection of all *seed* citations;
- The third method, the **At-Least-Two** method, forms the new citation by including terms appearing in at least two *seed* citations.

In this study we experimentally compare the effectiveness of these four methods: the Basic method, the All-Inclusive method, the Intersection method, and the At-Least-Two method.

Figure 24 gives an overview of the methodology PARS uses to compute the list of related citations to the *Master Citation* using the EPMRA method. In this approach, the PMRA similarity value between the *Master Citation* and each citation in the database needs to be estimated. In order to calculate the PMRA similarity values, the parameter values, i.e., the λ and μ values in Equations 5 and 6 in Chapter 2, in the original PMRA method need to

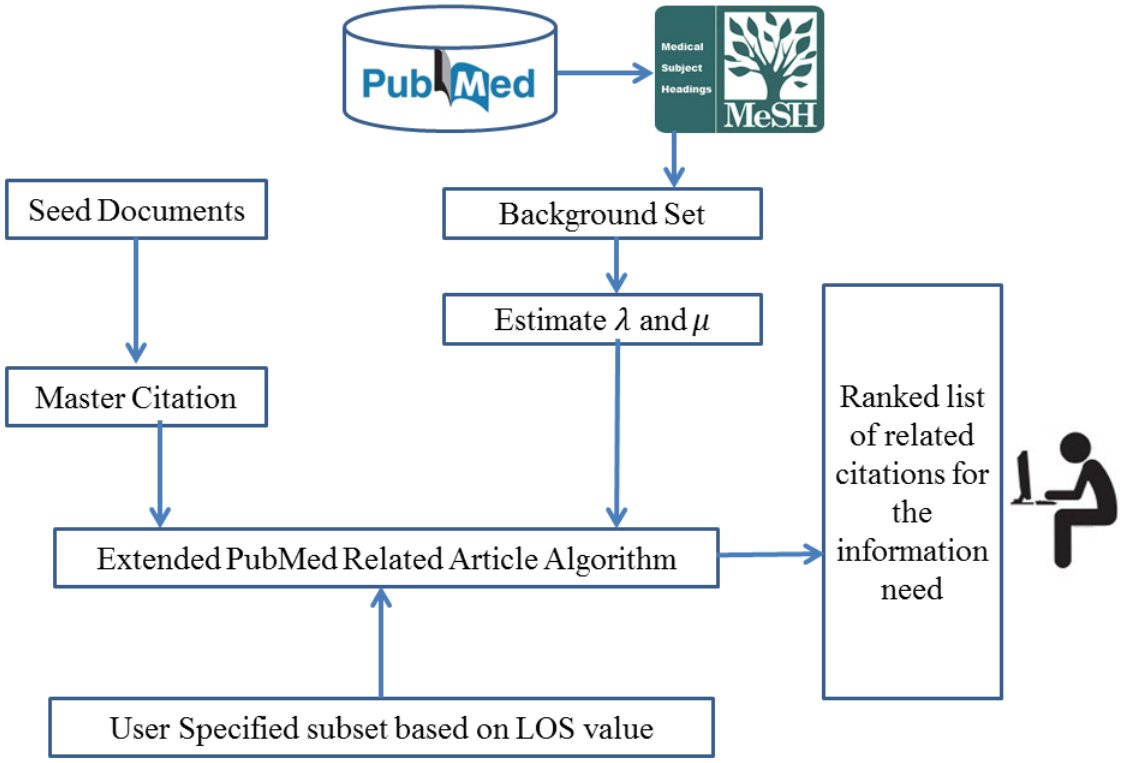

Figure 24: Methodology of developing PARS based on the PMRA feature. In this method, *seed* documents are combined in to a single *Master citation.*

be recalculated. The following section discusses the process of estimating the λ and μ parameters in the PARS system.

3.3.1 Estimating the Parameters in the EPMRA Method

In PMRA calculation, the λ and μ parameters represent the *elite* and *non-elite* frequencies of a given term in all the citations in the dataset. To estimate these two parameters for the dataset, one may choose to use the entire PubMed database, a random sample from the PubMed, or a biased sample from the PubMed (e.g., a subset provided by user, results of a PubMed search) as the random sample population. Due to computation concerns, a scaled down, stratified random sample [75] based on the Medical Subject Headings (MeSH) [12] is used to estimate the parameters. This random sample is referred to as the *Background Set*

in the PARS system. The next section discusses the procedure used to build the *Background Set*.

3.4 Building the Background Set

The Medical Subject Headings (MeSH) [12] was introduced by the United States National Library of Medicine (NLM) to help with searching and indexing the Medline citations. It is a comprehensive controlled vocabulary thesaurus. MeSH vocabulary includes four different types of terms namely, MeSH Headings (MeSH Descriptors), MeSH subheadings (qualifiers), Supplementary Concept Records, and Publication Characteristics. MeSH descriptors are organized in a hierarchical structure called the MeSH tree. In the MeSH tree, MeSH descriptors are organized in 16 categories. In each category, MeSH descriptors are arrayed from the most general to the most specific using 12 hierarchical levels. Subject experts in the NLM update this MeSH tree annually. In 2015, there are 27,455 MeSH descriptors in the MeSH tree [12]. The *Background Set* in PARS is built based on the 2014 MeSH tree descriptors.

To ensure that the λ and μ parameters estimated from the *Background Set* are a close to those estimated from the entire set of citations from PubMed, it is important that the *Background Set* constructed mimics the entire PubMed data. This is achieved by ensuring that the *Background Set* constructed covers all the concepts in the MeSH tree. First, for each MeSH descriptor in the MeSH tree, apply the Advanced PubMed search to retrieve the PubMed articles. Among these retrieved articles, only those having their MeSH Major topic matching the given MeSH descriptor will be kept for further consideration. Next, 20 unique citations are randomly selected and added to the *Background Set*. This process is repeated for the 27,455 MeSH descriptors and the final *Background Set* contains a total of 549,100 articles.

3.5 Finding and Displaying the Results

To find the relevant citations, PARS examines each citation in the subset of PubMed specified by the user. PARS calculates the relatedness between each citation and the *Master Citation* using the EPMRA method. After the EPMRA calculation is completed, all the citations are ranked according to the EPMRA relatedness value, and the top 100 most relevant citations are presented to the user. The top citations are presented with information including the citation title, PMID, authors and the EPMRA relatedness value. The EPMRA relatedness value can be used as a "goodness" indicator for the search output. A significant drop of the relatedness value indicates that the remaining citations in the output are much less related to the information need. If the relatedness value is low for all the output citations, then the user may want to modify the initial *seed* documents or the LOS value and perform a new search in order to get a better search output. If the user wishes to see more than 100 citations from the search output, he can request the next 100 citations from the system. As explained above, PARS with EPMRA is more efficient and effective in producing search output for the user information need than the other competitive systems.

CHAPTER 4

PARS WITH CLASSIFICATION BASED FILTERING

Chapter III presents the PARS system developed using an extended PMRA method. This chapter discusses the PARS system developed based on the text classification methodology. Text classification approaches automatically classify documents into two or more categories based on its content. Text classification have been used in applications such as spam filtering, news or web document classification, email routing and sentiment analysis [82, 83]. Text classification has recently been used in the field of information retrieval.

Text classification and PIR share many commonalities. For two class text classification, a training data set consists of positive and negative examples is required to train the text classifiers. In PIR, a set of relevant and non-relevant documents is needed to represent the user's information need. Once a text classifier is trained, during the classification step, it is used to classify the test documents into positive and negative classes. Similarly, a PIR system identifies the relevant citations by eliminating the non-relevant citations in the data.

Information retrieval tools developed based on text classification have been used in some citation database. For example both MedlineRanker [15] and MScanner [16] find relevant citations from PubMed using text classification. There are two main drawbacks in these systems. First, both systems require a user to input a large set (more than 100) of relevant citations as the input. Secondly, these systems produced large search output for the information need. The proposed PARS system addresses both issues using text classification based multi-stage filtering methodology. The Mutli-Stage Filtering (MSF) methodology only requires a user to provide a small set of relevant documents as the input. Also, MSF method returns a small set of highly relevant citations to the user's information need. Figure 25 gives an overview of the PARS system built using the MSF method.

The rest of this chapter discusses the details of the modules developed in this PARS system for user input, training data construction, the MSF procedure for eliminating false

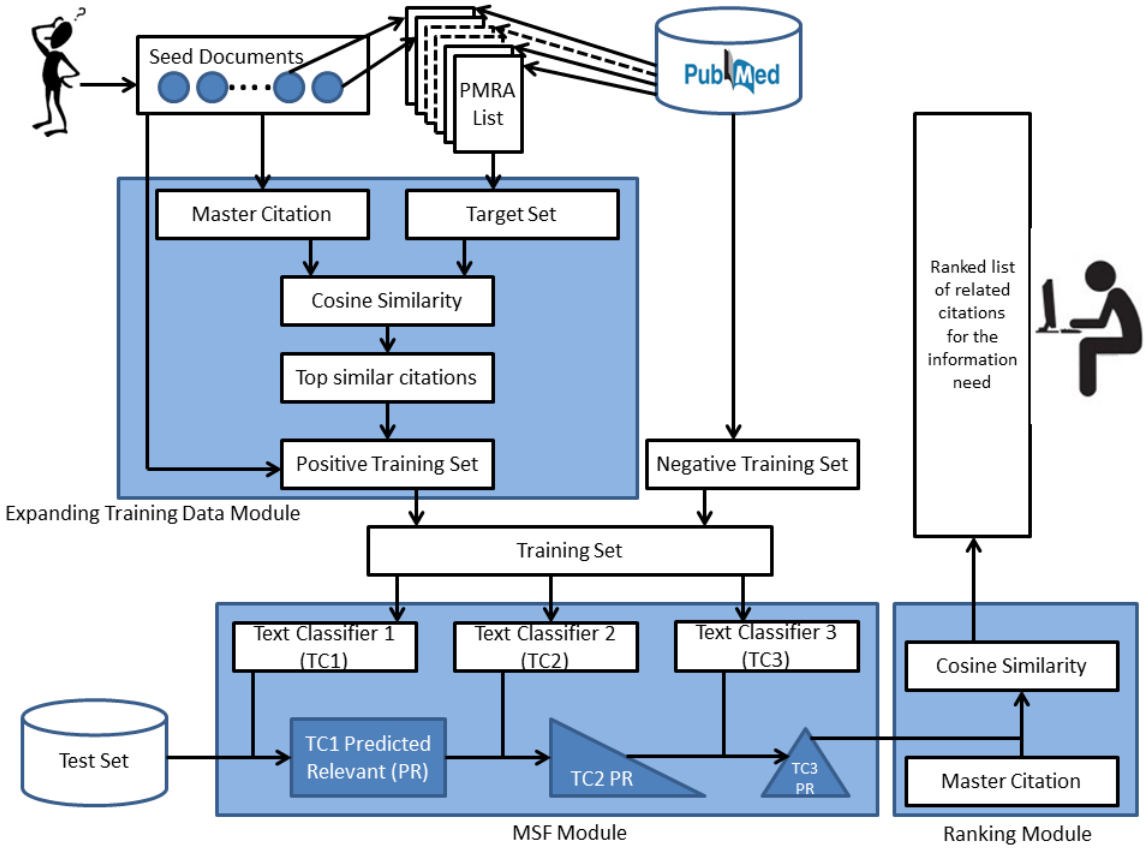

Figure 25: The overall approach of the PARS Multi-State Filtering (MSF) method.

positive citations, and the ranking procedure used for the search output. For information retrieval systems, the initial user input is very critical to its success.

4.1 User Input for the MSF Method

The user input for the MSF based PARS system is similar to that of the EPMRA based PARS system. A user is required to provide a small set of relevant citations to represent the user's information need. These initial relevant documents are referred to as the positive *seed* documents in the MSF approach. There are two ways of forming the positive *seed* document set in the MSF approach. A user can directly enter the relevant PubMed IDs, as shown in Figure 19, or he can choose a subset of the citations returned from a PubMed search when given keywords, as shown in Figure 22.

In addition, a user may choose to specify a set of PubMed citations that are considered not relevant to the information need. These citations will be used by the PARS system in forming the negative training examples during the training stage. If these non-relevant citations are not properly selected, they may negatively affect the quality of the final output of the PARS system. If the user is an expert in the subject area, then selecting non-relevant citations is recommended. Otherwise, a user may choose not to select any non-relevant citation, in which case, the non-relevant citations will be selected by the system. The non-relevant citations may be entered by specifying their PMIDs or by selecting from the set of non-relevant citations provided by the PubMed search output. These non-relevant citations are referred to as the negative *seeds* in the MSF method.

Finally, a user needs to specify the test set for the PARS. The test set can be the entire PubMed database or a subset of the PubMed database specified by the user. For example, if the user is interested in only retrieving the relevant citations published in the last five years, then the test set includes only the citations published in those five years. Typically, both the positive and the negative *seed* sets include 5 to 20 citations. It is a known fact that it is difficult to train text classifier with good accuracy based on a small data set. Therefore, an important step of building the text classifier based PARS system is to identify a proper technique to expand the training set based on the small set of citations (user seeds) provided by the user. Using an expanded training set, text classifier can be applied efficiently to find relevant citations from PubMed. The following section presents the procedure of expanding the training set based on the user seeds.

4.2 Expanding the Training Set Size

To increase the size of the training data based on seed citations supplied by a user, a similarity-based approach has been developed to find the citations that are the most similar to the *seed* citations from the entire database. However, given the size of the PubMed database,

to perform a real time similarity computation between each of the *seed* citation and every citation in the database is not practical. Therefore, PARS uses the PMRA feature [14] to build a smaller *Target Set*, based on which the training data set is to be expanded. For each citation in PubMed, its PMRA list is pre-calculated. To build the *Target Set*, first the PMRA list for each user seed is retrieved, then these PMRA lists are combined into a single citation list. This unique citation list is referred to as the Target set.

The idea is to form a Master Citation that combines the information needs presented in all the user seed citations. Merging all the seed citations into a single citation forms the Master Citation. Then, compute the cosine similarity values [36] between this Master Citation and each of the citations included in the Target Set. The citations in the Target Set having the highest similarity values are the citations closest to the user seed citations, thus the best candidates for expanding the test data. Together, the newly added citations and the user seeds form the positive (relevant) training examples. A similar size document set is randomly selected from the entire PubMed database and labeled as negative (irrelevant) training examples along with the negative user seeds entered by the user.

Next, a text classifier based Multi-Stage Filtering process takes place to gradually refine or reduce the test set citations. Given a large test set, e.g., the entire PubMed database as test set, one common problem with most text classification methods is that the classifiers identify many false positives. The main objective of this filtering process is to systematically remove the false positives from the final retrieval results.

4.3 Multi-Stage Filtering Using Text Classifiers

First, a Three-Stage Filtering procedure is presented in this section. At the beginning of the Three-Stage Filtering process, three classifiers are learned from the expanded training data set. In this study, Naive Bayes (NB) [35], Support Vector Machines (SVM) [35] and k-Nearest Neighbor (kNN) [35] text classifiers are used as the three base text classifiers.

The three learned classifiers are applied in 3 stages in refining and filtering of the retrieval results.

Stage 1 text classifier (TC1) is first used to classify the test set into two categories: relevant (positive) and irrelevant (negative). The set of citations predicted as positive by TC1 is often quite large, including many false positives.

To remove the false positives from the retrieval results, citations classified as positive by TC1 undergo two more classification steps using stage 2 Text Classifier (TC2) and stage 3 Text Classifier (TC3) in a pipeline fashion. Only the citations classified as positive from the previous stage are fed into the next classification stage for further refinement.

PARS uses three-stage text classifier based filtering to refine the set of retrieved citations. A different choice of the base classification scheme at each of the three stages can lead to a slightly different final retrieval results. We take a conservative approach in choosing the classification schemes: apply classification schemes having high recalls in the early stages of the filtering pipeline. And apply classifiers that are most susceptible in incorrectly remove true positives in later stages in the filtering pipeline, i.e., to preserve the true positives in the retrieval results as much as possible.

This approach is different from the standard voting schemes used for classification ensembles [84], where the accuracy of the voting schemes does not depend on the order of the classifiers used. This approach is also different from the active learning methods. While most active learning methods focus on improving the classification accuracy by incrementally modifying the training data, in PARS, the training data is improved just once through expansion. All the text classifiers are trained using the same expanded training data. After that, PARS focuses on reducing the false-positives in the search output rather than improving the accuracy of the text classifiers.

The three-stage filtering method may be generalized into a filtering pipeline with more or less stages, i.e., Multi-Stage Filtering. For example, one may use two-stage or four-stage

filtering with two or four classifiers respectively. Classification schemes other than Naive Bayes, kNN, and SVM may be used in each stage of the process. The conservative nature of the classifiers should be considered when ordering the classifiers in the filtering stages.

4.4 Ranking the Search Output

The classification results from TC3 represent a much-improved set of highly relevant citations to the user information need. However, it may still contain some of the false-positives. As the final step, PARS ranks the resulting set of citations based on the Cosine Similarity [36] of each against the Master Citation. The top ranked citations are presented as the final retrieval results.

CHAPTER 5

EVALUATION PROCEDURE FOR PARS

The previous two chapters presented the methodologies of building the PARS system. How effective are those methods in retrieving the relevant information from PubMed? Which of these techniques is more suitable for personalized retrieval? Does PARS produce better search output than its competitive systems? To answer these questions, a formal procedure is needed to measure the effectiveness and efficiency of the PARS system in this study.

The effectiveness of the PARS system has been evaluated using both a controlled test data and with an empirical study that analyzes the feedbacks from the real PARS users. With the controlled test data, for each information need, gold standard for relevant vs. non-relevant citations are available as they have been previously determined by a panel of domain experts. Therefore, it is straightforward to compare the retrieval results from the PARS system to those gold standard results. With empirical study, the retrieval results from the PARS system need to be validated with the individual users assuming their domain expertise. The details of the two evaluation procedures are discussed below.

5.1 Testing PARS Using a Test Collection

A test collection to measure the effectiveness of the PARS system needs to satisfy the following three main requirements:

1. The document collection is sufficiently large,
2. A set of pre-defined information needs is available, and
3. For each information need, there are gold standards concerning relevant vs. non-relevant citations labeled by domain experts.

TREC 2005 genomic track ad-hoc retrieval task dataset [71] is test data that satisfies all

these requirements, and is chosen as the test collection to measure the effectiveness of PARS.

5.1.1 TREC 2005 Dataset

A subset of TREC 2005 genomic track [71] was used in this study. In particular, Ad-Hoc retrieval task dataset from the TREC 2005 genomic track was used. This is the same dataset used in the original PMRA experiment study [14]. It contains 50 different information needs (topics) from biologists. Information needs in the TREC 2005 dataset are referred as topics ranging from number 100 to 149. The entire document collection for the 50 topics contains 34,633 unique PubMed citations. Each topic corresponds to a different subset of documents ranging in size from 290 to 1356 documents. Relevance of each document to the given topic was judged by a group of scientists. According to their opinion all the documents in the document pool were labeled as: Definitely Relevant (DR), Possibly Relevant (PR) or Non Relevant (NR). Table 1 shows the document distribution of the ten topics having the highest number of relevant documents (DR+PR) in the TREC 2005 Dataset [71].

Table 1: Ten topics (information needs) that contain the highest number of relevant documents (DR- Definitely Relevant documents, PR- Possibly Relevant documents, NR- Not Relevant documents).

Topic ID	# DR (Definitely Relevant)	# PR (Possibly Relevant)	# NR (Non Relevant)	Total # documents
117	527	182	385	1094
146	370	67	388	825
114	210	169	375	754
120	223	122	182	527
126	190	117	1013	1320
142	151	120	257	528
108	76	127	889	1092
111	109	93	473	675
107	76	114	294	484
109	165	14	210	389

To better suit the evaluation of PARS, we made the following modifications to the TREC

2005 data:

1. We have combined the DR and PR document sets to one Relevant set for each information need.

2. We have replaced the explicit description of an information need with a set of seed documents randomly selected from the Relevant set of documents from that information need.

After the modification, the two PARS methodologies have been evaluated using the unranked retrieval setting and the ranked retrieval setting.

In the unranked retrieval setting, measuring the effectiveness of PARS using the retrieved document set without considering the order of the retrieved documents. Evaluation measures, including *precision*, *recall* and $F_1 - Score$ have been used.

Precision measures the fraction of the citations returned that are actually relevant.

$$Precision = \frac{\#\ Relevant\ \ citations\ \ in\ \ the\ \ search\ \ output}{Total\ \ number\ \ of\ \ citations\ \ in\ \ the\ \ search\ \ output} \tag{40}$$

Recall measures the fraction of all the relevant citations in the dataset that are found and returned to the user.

$$Recall = \frac{\#\ Relevant\ \ citations\ \ in\ \ the\ \ search\ \ output}{Total\ \ number\ \ of\ \ Relevant\ \ citations\ \ in\ \ the\ \ dataset} \tag{41}$$

$F_1 - Score$ gives a balanced measure of *precision* and *recall*. Precision and recall are standalone measures to quantify the quality of an information retrieval system. However, $F_1 - Score$ measures the weighted average of the precision and recall. To maximize the information retrieval efficiency, both the precision and recall values need to be maximized. The higher $F_1 - Score$ indicates a high precision and recall values. Therefore, we can easily

measure the success of an information retrieval system using a single measure ($F_1 - Score$).

$$F_1 - Score = \frac{(2 \cdot precision \cdot recall)}{precision + recall} \tag{42}$$

In the ranked retrieval setting, users are interested in the accuracy of the top N retrieved citations. If a user can see more relevant citations in the top N citations, then the system performance is considered better. *"Precision at N"* measure, as shown in Equation 43 is used to evaluate the ranked citations.

$$Precision\ \ at\ \ N = \frac{Relevant\ \ citations\ \ within\ \ the\ \ first\ \ N\ \ citations\ \ in\ \ the\ \ search\ \ output}{N} \tag{43}$$

Equation 43 can be used to calculate any *"Precision at N"* measure for any given N. For example, *"Precision at 10"* measures how many relevant documents are present in the top 10 documents in the PARS search output. Finally, Mean Average Precision (MAP) values have been used to measure the overall accuracies of search results for the information needs. MAP is the arithmetic mean of multiple *"Precision at N"* values for the given information need.

5.2 Evaluating PARS with the Real Users (Scientists)

This section discusses the recruiting process for the real world users of the PARS system. It also presents the procedure used to evaluate the performance of the PARS system with real world users and the evaluation methods that compare the performance of the PARS system with the performance of the MedlineRanker [15] and the MScanner [16] PIR systems. Figure 26 presents the procedure of the experiment setup.

All the participants of this empirical study are from MTSU (Middle Tennessee State University). The institution IRB approval was obtained before the start of the study. Participants mostly were recruited from the department of biology and department of chemistry.

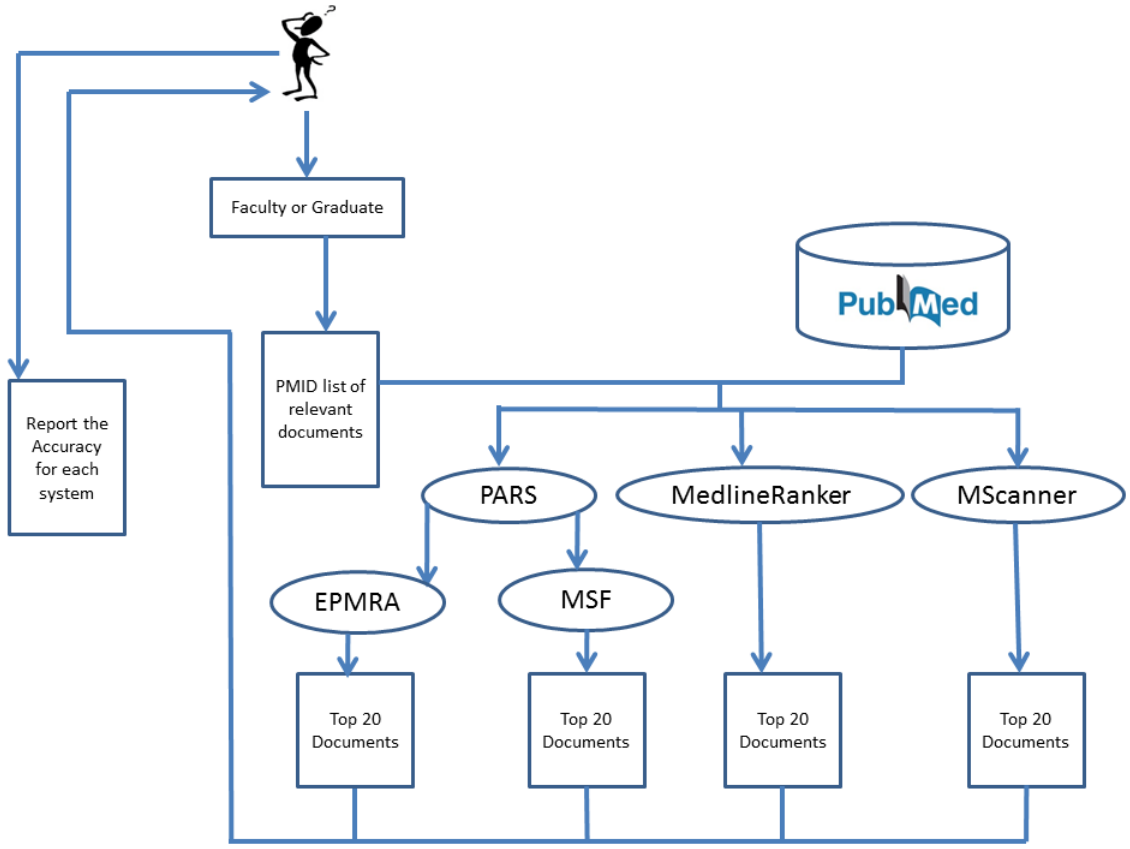

Figure 26: Real-world experiment procedure to evaluate the PARS methodologies.

Participants include undergraduate students, graduate students and faculty members. The participants were recruited through personal interviews. No advertisements or flyers were used in this study. Twenty users participated in this study.

The following 5 steps can describe the procedure used to recruit and conduct the empirical study:

1. Schedule personal interviews with potential users and explain the study, the expectations and the study goals to each user. If they agree to participate the study, they are asked to sign the consent to participate form.

2. Each participant is asked to formulate a research topic related to his research work. Based on the research topic, the participant is asked to provide 5-10 PubMed citations

closely related to the topic. These citations are used as the seed documents by PARS for retrieval. When performing comparative study with MedlineRanker [15] and MScanner [16], these *seed* documents were used to present the information need in those systems as well.

3. The two PARS methodologies, MedlineRanker [15] and MScanner [16] were then used to retrieve relevant citations from PubMed. The top 20 citations from each system were collected as the search output.

4. The top citations from each system, together with an abstract for each citation, were presented to the participants without specifying which system was used to retrieve the citations. The participants labeled each citation according to whether they think it is relevant to their research topic based on reading the abstract.

5. Finally the P10 and P20 measures were computed for each system, and the results were compared across different systems.

In this study, the participants were required to provide the initial seed documents, read the four sets of abstracts returned from the systems, and to provide explicit feedbacks for the search output. Each participant volunteered at least 3 hours of his or her time for this study.

CHAPTER 6

RESULTS AND DISCUSSION

In this study, we evaluate the performance of PARS using two approaches, one with a test collection and one with a real world information retrieval experiment with subject experts.

The TREC 2005 Genomics track ad-hoc retrieval task dataset [71] is used as the test collection in this study. It consists of 50 information needs. Each information need in the TREC dataset contains a document pool labeled as Definitely Relevant (DR), Possibly Relevant (PR) and Not Relevant (NR). To perform a binary assessment of the retrieved documents, we combine the DR and PR documents into a single class, the relevant documents to the given information need.

To mimic the real world information retrieval using the TREC dataset [71], it is necessary to form an information need in the form of a small set of input citations, referred to as the *seed* documents. Since the *seed* document set is not readily available in the TREC dataset, we created the seed document sets for each information need using the "Relevant" document set in the TREC dataset. To form the seed citation set for the given information need, n citations were randomly selected from the "Relevant" citation set and labeled as the user seeds for a given TREC information need. In this study, the number of user seed citations (n) varies corresponding to the values in the set $\{5, 10, 15, 20, 25\}$. Since the maximum seed citation set size is 25, the size of the "Relevant" documents for the given information need must be greater than 25. 21 out of 50 information needs in the TREC dataset has 30 or fewer relevant citations. We did not perform information retrieval for these information needs due to lack of relevant documents in the dataset. The rest of the 29 information needs in the TREC dataset were used in this study.

The experimental results from this study are presented in three sections. The first section presents the results of EPMRA based PARS methodology performed using the TREC

dataset. The second section presents the results from the Multi-Stage Filtering method using the TREC dataset. Finally, the experimental results of the PARS methodologies in a real world information retrieval setting are presented. In this section, it compares the results from the PARS methodologies with those from the two existing PIR systems namely the MedlineRanker [15] and the MScanner [16].

6.1 Experiment Results in EPMRA Method

The following procedure is used to test the EPMRA method in PARS using the TREC 2005 dataset.

Experiment Procedure:

1. Randomly select *n* seed citations from the "Relevant" document set of the given information need.

2. Combine multiple seed citations into a single citation (Master Citation). There are multiple ways of combining the set of seed citations into a single citation:

   - The first method, the All-Inclusive method, simply combines the terms from all the seed citations;
   - The second method, the Intersection method, forms the new citations by only including terms that simultaneously appeared in every single seed citation, i.e., intersection of all the seed citations;
   - The third method, the At-Least-Two method, forms the new citation by including terms appearing in at least two seed citations.

   We experimentally compared these three methods using the Basic method. That is, combine the PubMed related article lists obtained from the individual seed citations, and sort all the derived articles according to their PMRA similarity values.

3. *n* user seeds are first combined using the three methods discussed above. Then the relatedness between the combined citation and all the other citations in TREC dataset is computed using the PMRA method. Then, all the citations are sorted in descending order based on their PMRA values. The precision is computed in terms of the percentage of the top citations in the sorted list that were originally labeled as Definitely Relevant (DR) or Possibly Relevant (PR). Each experiment for the given information need is repeated 10 times by randomly choosing different seeds from the DR and PR set.

6.1.1 Experiment Results

The ten topics having the highest number of relevant documents (DR and PR) were used in the first set of experiments. Table 2 shows results from the three seed combination methods and the Basic method. It is seen that the Intersection method and At-Least-Two method out-perform the Basic and the All-Inclusive methods. The boldface values show the highest P10 result for the given information need. In fact, the Intersection method and At-Least-Two method were able to produce citation lists that are 15% or more accurate than those produced by the other two methods. Also, the Basic method produced the worst results in terms of P10 measure. This validates the approach to extend the basic PMRA approach in the context of personalized information retrieval. If a user provides multiple citations to an IR system, then it is advantageous to combine these citations when representing the information need in the IR system. This is because each citation represents a different aspect of the user information need. Combined citation, on the other hand, provides a more complete representation of the user need. Relevant citations found using the combined citation are of a higher quality than those found using individual citation.

Figure 27 shows the change of P10 values for different seed sizes for four different information needs. Intersection and At-Least-Two methods outperformed the Basic method and the All-Inclusive method. Moreover, P10 values for the Intersection and the At-Least-Two

Table 2: Average P10 values for ten information needs. The average was calculated using five different seed sizes i.e, 5, 10, 15, 20, 25. For each seed size, the experiment was repeated 10 times by randomly selecting different seeds from the "Relevant" set of the information need.

Topic ID	Basic method	All-Inclusive method	Intersection method	At-Least-Two method
117	0.255	0.720	0.875	**0.915**
146	0.220	0.400	0.885	**0.930**
114	0.225	0.260	0.790	**0.815**
120	0.365	0.520	0.855	**0.885**
126	0.130	0.385	0.475	**0.545**
142	0.455	0.360	0.480	**0.650**
108	0.275	0.085	0.615	**0.635**
111	0.085	0.285	**0.750**	0.725
107	0.180	0.350	**0.695**	0.655
109	0.550	0.490	**0.990**	0.970

methods are more stable than those of the other two methods. These results show that the At-Least-Two and the Intersection methods are less sensitive to seed document choices in the initial user citation set. According to Figure 27, in most cases, providing a larger seed citation set corresponds to a higher P10 value. This means that providing more seed citations has the effect of increasing the efficiency of retrieval.

In the next step, we compare the At-Least-Two and Intersection method more thoroughly using the P10, P100 and P1000 measures. Table 3 shows the 95% confidence interval and the mean for the P10, P100 and P1000 values for the At-Least-Two and Intersection methods for the top ten information needs from the TREC 2005 dataset [71]. To calculate the confidence interval, we assumed that standard deviation (σ) for the average *"Precision at N"* values are unknown. Also, sample size for the confidence interval is less than 30 observations. Therefore, we used t-distribution based confidence interval for the mean P values.

According to the Table 3, all the mean P10, P100 and P1000 values of the At-Least-Two methods are greater than those of the Intersection method. However, 95% Confidence Interval (CI) for the mean precision values for the two methods overlapped. For the At-

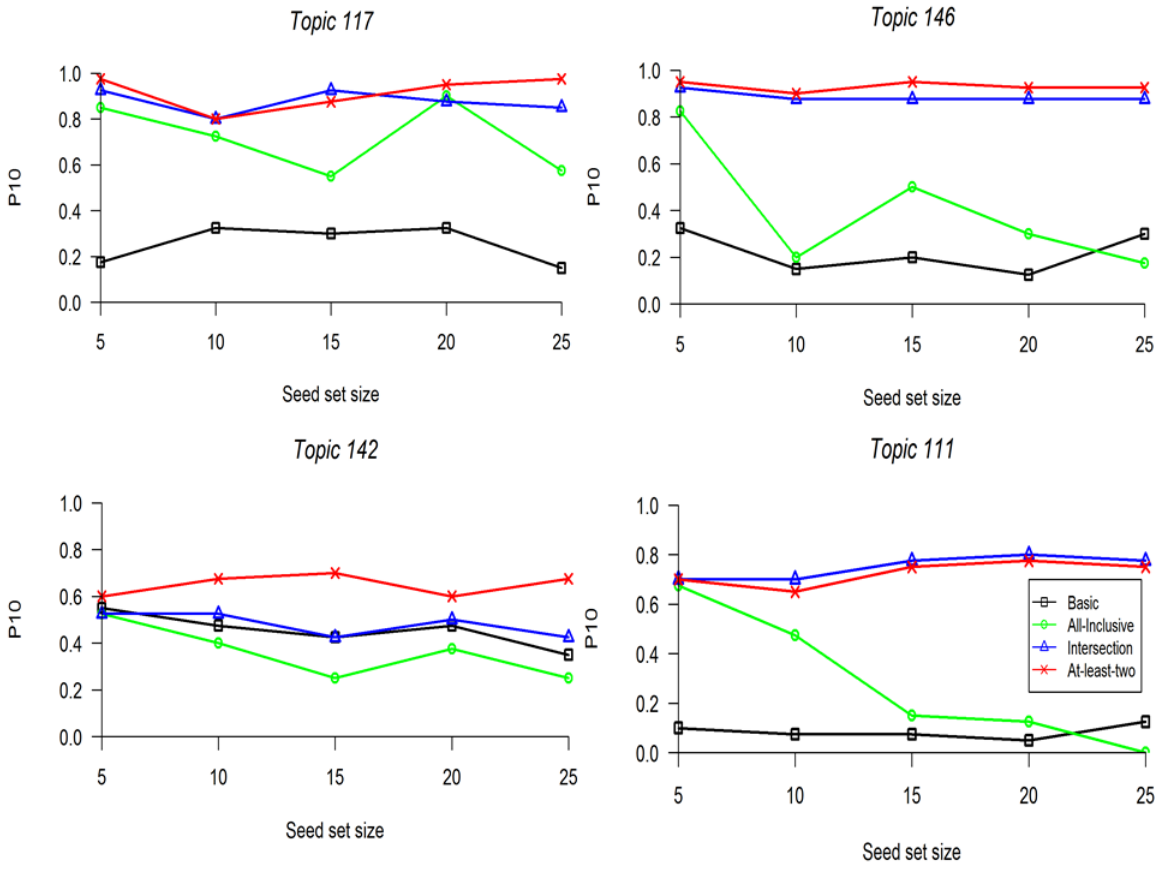

Figure 27: Distribution of P10 values for different seed set sizes for 117, 146, 142 and 111 information needs (topics).

Table 3: Mean and 95% confidence interval for P10, P100 and P1000 values. Confidence Intervals were calculated using the t-distributions assuming unknown standard deviation of P values.

"Precision at N"	Intersection Method		At-Least-Two method	
	Mean	95 % CI	Mean	95% CI
P10	0.741	(0.617, 0.865)	0.773	(0.668, 0.881)
P100	0.640	(0.494, 0.787)	0.655	(0.521, 0.790)
P1000	0.256	(0.160, 0.352)	0.265	(0.169, 0.361)

Least-Two method, all the lower bounds of the confidence intervals are higher than those of the Intersection method. Also, upper bound of CI's for the At-Least-Two method are greater than those of the Intersection method. Therefore, by considering all the values one can say that the At-least two method performed better than the Intersection method. Therefore, in

the next set of experiments, the performance of PARS with the At-Least-Two method is singled out to compare against the original results from the TREC conference.

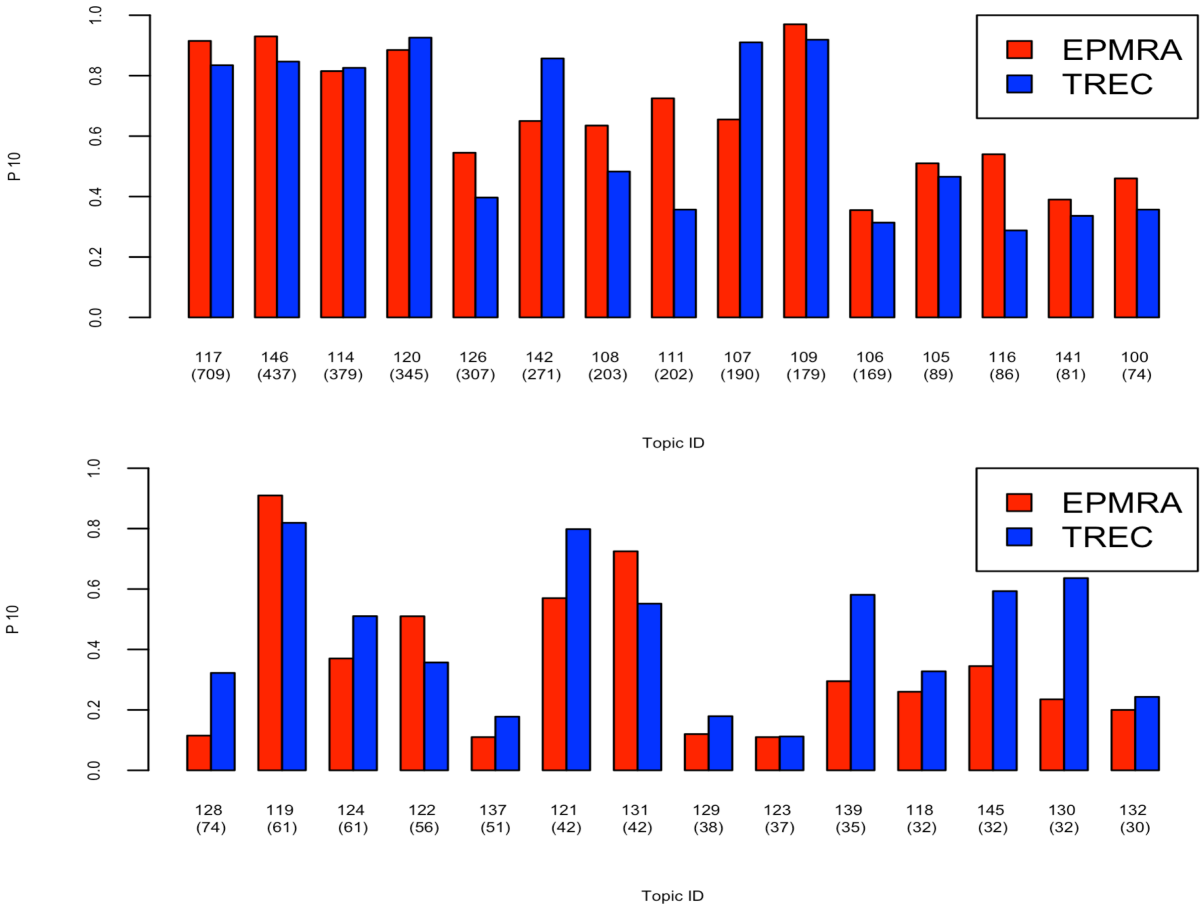

Figure 28: P10 values for PARS with the EPMRA method compared against the TREC conference results for 29 information needs. The value in the parenthesis shows the *"Relevant"* set size for each information need.

Figures 28, 29 and 30 show the P10, P100, and P1000 values for PARS with the EPMRA method compared against the TREC 2005 conference results for the 29 different information needs respectively.

It is observed that the results from PARS with the EPMRA are mostly comparable with those from the original TREC conference. In Figure 28, for 14 out of 29 topics, PARS with the EPMRA outperforms the TREC conference results. In Figure 29, PARS performed better

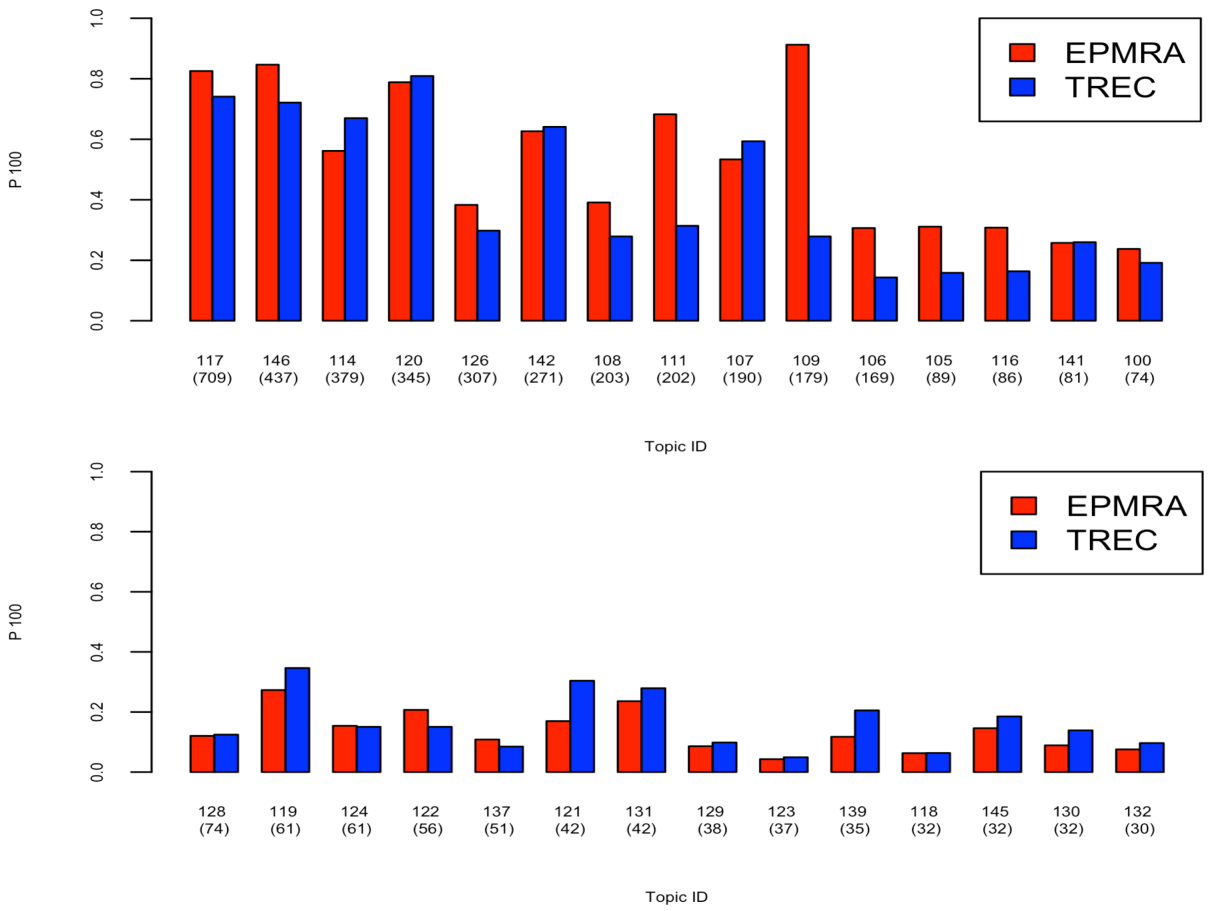

Figure 29: P100 values for PARS with the EPMRA method compared against the TREC conference results for 29 information needs. The value in the parenthesis shows the *"Relevant"* set size for each information need.

in 13 topics when measured with P100 measure. Similarly, for 18 out of 29 topics, PARS with the EPMRA produced higher P1000 values than those from the TREC conference results.

For PARS with the EPMRA, the P10, P100 and P1000 values were calculated by varying the size of the initial user set to 5, 10, 15, 20 and 25. For each size, the experiment was repeated 10 times, where a different randomly selected set of seed documents were used each time. Therefore, PARS with the EPMRA method replicate the PubMed information retrieval in a more systematic approach.

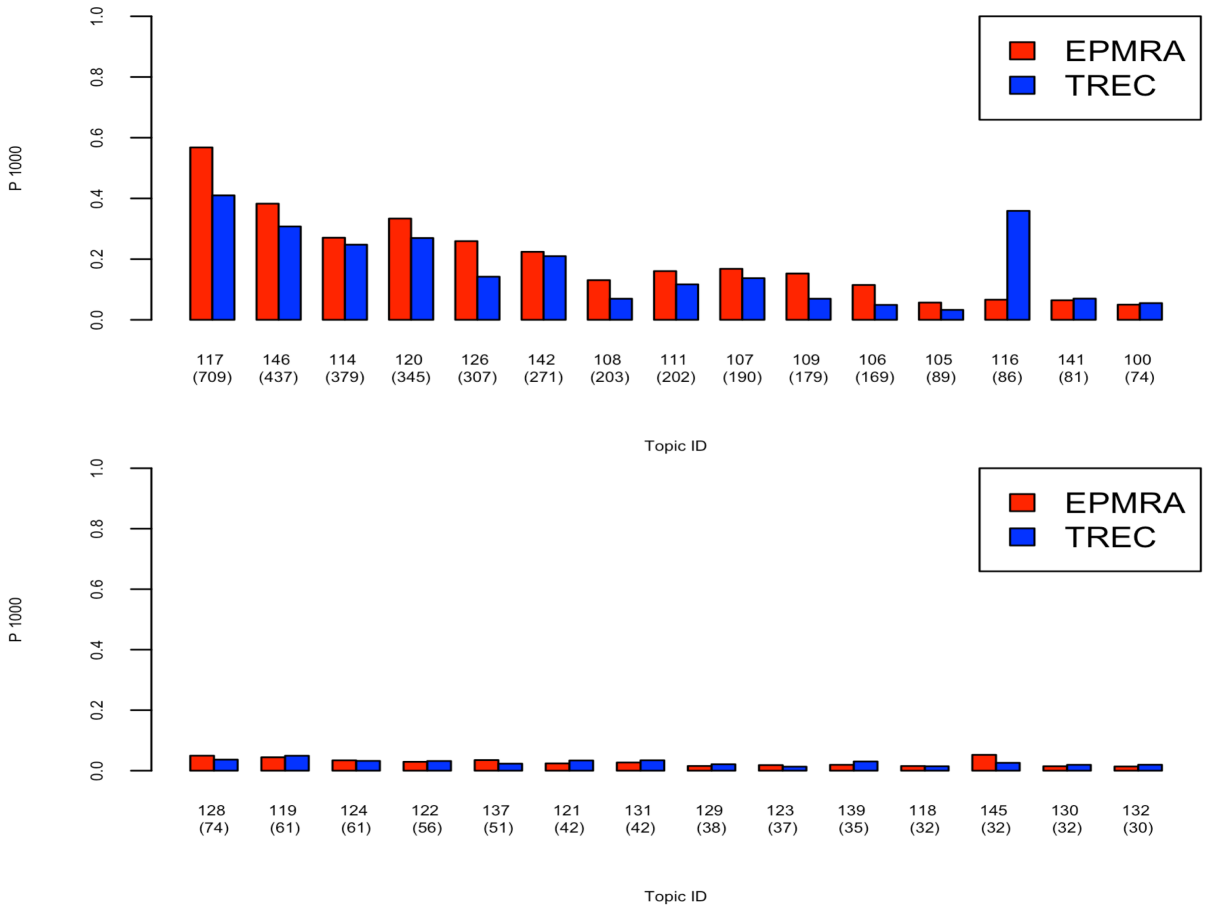

Figure 30: P1000 values for PARS with the EPMRA method compared against the TREC conference results for 29 information needs. The value in the parenthesis shows the *"Relevant"* set size for each information need.

6.2 Experiment Results for the MSF Method

In this experiment, we present the experimental results of the Multi-Stage Filtering (MSF) method in PARS. In particular, a Three Stage Filtering (TSF) method is employed to find the relevant citations for the given information need. First, the experimental procedure of the TSF method is described. Then, the results of the TSF method tested using the 10 topics having the highest number of relevant documents (Definitely Relevant and Possibly Relevant) from the TREC data [71] are discussed.

6.2.1 Experiment Procedure

For each chosen information need (topic), TSF uses the following steps to form the user information need and to retrieve the relevant citations. Steps 1-5 are used to form the expanded training data to be used to train the text classifiers. Steps 6-9 perform the three stage filtering to select the ”relevant” citations. Step 10 ranks the final set of citations to present to the user:

1. n ($n = \{5, 10, 15, 20, 25\}$) citations are randomly selected from the Definitely Relevant (DR) and Possibly Relevant (PR) set of the topic. These n citations are labeled as the user seeds for the current topic;

2. The PMRA lists for the *seeds* are retrieved and combined to form the *Target Set*;

3. The *seeds* are pre-processed into terms and used to form the *Master Citation*;

4. N ($N = 50$) citations that have the highest cosine similarity to the *Master Citation* are computed from the *Target Set*; The highest similar citations and the original n seeds form the positive training examples;

5. Randomly select $n + N$ citations from the TREC data to form the negative examples. These negative examples together with the positive training examples derived in step 4 form the expanded training data;

6. Train the three base Text Classifiers using the expanded training data; This results in three trained text classifiers: TC1, TC2, and TC3;

7. Classify the TREC data using TC1; TC1 classifies a subset of citations as “relevant”;

8. Apply TC 2 to refine and reduce the set of the “relevant” citations;

9. Apply TC3 to further refine and reduce the set of the “relevant” citations;

10. Compute the Cosine similarities between each of the "relevant" citations (from step 9) and the *Master Citation*. Rank the "relevant citations" according to the Cosine similarity values computed.

Each experiment is repeated 10 times by randomly selecting the seeds from the given information need. *Seed* set size (n) ranges from 5 to 25 with an increment of 5. Results for the 10 information needs are presented in the results section.

6.2.3 Experiment Results

The following experiments were designed to evaluate the effectiveness of the three main components in the TSF approach; (1) expanding training set size by building Target Set and forming Master Citation, (2) Three-Stage Filtering process, and (3) ranking of the final output.

Expanding the Training Data

If the size of the initial user *seeds*, n, equals 5, after adding n negative training examples, the size of the initial training data is 10. After expanding the training data, a larger training set size of size 110 is obtained. This larger training set is used to train the three base classifiers. For kNN, three nearest neighbors are used to classify the new instances. Linear SVM method from the LibSVM [76] in WEKA [77] is used as the SVM text classifier. The classification accuracies obtained using the expanded training data are compared against those of the original training data (*seed* only training data). Equation 44 calculates the improvement of accuracy for a topic:

$$AI = \frac{(AETS - ASTS)}{ASTS} * 100 \tag{44}$$

Where, AI = *Accuracy Improvement*, $ASTS$= *Accuracy with the Seed only Training Set (Initial Training Set)* and $AETS$= *Accuracy with the Expanded Training Set)*. The average

improvement computed based on the results obtained from five different training sets is reported in Table 4. It shows the average improvement of classification accuracies for the 10 information needs. From Table 4, it is clear that expanding the training set using the Target Set and Master Citation lead to a big improvement of the classification accuracies of the base text classifiers across all 10 information needs. The PMRA feature helps to build a high quality small *Target Set*, and Cosine Similarity is effective in identifying the citations having the highest similarity to the *Master Citation* from the *Target Set*.

Table 4: Improvement of classification accuracy for the three base classifiers using expanded training data.

	Average Improvement (%)		
Topic ID	NB	3-NN	SVM
117	+ 61.20	+ 184.81	+ 60.92
146	+ 82.26	+ 155.81	+ 14.86
114	+ 73.82	+ 158.35	+ 68.80
120	+ 150.31	+ 334.95	+ 61.13
126	+ 150.31	+ 110.65	+ 11.75
142	+ 182.62	+ 190.39	+ 150.09
108	+ 67.37	+ 131.71	+ 27.42
111	+ 89.73	+ 225.86	+ 139.93
107	+ 66.96	+ 104.96	+ 24.58
109	+ 67.20	+ 82.81	+ 9.74

Three-Stage Filtering Results

To select the classification schemes for each of the three stages for the PARS with TSF approach is to select classification schemes that are less likely to eliminate the true positive citations in early stages of the filtering process. Since Naive Bayes (NB) classifier has a higher recall value than that of SVM and 3-NN classifiers, NB classifier is used in the first stage filtering process. The 3-NN classifier is used in the second stage filtering, and SVM is used in the final stage of the filtering process.

For comparison purposes, two-stage filtering process using the NB and the 3-NN classifications, as well as three one-stage filtering processes with just the NB, the 3-NN, or the

SVM were also performed. Figure 31 shows the $F_1 - Scores$ using the five different filtering methods for four topics. $F_1 - Scores$ are calculated using Equation 42. The $F_1 - Score$ provides a balanced measure of both recall and precision.

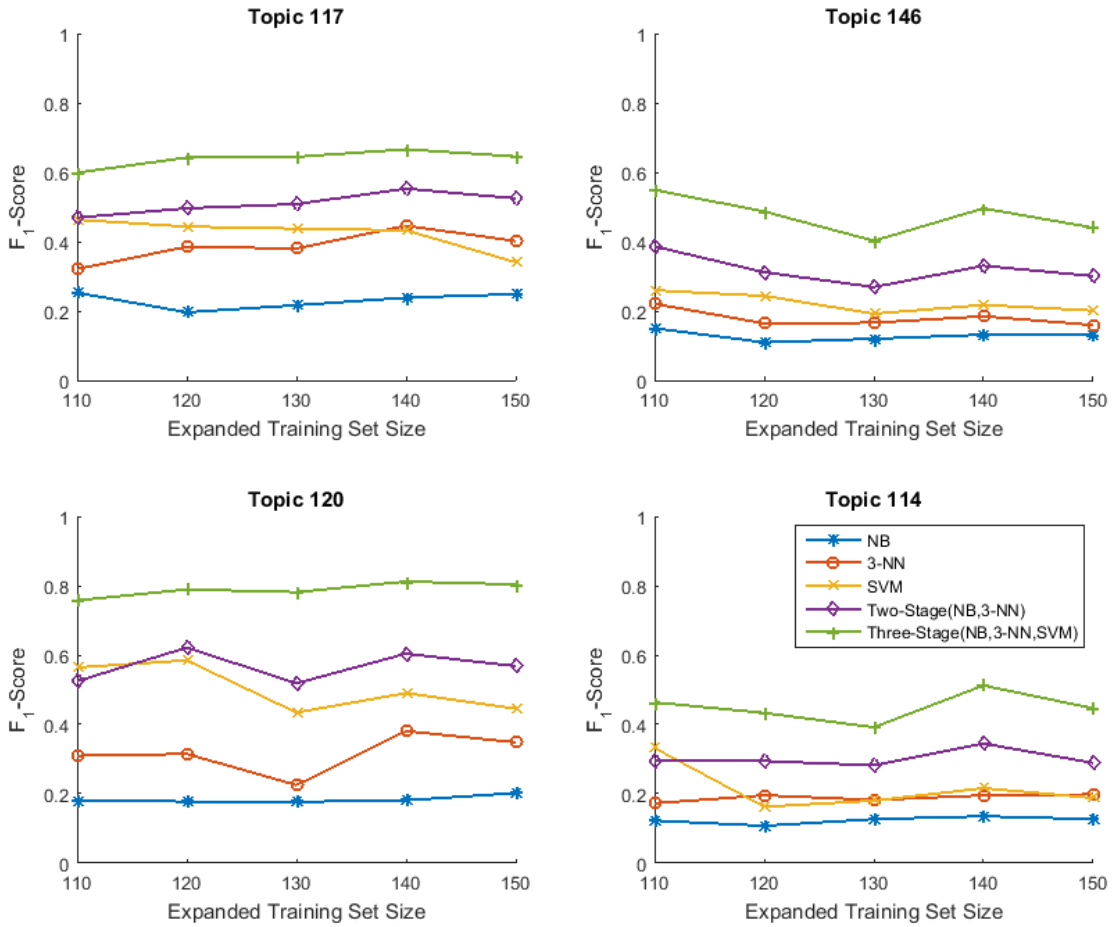

Figure 31: $F_1 - Scores$ computed for topics 117, 146, 120 and 114 using Three-stage (NB, 3-NN, SVM) filtering, Two Stage (NB, 3-NN) filtering, and NB, SVM and 3-NN only methods.

As shown in Figure 31, the $F_1 - Scores$ of the classification results from the Three-Stage Method (NB, 3-NN, SVM) are higher than the other four methods across all four topics.

The final search output is dependent on the order of the text classifiers chosen for the three stages. For example, a three-stage filtering with the three classifiers in the order: (1) NB, (2) 3-NN, and (3) SVM produced a different result than one with the three classifiers in the order: (1) NB, (2) SVM and (3) 3-NN. The first ordering produced a better result with

a higher $F_1 - Score$. This is because the NB text classifier generates classifications with high recall for a given topic. In the second stage, the 3-NN is used, followed by the SVM text classifier. Since the SVM outperformed the other two text classifiers in the one-stage method, SVM is used in the third stage to get an accurate final output.

Ranked Retrieved Results

PARS ranks the set of predicted relevant citations from the three-stage process against the *Master Citation* using cosine similarity. The top N ranked citations are considered the final retrieval results to be presented to the user. The retrieval accuracy is computed in terms of the percentage of the top N citations having the label Definitely Relevant (DR) or Possibly Relevant (PR) to the information need. Figure 32 shows the retrieval accuracy of the top 10 citations (P10) and Figure 33 shows the retrieval accuracy for the top 100 citations (P100) in the final search output for the 29 topics.

It is observed that P10 and P100 values for the PARS with TSF method are mostly comparable with the TREC results. In addition, for 14 out of 29 topics the P10 values are higher than those of the TREC results. Similarly, for 18 out of 29 topics the P100 values of the PARS with TSF method are higher than those of the TREC results. Considering that the percentage of the citations relevant to each topic is rather small in the entire database, these results are very encouraging. However, a much lower accuracy is observed for few topics. This may be attributed to the fact that there are too few positive citations for the topic. For each experiment n ($n = 5, 10, 15, 20, 25$) positive citations were selected to form the seed citation set. The number of remaining positive citations is very small compared to the size of the TREC dataset. This makes the retrieval tasks extremely hard if only the top 100 citations are to be returned. However, the average of P100 measure from the PARS system present 17% improvement over the results reported by the systems competed at the TREC conference [71]. These results show that the PARS MSF approach is very effective

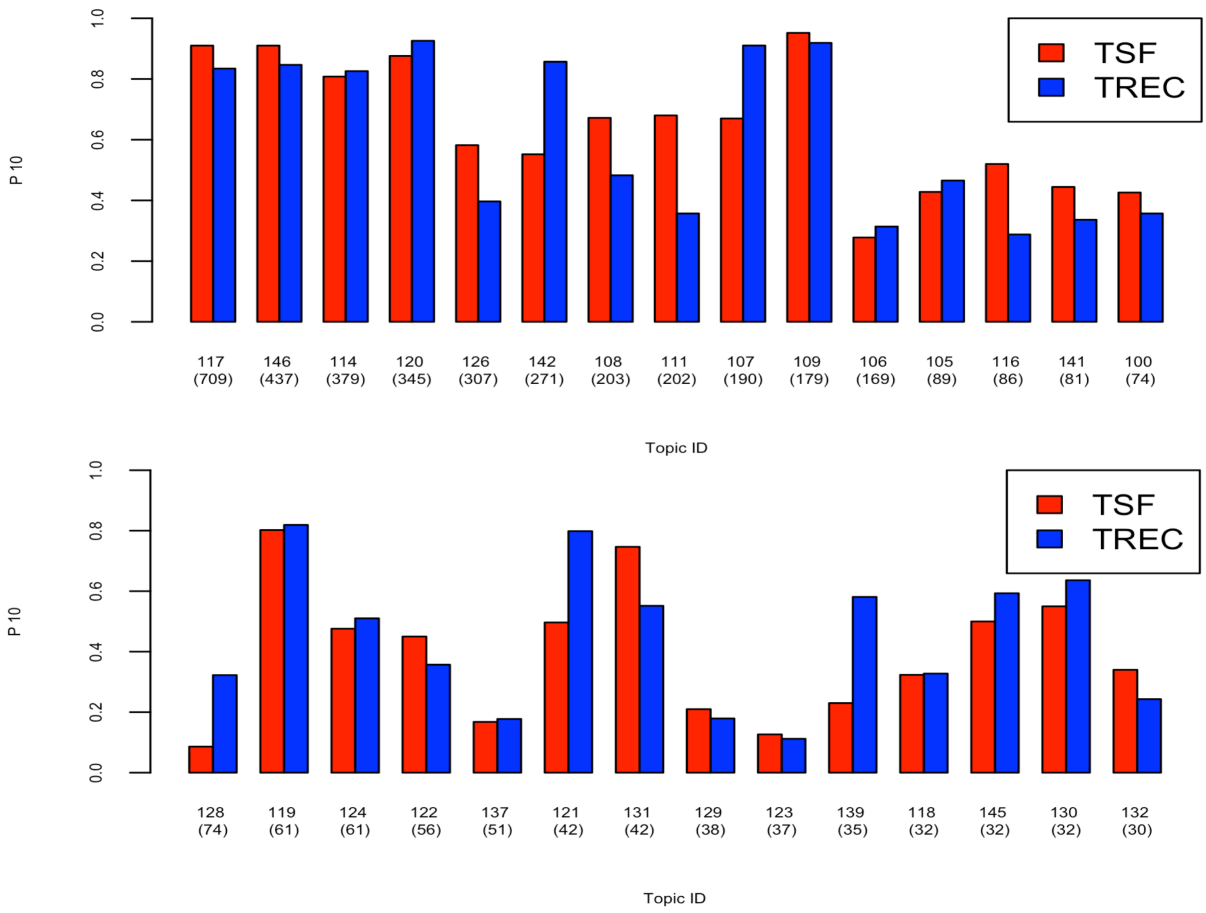

Figure 32: P10 values for PARS with the TSF method compared against the TREC conference results for 29 information needs.

for personalized information retrieval.

Next, we compared the EPMRA and TSF method results together with the TREC conference results. Figure 34 shows the P10 values for the EPMRA method, the TSF method and the TREC conference results. Similarly, Figure 35 shows the P100 values for the EPMRA method, TSF method, and the TREC conference results. According to the Figure 34 and Figure 35, one can clearly see that both methods in PARS produced similar results for most of the topics. In Figure 34, the P10 values in the EPMRA method outperformed the TSF method for 16 topics in TREC data. Moreover, P10 values for 17 topics in PARS outperformed the TREC results. According to the Figure 35, P100 values for 17 topics outperformed TREC results. Therefore, one can say that PARS produced better results than

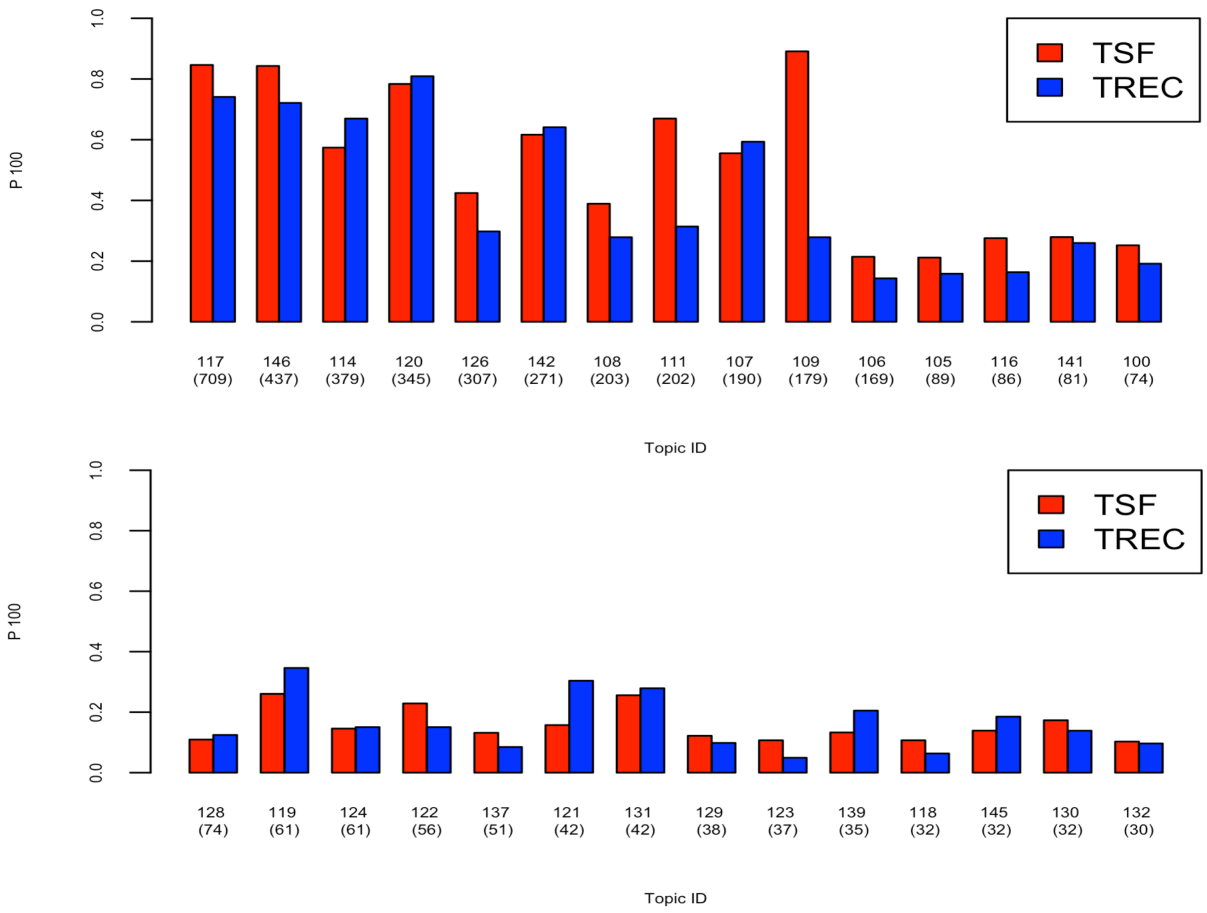

Figure 33: P100 values for PARS with the TSF method compared against the TREC conference results for 29 information needs.

the existing TREC results. Then we summarized the EPMRA, TSF and TREC results for 29 information needs.

Table 5: Average P10 and P100 values for EPMRA, TSF and TREC for 29 information needs. Information needs (topics) were divided into three groups based on their relevant set size.

TREC topic group	P10			P100		
	EPMRA	TSF	TREC	EPMRA	TSF	TREC
1-10	0.7725	0.7612	0.7355	0.6551	0.6593	0.5345
11-20	0.4270	0.4078	0.3946	0.2283	0.2110	0.1774
21-29	0.3178	0.3915	0.4469	0.1139	0.1442	0.1578

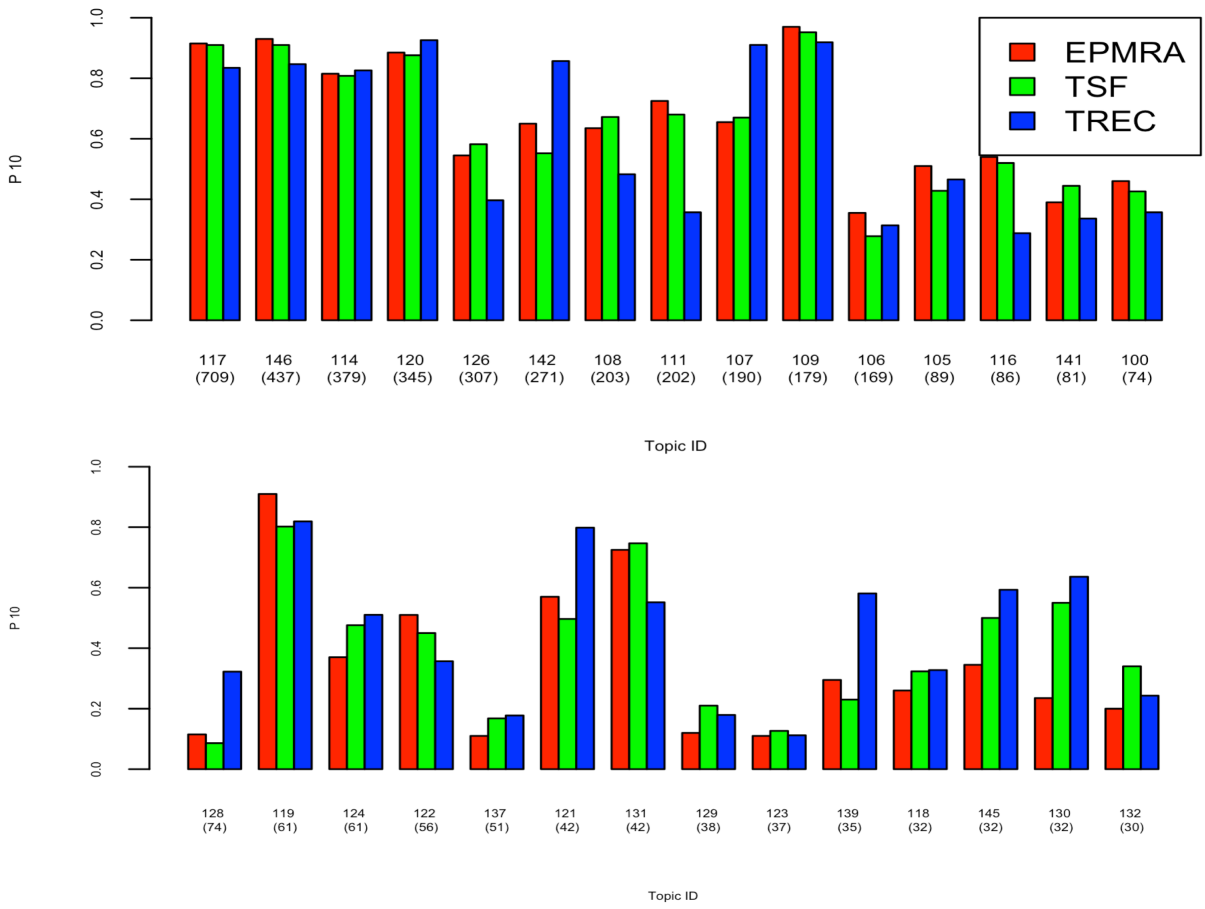

Figure 34: P10 values for PARS with the EPMRA method and the TSF method compared against the TREC conference results for 29 information needs.

According to the Table 5, one can clearly see that the PARS results for first 20 topics outperformed TREC conference results. However, the last nine topics do not produce good results compared to TREC results. This may be attributed to the fact that there are too few positive citations for the topic. For example, topic 132 has a total of 30 positive citations. For each experiment n ($n = 5, 10, 15, 20, 25$) positive citations are selected to form the seed citation set. The number of remaining positive citations is very small compared to the size of the TREC dataset. This makes the retrieval tasks harder in this experiment setting.

In the next section, both PARS with EPMRA and PARS with MSF were tested in a real world personalized information retrieval task. In this experiment, researchers in the field of biology and chemistry were recruited to participate in the experiment. Results from the two

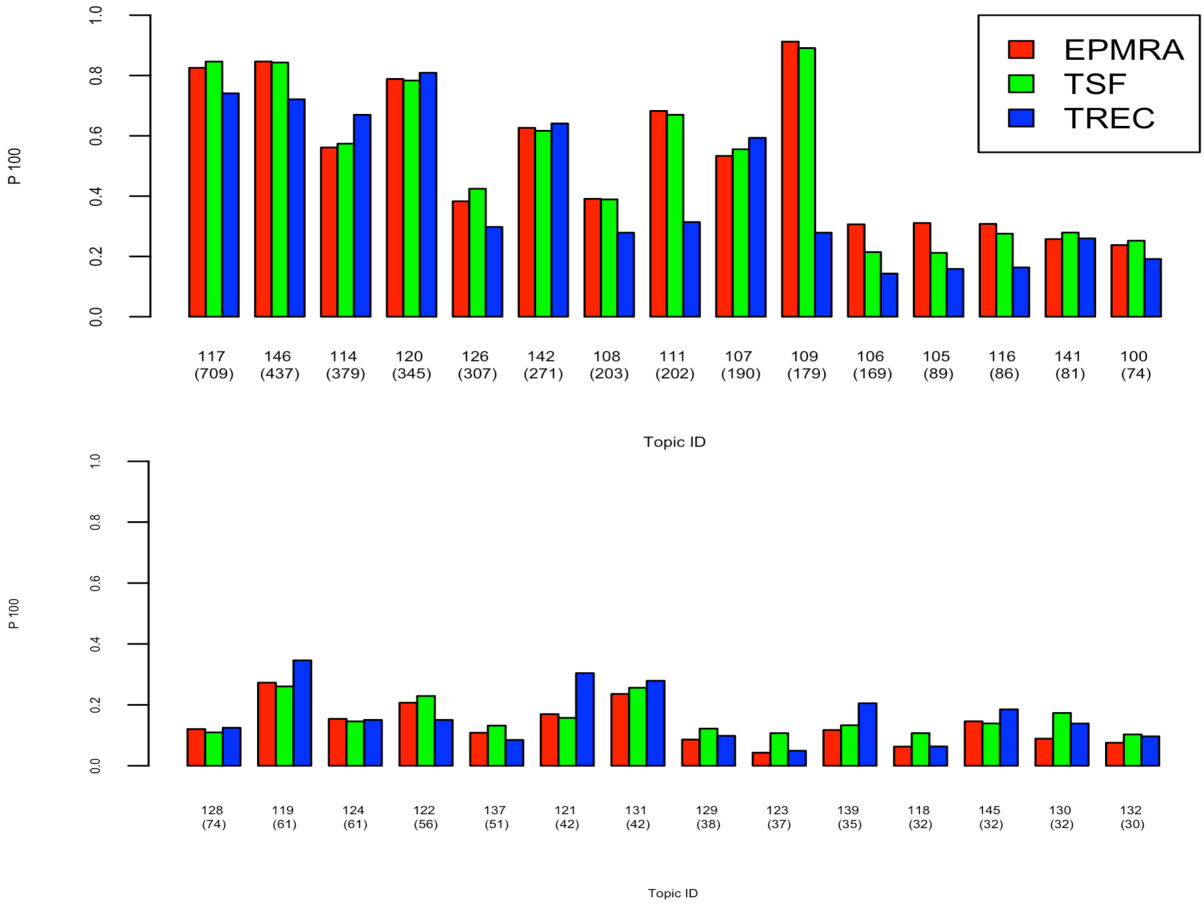

Figure 35: P100 values for PARS with the EPMRA method and the TSF method compared against the TREC conference results for 29 information needs.

PARS approaches were compared with the two competitive PIR systems: the MedlineRanker [15] and the MScanner [16].

6.3 Experiment Results with the Real Users (Scientists)

An experimental study was conducted to evaluate the effectiveness of the PARS system in a real world personalized information retrieval application. The 23 researchers in the fields of biology and chemistry agreed to participate in the study. Due to various reasons, 15 participants (10 from department of Biology and 5 from the department of Chemistry) were able to complete the study.

First, a set of *seed* citations were collected from each participant as the input to the

PIR system. The *seed* citations from each participant closely relates to his or her specific information need. Table 5 shows the distribution of the sizes of the *seed* set used in this study. Many participants provided a small set of *seed* citations to define their information need. This observation reinforces the need of IR system such as PARS, which is capable of finding the relevant citations from a small set of input citations.

Table 6: The distribution of seed set size for 15 participants in the empirical study.

Number of seed citations for the information need	Number of Participants
1	2
2	2
3	4
4	1
5	3
6	2
10	1

Based on the seed citations provided by the participants, the information need for each participant was constructed for the MedlineRanker [15], the MScanner [16], and the two versions of the PARS system; PARS with EPMRA and PARS with TSF. Retrieval results from each of these four systems were collected and ranked. The top 20 citations from each system were presented to the participants for verification, i.e., whether each is considered relevant to their information needs. Figure 34 shows the example search output given to the user to evaluate the search results.

Each citation in the user search output is presented with the journal information, author information, citation abstract, and the PMID information. The participants need to go through each citation returned from each of the four systems, and specify whether each citation is relevant to the information need. The three options are: Definitely Relevant, Possibly Relevant, and Non-Relevant.

Figure 37 shows the distribution of the P10 values for the top 10 citations returned to the 15 participants. Figure 38 shows the distribution of the P20 values for the top 20 citations

1. Biochim Biophys Acta. 2007 May;1768(5):1059-69. Epub 2007 Jan 24.

Identification of a segment in the precursor of pulmonary surfactant protein SP-B, potentially involved in pH-dependent membrane assembly of the protein.

Serrano AG(1), Cabré EJ, Pérez-Gil J.

Author information:
(1)Dept. Bioquímica y Biología Molecular I, Facultad de Biología, Universidad Complutense, 28040 Madrid, Spain.

In the present work, the hydrophobic properties of proSP-B, the precursor of pulmonary surfactant protein SP-B, have been analyzed under different pH conditions, and the sequence segment at position 111-135 of the N-terminal domain of the precursor has been detected as potentially possessing pH-dependent hydrophobic properties. We have studied the structure and lipid-protein interactions of the synthetic peptides BpH, with sequence corresponding to the segment 111-135 of proSP-B, and BpH-W, bearing the conservative substitution F127W to use the tryptophan as an intrinsic fluorescent probe. Peptide BpH-W interacts with both zwitterionic and anionic phospholipid vesicles at neutral pH, as monitored by the blue-shifted maximum emission of its tryptophan reporter. Insertion of tryptophan into the membranes is further improved at pH 5.0, especially in negatively-charged membranes. Peptides BpH and BpH-W also showed pH-dependent properties to insert into phospholipid monolayers. We have also found that the single sequence variation F120K decreases substantially the interaction of this segment with phospholipid surfaces as well as its pH-dependent insertion into deeper regions of the membranes. We hypothesize that this region could be involved in pH-triggered conformational changes occurring in proSP-B along the exocytic pathway of surfactant in type II cells, leading to the exposure of the appropriate segments for processing and assembly of SP-B within surfactant lipids.

PMID: 17306759 [PubMed - indexed for MEDLINE]

Please tick the appropriate box:					
Relevant:		Possibly Relevant		Not Relevant	

Figure 36: Sample search output given to the participant to receive their feedback about the search results.

returned to the participants for each of the four PIR systems.

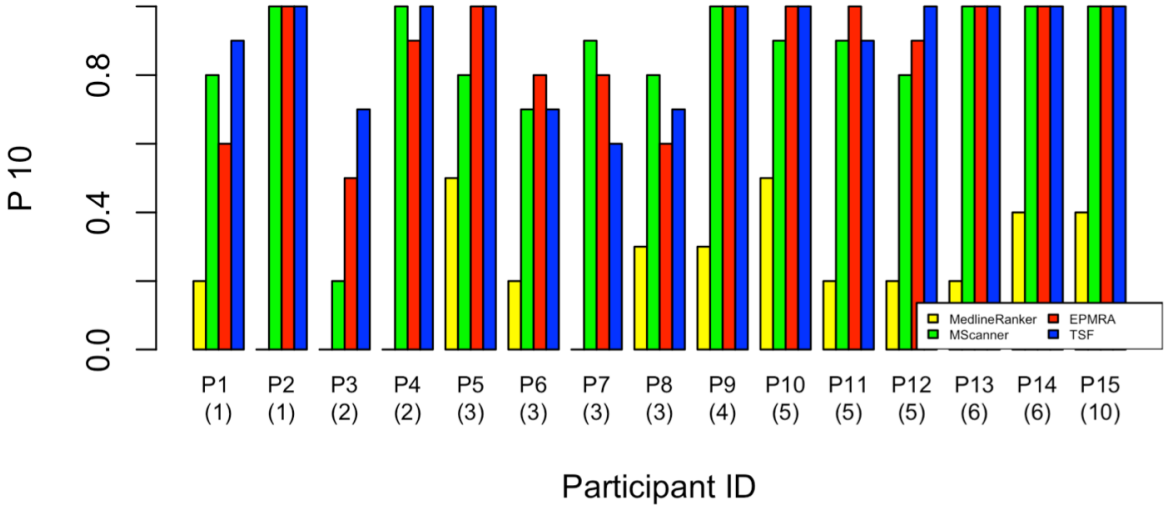

Figure 37: Distribution of the P10 values for the participants for the MedlineRanker, the MScanner, PARS with EPMRA, and PARS with TSF. Each participant seed set size is given in parenthesis.

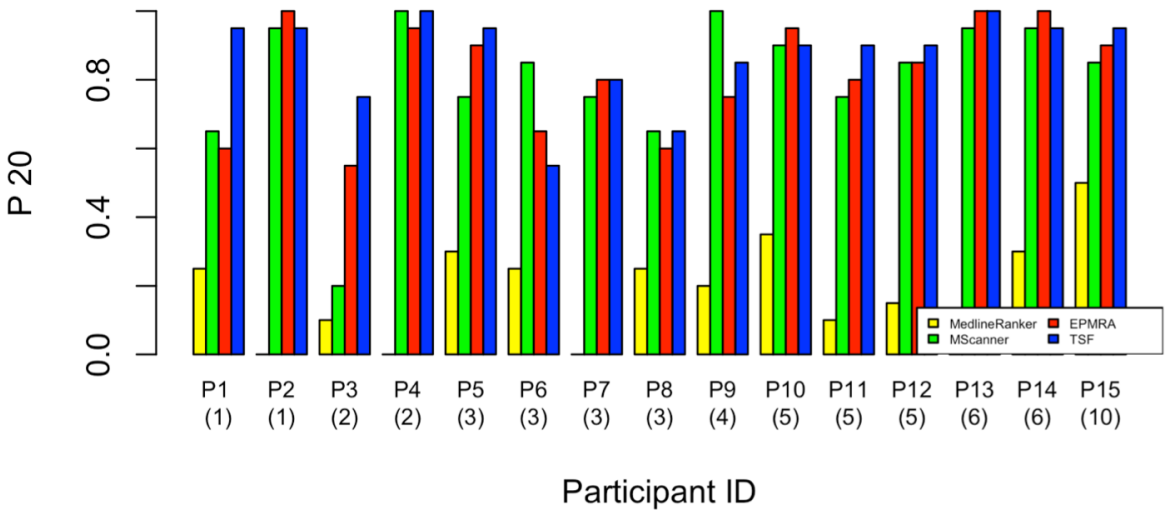

Figure 38: Distribution of the P20 values for the participants for the MedlineRanker, the MScanner, PARS with EPMRA, and PARS with TSF. Each participant seed set size is given in the parenthesis.

Figure 37 and 38 show that the P10 and P20 values for MedlineRanker are the lowest compared to the other three systems, i.e., it has the worst performance in retrieving the relevant citations. The poor performance of MedlineRanker can mainly be attributed to the current experimental setup, where MedlineRanker searches from a subset of the PubMed database for the relevant citations. On the other hand, the other three methods search for the relevant citations for an information need from the entire PubMed database.

From results shown in Figure 37, the performance of the MScanner, PARS with EPMRA and PARS with TSF are comparable in finding the relevant citations to the information need. For 5 of the 15 participants, the retrieved results are the same using all three methods. For the remaining 10 participants, results from MScanner is better for two participants and PARS is better for the other 8 participants. From the results in Figure 38, one can observe that the two PARS methods clearly outperform the MScanner search results. For 13 of the 15 participants, the P20 values for the two PARS values tied or are higher than that of MScanner. Moreover, MScanner can only find the PubMed citations when the MeSH information is presented. However, the PARS methods can handle citations even with missing MeSH

annotations. Therefore, one can say that PARS methods perform better than MScanner.

Table 7 shows the initial target set size and the predicted relevant set size from each classifier in the PARS TSF method. It is observed that PARS with TSF method is able to dramatically reduce the search output size by eliminating the false positives citations. However, for some topics, after three filtering steps, there are still over 20,000 predicted relevant citations. One possible solution to this is to increase the number of filtering steps to eliminate more false positive citations in the final result.

Table 7: Predicted Relevant document set size from each classifier in the TSF method for each participant in the study

Participant ID	Target Set Size (LOS=3)	NB Predicted set size	KNN Predicted set size	SVM Predicted set size
P1	55667	38629	19227	12973
P2	105254	59128	25179	13099
P3	16414	13951	12327	11104
P4	94683	62526	33943	10744
P5	98443	42157	18713	4407
P6	168464	51929	28498	10927
P7	86292	63803	25479	4890
P8	12232	10228	5838	3563
P9	68937	28620	20314	12826
P10	99615	84497	58794	39982
P11	41047	33115	9643	3210
P12	322755	160183	60879	21785
P13	138344	65171	8130	3772
P14	64015	35209	10336	2654
P15	108026	76081	52174	26573

CHAPTER 7

CONCLUSIONS

The main objective of this thesis work is to build a PIR system to help the PubMed users in searching for citations in PubMed that are relevant to their information need. The main assumption we made about the users of this system is that they are able to provide a small set of input citations directly related to their research topic of interest. This eliminates the need for the user to thoroughly articulate their information need as keywords at the search prompt. One main goal of this PIR system is to be able to retrieve and present to the user a small set of "highly relevant" citations as opposed to a long list of citations containing many false-relevant citations.

One of the main difficulties with the PIR system is to try to achieve high retrieval quality by training a PIR system using only a small training data supplied by the user. In this study, two approaches have been developed to address this difficulty. In the first approach, an Extended PMRA (EPMRA) based probability similarity measure was developed to better utilize the information presented in *seed* citations provided by the user to search for citations that are similar to a more comprehensive representation of the user's information need. In the second approach, a classification based multi-stage filtering procedure was developed to gradually eliminate false positive citations from the search results in order to improve the quality of the final set of citations.

Both approaches have been evaluated using the TREC 2005 Genomic track dataset in a controlled experiment setting. For the EPMRA approach, three different methods have been developed to extend the PMRA measure, the All-inclusive, the Intersection, and the At-Least-Two methods. The experimental results show that simply combining the PMRA lists derived from individual citations is not an appropriate method to find the relevant citations for the given information need. The At-Least-Two method is the best method to extend PMRA when a user provides multiple *seed* citations. This method best captures the

important terms and discards the less important terms from the *seed* documents. In contrast, the Intersection method uses only the terms appearing in all the *seed* documents, leading to a representation with few terms. The small number of terms from the combined citation is insufficient in accurately computing similarities between pairwise citations.

For the multi-stage filtering approach, the training set is first expanded to a reasonably large dataset based on the *seed* citations. Including citations considered highly similar to the *Master Citation* based on the PMRA similarity measure expands this small training data. With this expanded data, the NB, kNN and SVM text classifiers are better trained, leading to classifiers of higher quality. Experimental results show that the procedure of expanding the training set is successful in producing better quality classifiers. These trained classifiers are used in the three-stage filtering method, which successively remove the citations incorrectly classified as relevant from the results. For all the information needs, the $F_1 - Scores$ of the three-stage method improved dramatically over the base text classifiers. Also, there is a significant improvement in P100 measure for a majority of the information needs over the TREC conference results. Therefore, one can conclude that the three-stage filtering method improves the quality of the final retrieval results.

Empirical evaluation with the biology and chemistry researchers in the real world applications show that the two PARS approaches are able to produce information retrieval results that are of better quality than those returned from the existing PIR systems, the MScanner and the MedlineRanker. Also, the PARS approaches are able to handle citations with missing MeSH annotations, while the MScanner is unable. In summary, PARS provides better information retrieval results for PubMed compared to the existing systems.

For future study, we would like to expand the TSF method to approaches with more than three filtering stages and to perform more extensive empirical evaluations of such approaches. We would also like to apply the TSF approach to other information retrieval applications, such as IR in an online document collection, such as Google Scholar, or IR

in the more general web applications. Developing schemes to combine the two PARS approaches for a better information retrieval experience is also a topic of further study.

BIBLIOGRAPHY

[1] Google Scholar Home Page. (2015, 04 30). Retrieved from Google Scholar: https://scholar.google.com/

[2] PubMed Home Page. (2015, 04 30). Retrieved from PubMed: http://www.ncbi.nlm.nih.gov/pubmed

[3] SAO/NASA Astrophysics Data System. (2015, 04 30). Retrieved from Astrophysics Data System: http://adswww.harvard.edu/

[4] CiteSeerX Home Page. (2015, 04 30). Retrieved from CiteSeerX: http://citeseer.ist.psu.edu/index

[5] PubMed Fact Sheet. (2015, 05 02). Retrieved from U. S National Library of Medicine: http://www.nlm.nih.gov/pubs/factsheets/pubmed.html

[6] MEDLINE Fact Sheet. (2015, 05 02). Retrieved from U. S National Library of Medicine: http://www.nlm.nih.gov/pubs/factsheets/medline.html

[7] Dogan, R. I., Murray, G. C., Neveol, A., & Lu, Z. (2009). Understanding PubMed user search behavior through log analysis. Database, 2009, bap018, Oxford University Press.

[8] PubMed Advanced Search Builder. (2015, 05 02). Retrieved from PubMed: http://www.ncbi.nlm.nih.gov/pubmed/advanced

[9] Chapman, D. (2009). Advanced search features of PubMed. Journal of the Canadian Academy of Child and Adolescent Psychiatry, 18(1), Pg. 58.

[10] Lu, Z., Wilbur, W. J., McEntyre, J. R., Iskhakov, A., & Szilagyi, L. (2009, November). Finding query suggestions for PubMed. In AMIA.

[11] PubMed's Automatic Term Mapping Enhanced. Published in NLM Tech Bulletin. 2004 Nov- Dec;(341):e7.

[12] Medical Subject Headings (MeSH) Fact Sheet. (2015, 05 02). from U. S National Library of Medicine: http://www.nlm.nih.gov/pubs/factsheets/mesh.html

[13] PubMed Tutorial - Building the Search Filters. (2016, 01 12) from PubMed Tutorials. https://www.nlm.nih.gov/bsd/disted/pubmedtutorial/020_210.html

[14] Lin, J., & Wilbur, W. J. (2007). PubMed related articles: a probabilistic topic-based model for content similarity. BMC bioinformatics, 8(1), Pg. 423.

[15] Fontaine, J. F., Barbosa-Silva, A., Schaefer, M., Huska, M. R., Muro, E. M., & Andrade-Navarro, M. A. (2009). MedlineRanker: flexible ranking of biomedical literature. Nucleic acids research, 37(suppl 2), W141-W146.

[16] Poulter, G. L., Rubin, D. L., Altman, R. B., & Seoighe, C. (2008). MScanner: a classifier for retrieving Medline citations. BMC bioinformatics, 9(1), Pg. 108.

[17] Goetz, T., & von der Lieth, C. W. (2005). PubFinder: a tool for improving retrieval rate of relevant PubMed abstracts. Nucleic acids research, 33(suppl 2), W774-W778.

[18] Kim, J. J., Pezik, P., & Rebholz-Schuhmann, D. (2008). MedEvi: retrieving textual evidence of relations between biomedical concepts from Medline.Bioinformatics, 24(11), 1410-1412.

[19] Rebholz-Schuhmann, D., Kirsch, H., Arregui, M., Gaudan, S., Riethoven, M., & Stoehr, P. (2007). EBIMedtext crunching to gather facts for proteins from Medline. Bioinformatics, 23(2), e237-e244.

[20] Soldatos, T. G., ODonoghue, S. I., Satagopam, V. P., Barbosa-Silva, A., Pavlopoulos, G. A., Wanderley-Nogueira, A. C., ... & Schneider, R. (2012). Caipirini: using gene sets to rank literature. BioData mining, 5(1), Pg. 1.

[21] Quertle Biomedical Search Engine. (2016 02 10), Retrieved from: https://www.quetzal-search.info/home

[22] Errami, M., Wren, J. D., Hicks, J. M., & Garner, H. R. (2007). eTBLAST: a web server to identify expert reviewers, appropriate journals and similar publications. Nucleic acids research, 35(suppl 2), W12-W15.

[23] Teevan, J., Dumais, S. T., & Horvitz, E. (2005, August). Personalizing search via automated analysis of interests and activities. In Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval (pp. 449-456). ACM.

[24] Qiu, F., & Cho, J. (2006, May). Automatic identification of user interest for personalized search. In Proceedings of the 15th international conference on World Wide Web (pp. 727-736). ACM.

[25] Shen, X., Tan, B., & Zhai, C. (2005, October). Implicit user modeling for personalized search. In Proceedings of the 14th ACM international conference on Information and knowledge management (pp. 824-831). ACM.

[26] Smalheiser, N. R., Zhou, W., & Torvik, V. I. (2008). Anne O'Tate: A tool to support user-driven summarization, drill-down and browsing of PubMed search results. Journal of biomedical discovery and collaboration, 3(1), Pg. 2.

[27] Yamamoto, Y., & Takagi, T. (2007). Biomedical knowledge navigation by literature clustering. Journal of biomedical informatics, 40(2), 114-130.

[28] Plikus, M. V., Zhang, Z., & Chuong, C. M. (2006). PubFocus: semantic MEDLINE/PubMed citations analytics through integration of controlled biomedical dictionaries and ranking algorithm. BMC bioinformatics, 7(1), Pg. 1.

[29] Doms, A., & Schroeder, M. (2005). GoPubMed: exploring PubMed with the gene ontology. Nucleic acids research, 33(suppl 2), W783-W786.

[30] Ade, A. S., Wright, Z. C., Bookvich, A. V., & Athey, B. D. (2009). MiSearch adaptive pubMed search tool. Bioinformatics, 25(7), 974-976.

[31] Tsai, R. T. H., Dai, H. J., Lai, P. T., & Huang, C. H. (2009). PubMed-EX: a web browser extension to enhance PubMed search with text mining features.Bioinformatics, 25(22), 3031-3032.

[32] Yu, H., Kim, T., Oh, J., Ko, I., Kim, S., & Han, W. S. (2010). Enabling multi-level relevance feedback on PubMed by integrating rank learning into DBMS.BMC bioinformatics, 11(2), Pg. 1.

[33] Joachims, T. (2006, August). Training linear SVMs in linear time. InProceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 217-226). ACM.

[34] Crawling & Indexing. (2016, 01 20). Google Inside Search: `https://www.google.com/insidesearch/howsearchworks/crawling-indexing.html`

[35] Manning, C. D., Raghavan, P., & Schtze, H. (2008). Introduction to information retrieval (Vol. 1, No. 1, p. 496), Cambridge: Cambridge University Press.

[36] Lee, M., Pincombe, B., & Welsh, M. (2005). An empirical evaluation of models of text document similarity. Cognitive Science.

[37] Salton, G., Fox, E. A., & Wu, H. (1983). Extended Boolean information retrieval. Communications of the ACM, 26(11), 1022-1036.

[38] Kokare, M., Chatterji, B. N., & Biswas, P. K. (2003, October). Comparison of similarity metrics for texture image retrieval. In TENCON 2003. Conference on Convergent Technologies for the Asia-Pacific Region (Vol. 2, pp. 571-575). IEEE.

[39] Gionis, A., Indyk, P., & Motwani, R. (1999, September). Similarity search in high dimensions via hashing. In VLDB (Vol. 99, pp. 518-529).

[40] Croft, W. B., & Harper, D. J. (1979). Using probabilistic models of document retrieval without relevance information. Journal of documentation, 35(4), 285-295.

[41] Amati, G., & Van Rijsbergen, C. J. (2002). Probabilistic models of information retrieval based on measuring the divergence from randomness.ACM Transactions on Information Systems (TOIS), 20(4), 357-389.

[42] Turtle, H., & Croft, W. B. (1991). Evaluation of an inference network-based retrieval model. ACM Transactions on Information Systems (TOIS), 9(3), 187-222.

[43] Turtle, H., & Croft, W. B. (1989, December). Inference networks for document retrieval. In Proceedings of the 13th annual international ACM SIGIR conference on Research and development in information retrieval (pp. 1-24). ACM.

[44] Ponte, J. M., & Croft, W. B. (1998, August). A language modeling approach to information retrieval. In Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval (pp. 275-281). ACM.

[45] Croft, B., & Lafferty, J. (Eds.). (2013). Language modeling for information retrieval (Vol. 13). Springer Science & Business Media.

[46] Page, L., Brin, S., Motwani, R., & Winograd, T. (1999). The PageRank citation ranking: bringing order to the Web. Stanford InfoLab.

[47] Algorithms. (2016, 01 20). Google Inside Search: https://www.google.com/insidesearch/howsearchworks/algorithms.html

[48] Brin, S., & Page, L. (2012). Reprint of: The anatomy of a large-scale hypertextual web search engine. Computer networks, 56(18), 3825-3833.

[49] Macdonald, C., & Ounis, I. (2006, August). Combining fields in known-item email search. In Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval (pp. 675-676). ACM.

[50] Kelly, L., Bunbury, P., & Jones, G. J. (2012). Evaluating personal information retrieval. In Advances in Information Retrieval (pp. 544-547). Springer Berlin Heidelberg.

[51] Kim, J. (2012). Retrieval and evaluation techniques for personal information(Doctoral dissertation, University of Massachusetts Amherst).

[52] Zanasi, A. Text Mining and its Applications to Intelligence, CRM and Knowledge Management. Advances in Management Information.

[53] Gmez-Prez, A., Ortiz-Rodrguez, F., & Villazn-Terrazas, B. (2006, May). Ontology-based legal information retrieval to improve the information access in e-government. In Proceedings of the 15th international conference on World Wide Web (pp. 1007-1008). ACM.

[54] Luo, G., Tang, C., Yang, H., & Wei, X. (2008, October). MedSearch: a specialized search engine for medical information retrieval. In Proceedings of the 17th ACM conference on Information and knowledge management (pp. 143-152). ACM.

[55] Science.gov Home Page. (2015, 04 30). Retrieved from PubMed: https://www.science.gov/

[56] Lu, Z., Wilbur, W. J., McEntyre, J. R., Iskhakov, A., & Szilagyi, L. (2009, November). Finding query suggestions for PubMed. In AMIA.

[57] Medical Subject Headings (MeSH) in MEDLINE/PubMed: A Tutorial. (2015, 05 10). Retrieved from U.S. National Library of Medicine: https://www.nlm.nih.gov/bsd/disted/meshtutorial/introduction/

[58] PubMed Relevance Sort. (2015, 05 30). Retrieved from NLM Technical Bulletin:https://www.nlm.nih.gov/pubs/techbull/so13/so13_pm_relevance.html

[59] Computation of Similar Articles. (2015, 05 30). Retrieved from PubMed Tutorials.http://www.ncbi.nlm.nih.gov/books/NBK3827/#pubmedhelp.Computation_of_Similar_Articl

[60] Berger, A. L., Pietra, V. J. D., & Pietra, S. A. D. (1996). A maximum entropy approach to natural language processing. Computational linguistics, 22(1), 39-71.

[61] Douglas, S. M., Montelione, G. T., & Gerstein, M. (2005). PubNet: a flexible system for visualizing literature derived networks. Genome biology, 6(9), Pg. R80, BioMed Central Ltd

[62] Cherkassky, V., & Mulier, F. M. (2007). Learning from data: concepts, theory, and methods. John Wiley & Sons.

[63] Burges, C. J. (1998). A tutorial on support vector machines for pattern recognition. Data mining and knowledge discovery, 2(2), 121-167.

[64] Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. The Journal of Machine Learning Research, 3, 1157-1182.

[65] Caruana, R., & Freitag, D. (1994, July). Greedy Attribute Selection. In ICML(pp. 28-36).

[66] Yang, J., & Honavar, V. (1998). Feature subset selection using a genetic algorithm. In Feature extraction, construction and selection (pp. 117-136). Springer US.

[67] Debuse, J. C., & Rayward-Smith, V. J. (1997). Feature subset selection within a simulated annealing data mining algorithm. Journal of Intelligent Information Systems, 9(1), 57-81.

[68] Cha, S. H. (2007). Comprehensive survey on distance/similarity measures between probability density functions. City, 1(2), Pg. 1.

[69] Robertson, S. E., Walker, S., Jones, S., Hancock-Beaulieu, M. M., & Gatford, M. (1995). Okapi at TREC-3. NIST SPECIAL PUBLICATION SP,109, 109.

[70] Jones, K. S., Walker, S., & Robertson, S. E. (2000). A probabilistic model of information retrieval: development and comparative experiments: Part 2.Information Processing & Management, 36(6), 809-840.

[71] Hersh, W. R., Cohen, A. M., Roberts, P. M., & Rekapalli, H. K. (2006, November). TREC 2006 genomics track overview. In TREC.

[72] Google Images. (2016, 01 20). Retrieved from Google Image Search. https://images.google.com/

[73] The Google app Voice Search, Answers and Assistance. (2016, 01 20) Retrieved from Google App: https://www.google.com/search/about/

[74] Lu, Z., Kim, W., & Wilbur, W. J. (2009). Evaluating relevance ranking strategies for MEDLINE retrieval. Journal of the American Medical Informatics Association, 16(1), 32-36.

[75] Hansen, M. H., Hurwitz, W. N., & Madow, W. G. (1953). Sample Survey Methods and Theory, John Wiley & Sons.

[76] C.-C. Chang and C.-J. Lin. LIBSVM (2011) : a library for support vector machines. ACM Transactions on Intelligent Systems and Technology, 2:27:1–27:27.

[77] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, Ian H. Witten (2009); The WEKA Data Mining Software: An Update; SIGKDD Explorations, Volume 11, Issue 1.

[78] Dasan, V. S. (1998). U.S. Patent No. 5,761,662. Washington, DC: U.S. Patent and Trademark Office.

[79] Thomas, P. (2005). Personal information retrieval. Poster at HCSNet. Sydney, Australia (December 2005).

[80] Liu, Y. H., & Wacholder, N. (2008). Do humandeveloped index terms help users? An experimental study of MeSH terms in biomedical searching. Proceedings of the American Society for Information Science and Technology, 45(1), 1-16.

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

[82] Sebastiani, F. (2002). Machine learning in automated text categorization. ACM computing surveys (CSUR), 34(1), 1-47.

[83] Dave, K., Lawrence, S., & Pennock, D. M. (2003, May). Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In Proceedings of the 12th international conference on World Wide Web (pp. 519-528). ACM.

[84] Bauer, E., & Kohavi, R. (1999). An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. Machine learning, 36(1-2), 105-139, Chicago.