"PERSONALIZED WEB SEARCH AND EVOLUTIONARY TECHNIQUES"

Dissertation Submitted to Jawaharlal Nehru University in Partial Fulfillment of the Requirement for the Award of the Degree of

Master of Technology

Submitted by

MAYANK SAINI

Submitted to

Dr. Aditi Sharan

School of Computer & Systems Sciences

Jawaharlal Nehru University

New Delhi-110067

India

2012

जवाहरलाल नॅहरू विश्वविद्यालय

School of Computer & Systems Sciences

JAWAHAR LAL NEHRU UNIVERSITY

NEW DELHI-110067, INDIA

DECLARATION

I hereby declare that the dissertation entitled **"PERSONALIZED WEB SEARCH AND EVOLUTIONARY TECHNIQUES"**, submitted by me to the School of Computer and Systems Sciences, **Jawaharlal Nehru University**, New Delhi, in partial fulfillment of the requirements for the award of the degree of **Master of Technology** in **Computer Science and Technology**, is a bona fide work carried out by me under the supervision of **Dr. Aditi Sharan**.

The matter embodied in this dissertation has not been submitted to any other university or institution for the award of any other degree or diploma.

27/07/12

MAYANK SAINI

M.TECH-CSE

SC&SS, JNU,

New Delhi-110067

जवाहरलाल नॅहरू विश्वविद्यालय

School of Computer & Systems Sciences

JAWAHAR LAL NEHRU UNIVERSITY

NEW DELHI-110067, INDIA

CERTIFICATE

This is to certify that the dissertation entitled **"PERSONALIZED WEB SEARCH AND EVOLUTIONARY TECHNIQUES"**, submitted by Mayank Saini to the School of Computer and Systems Sciences, **Jawaharlal Nehru University**, New Delhi, in partial fulfillment of the requirements for the award of the degree of **Master of Technology** in **Computer Science and Technology**, is a bona fide work carried out by him under my supervision.

The matter embodied in this dissertation has not been submitted to any other university or institution for the award of any other degree or diploma.

Dr. Aditi Sharan

(Supervisor)

SC&SS, JNU, New Delhi

Prof. Karmeshu

(Dean)

SC&SS, JNU, New Delhi

प्रोफेसर कर्मेशु/Professor Karmeshu
डीन / Dean
स्कूल ऑफ कम्प्यूटर और पद्धति विज्ञान संस्थान
School of Computer and Systems Sciences
जवाहरलाल नेहरू विश्वविद्यालय
Jawaharlal Nehru University
नई दिल्ली / New Delhi-110067

Dedicated to my mom and dad

Acknowledgement

I would like to gratefully acknowledge the enthusiastic supervision of **Dr. Aditi Sharan** during this work. This work wouldn`t have been possible without her constant support, valuable suggestions and comments during my whole tenure of this dissertation work. I feel privileged to work under her for my master`s dissertation. Apart from the academic guidance she has always been a great mentor of mine in encouraging me to be disciplined and well organized. I must surely say, she has given her best in providing me the infrastructure required, which led to the successful completion of my dissertation. I would take this opportunity to thank her once again for her esteemed support and I, from the bottom of my heart would like to wish her the best in all her future endeavors.

I wish to thank my colleagues Mr. Nitin Prajapati, Mr. Jagendra Singh and my senior Mrs. Manju for creating a home like environment in our lab to keep the stress away. I would also like to thank my best critics Mr.Ranjeet Kumar Ranjan, Mr. Anuj Kumar, Mr. Arun Kumar Teotia and Mr. Vipin Kumar for suggesting to remove errors in my dissertation. Thank you guys!!

Finally, I would like to thank the whole faculty member of our department for clarifying my doubts throughout this work and last but not least, the JNU administration for creating such a secular and healthy environment amongst the students.

Abstract

Search engines return results mainly based on the submitted query; however, the same query could be in different contexts because individual users have different interests. To improve the relevance of search results, we have proposed personalized web search model based on a learned user profile. We have used user profile to map a user query into a set of categories which represent the user's search intention and serve as a context to disambiguate the words in the user's query.

Studies of Web search user behavior have shown that a large portion of Web search queries consist of only one to three terms. These short queries provide an indication that users of Web search engines often have difficulties crafting queries that accurately reflect their information needs. Clearly, most Web search engines provide little support for users as they attempt to construct and reformulate their queries .The goal of this research is to address the fundamental issues related to these shortcomings of the current Web search engines by establishing a new paradigm for interactive query formation by suggesting terms for query expansion.

We have discussed the appropriateness of evolutionary techniques for information retrieval along with its applications in this field. We have used genetic algorithm for finding thematically rich terms because suitability of problem characteristics with genetic algorithm. The expansion terms has shown significant improvement in retrieval effectiveness. Thematic richness of the query has been used for evaluating query quality. Experimental outcomes prove that this proposed personalized Web search system is very effective and efficient.

Table of Contents

Declaration.....i

Certificate.....ii

Acknowledgement.....iv

Abstract.....v

Table of Contents…..……...……..…….vi

List of Figures.....ix

List of Tables.....x

List of Abbreviations…….. .xi

Chapter-1 : INTRODUCTION.....1-7

1.1 Web Search: Origin and Usages.....2

1.2 Motivation……5

1.3 Aim of the Dissertation……...……..6

1.4 Outline of Dissertation.....7

Chapter-2 : INFORMATION RETRIEVAL AND EVOLUTIONARY TECHNIQUES ……. 8-16

2.1 Introduction.....8

2.2 Information Retrieval Systems.....8

2.2.1 Components of IRS.....9

2.2.2 Information Retrieval Models.....10

2.2.3 Evaluation of IRS.....10

2.3 Evolutionary Algorithms.....11

2.3.1Genetic Algorithms.....11

2.3.2Multi-Objective Genetic Algorithm (MOGA):.....12

2.4 Role of Evolutionary Algorithms for Solving IR Problems…… ……..……..…...13

2.4.1 EA`s Appropriateness for Information Retrieval ……...……..13

2.4.2 EA`s Applications in IR…….……...….14

Chapter-3 : PERSONALIZED WEB SEARCH.....17-40

3.1 Definition.....17

3.2 Historical Background 17

3.3 Foundation ……..……..18

3.3.1User Modeling 18

3.3.2Personalized Search Strategies accommodating User-context.....25

3.3.3Evaluation Approach.....29

3.4 Challenges of Personalized Web Search……..…….……..30

3.5 Literature Survey ……..….…….32

Chapter-4 : Proposed Work and Experiments 41-56

4.1 Objective.....41

4.2 Proposed Model 41

4.2.1 User Profiling ……...……42

4.2.2 Mapping queries to most relevant interest category……....44

4.2.3 GA for finding good query terms…….……...…44

4.2.4 Suggesting expansion terms to users……..……..…50

4.3 Experiments and Results……..……51

4.4 Analysis of Results.....57

4.4.1 General Role of GA……..……..……..……..…57

4.4.2 Significance of Mutation……..…..…….……..…57

4.4.3 Evaluating Thematic Richness……..……...…….……..…57

4.4.4 User Profile……...……...……..……..…59

Chapter-5 : Conclusion and Future Scope 60

References 61-67

Appendix 68

List of Figures

Figure 1.1 Overall Architecture of Research 7
Figure 2.1 Basic Information Retrieval System.....9
Figure 3.1 Methodlology Used for User Profiling 21
Figure 4.1 Proposed Generalized Framework of Personalized Web Search System 42
Figure 4.2 Proposed Framework for Finding Good Query Expansion Terms using GA 49
Figure 4.3 Average query Quality in each generation 52
Figure 4.4 Quality of best query in each generation..... 53
Figure 4.5 Average query quality in each generation for m=0 54
Figure 4.6Average query quality in each generation for m=0.1 55
Figure 4.7 Average query quality in each generation for m=0.3 56
Figure 4.8 Quality of best query over all generation for m=0 54
Figure 4.9 Quality of best query over all generation for m=0.1 55
Figure 4.10 Quality of best query over all generation for m=0.3 56
Figure 4.11 Average thematic quality of queries in each generation 58
Figure 4.12Quality of best thematic queries in each generation..... 59

List of Tables

Table 1.1 Google annual search statistics 4
Table 3.1 Document- Term matrix DT 23
Table 3.2 Document- Category matrix DC 24
Table 3.3 Category- Term matrix M represents a user profile 24
Table 3.4 Classification of Query Expantion Based upon Work Performed by User 29
Table 4.1 AlchemyApi to DMOZ Category Mapping 43
Table Appendix List of Few Sample Results 67

List of Abbreviations

IR	Information Retrieval
GA	Genetic Algorithm
MOGA	Multi-Objective Genetic Algorithm
PWS	Personalized Web Search
VSM	Vector Space Model
TF	Term Frequency
IDF	Inverse Term Frequency
IRS	Information Retrieval System
TSPR	Topic Sensitive Page Rank
ODP	Open Directory Project

CHAPTER 1

INTRODUCTION

One hundred users, one hundred needs. Current web search engines are built to serve all users, independent of the needs of any individual user. It is increasingly difficult to let the search engine know what we want. Coping with ambiguous queries has long been an important part of research in information retrieval.

User queries tend to be short, and hence often ambiguous, which can lead to inappropriate results from general search engines. Query formulation is an essential part of successful information retrieval. The result of various studies suggested that experience in using computers; web and web search engines may affect the query formulation process. Generally experienced user formulate longer and more specific query whereas the query of user with less experience consist of fewer and more generic terms [1]. The users' of search engines are extremely heterogeneous consisting of, for example, computer novices and highly skilled expertise, searchers looking for material just for fun and user requiring accurate and efficient search facility for professional purpose. A number of studies have shown that a vast majority of queries to search engines are short and under specific and user may have completely different intention for the same query. For example for the query "apple" some users may be interested in documents dealing with apple as a "fruit", while other users may want documents related to apple computers. In order to deal with such ambiguity, it is important to consider the context of user for personalized search.

Personalized search has recently got recently significant attention to disambiguate the user's query by incorporating user context. Personalized search has the potential to significantly improve user experience, for example, according to a recent statistic if we can reduce the time user spend on searching on Google by a mere 1% throughout effective personalization over 1, 87,000 person-hour (21 years) will be saved each month [2]. Unfortunately studies have shown that the vast majority of users are reluctant to provide the explicit feedback on a search result and their interest [2]. Therefore a

personalized search engine intended for a large audience has to learn the user preference automatically without any explicit input by the user.

Personalized web search is an open research area. In this work we have proposed a framework for personalized web search using query expansion incorporating thematic context along with user profile. To find good query terms we intended to use genetic algorithm. Genetic algorithm is suitable because of the problem characteristics such as high dimensional space, multiple solution etc.

1.1 Web Search: Origin and Usages

The concept of hypertext and a memory extension really came to life in July of 1945 when Vannevar Bush's "As We May Think" was published in The Atlantic Monthly magazine. He proposed the idea of a virtually limitless, fast, reliable, extensible, associative memory storage and retrieval system. Thereafter in 1960 Ted Nelson created Project Xanadu and coined the term hypertext in 1963. His goal with Project Xanadu was to create a computer network with a simple user interface that solved many social problems like attribution. While Ted was against complex markup code, broken links, and many other problems associated with traditional HTML on the WWW, much of the inspiration to create the WWW was drawn from Ted's work.

Gerard Salton (1927 - 1995), a Professor of Computer Science at University was perhaps the leading computer scientist working in the field of information retrieval during his time. He was also known as the father of modern search technology. His teams at Harvard and Cornell developed the SMART informational retrieval system. Salton's Magic Automatic Retriever of Text included important concepts like the vector space model (VSM), Inverse Document Frequency (IDF), Term Frequency (TF), term discrimination values, and relevancy feedback mechanisms.

In the beginning of the nineties, there was a complete directory of the whole World Web. These were the times when one could know all the existing servers in the web. Later, other web directories appeared. The EINet Galaxy web directory was born in January of 1994. It was organized similar to how web directories are today. The biggest reason the

EINet Galaxy became a success was that it also contained Gopher and Telnet search features in addition to its web search feature. The web size in early 1994 did not really require a web directory; however, other directories soon did follow. In April 1994 David Filo and Jerry Yang created the Yahoo! Directory as a collection of their favorite web pages. As their number of links grew they had to reorganize and become a searchable directory. As time passed and the Yahoo! Directory grew, Yahoo! began charging commercial sites for inclusion. In 1998 Rich Skrenta and a small group of friends created the Open Directory Project, which is a directory which anybody can download and use in whole or part. The ODP (also known as DMOZ) is the largest internet directory, almost entirely ran by a group of volunteer editors. The Open Directory Project was grown out of frustration webmasters faced waiting to be included in the Yahoo! Directory.

These newer web directories kept a hierarchy of the web pages based on their topics. Web directories are human-edited, thus making them very hard to maintain when the web is growing up so fast. The first search engine created was Archie, created in 1990 by Alan Emtage, a student at McGill University in Montreal. Those web search engines did not keep information about the content of the web pages; instead, they only indexed information about the title of the pages. It was in 1994, when web search engines started to index the whole web content, so that the user could search into the content of the web pages, not only in the title.

In 1998, Google appeared and this changed everything. The searches done by this search engine got better results than the previous search engines would get. This new search engine considered the link structure of the web, not only its contents. The algorithm used to analyze the link structure of the web was called Page Rank. This algorithm introduced the concept of "citation" into the web: the more citations a web page has, the more important it is. The information about the citations was taken from links in the web pages.

On June 1, 2009, Microsoft launched Bing, a new search service which changed the search landscape by placing inline search suggestions for related searches directly in the

result set. For instance, when you search for *Hollywood* they will suggest related phrases like.

- Hollywood movie
- Hollywood video
- Hollywood songs
- Hollywood movie download
- Hollywood movies in Hindi
- Hollywood wallpaper
- Hollywood movies free download

Microsoft released a Bing SEO guide which claimed that the additional keyword suggestions helped pull down search demand to lower listed results when compared against the old results 6 through 10 when using a single linear search result set.

Nowadays, web search engines are widely used, and their usage is still growing. Web search engines are today used by everyone with access to computers, and those people have very different interests. The table given below shows Google annual search statistics [3].

Year	**Annual Number of Google Searches**	**Average Searches Per Day**
2011	1,722,071,000,000	4,717,000,000
2010	1,324,670,000,000	3,627,000,000
2009	953,700,000,000	2,610,000,000
2008	637,200,000,000	1,745,000,000
2007	438,000,000,000	1,200,000,000
2000	22,000,000,000	60,000,000
1998	3,600,000	9,800

Table 1.1 Google Annual Search Statistics

Although the Web search has become the primary means by which people find and access information on the Web a very little support is provided for the tasks of crafting and refining queries. The goal of this research is to explore methods by which query expansion using additional personalized query term suggestion can be used to support Web searchers.

1.2 Motivation

While it is clear that significant effort has gone into creating Web search engines that can index billions of documents and return the search results in fractions of a second [4,5], the methods by which users craft queries, and the techniques used to present the search results to the users have remained essentially unchanged since the early days of Web search. Studies of Web search user behavior have shown that a large portion of Web search queries consist of only one to three terms [6, 7]. These short queries provide an indication that users of Web search engines often have difficulties crafting queries that accurately reflect their information needs. Clearly, most Web search engines provide little support for users as they attempt to construct and reformulate their queries; it is up to the user to decide which terms to use (both initially and during query reformulation), and manually add or remove these terms from the query. As a result of this lack of support, Web searchers seldom make subsequent modifications to their queries [8,7]. Even if the users are able to effectively craft a query, a high probability is there that the terms used to craft the query may not be the same in the relevant document. As per result the query will not be able to find relevant documents or they may be ranked lower in search result. Spink et al. noted that "the public has a low tolerance of going in depth through what is retrieved" [7].

Information search is a complex process consisting of the four main steps: problem identification, need articulation, query formulation, and results evaluation [9]. When a person uses World Wide Web through search engine then it may be a general case that he/she may not be familiar with query format notions or there a may be a term mismatch between the user query and the term used in relevant documents. This may lead to inefficient retrieval results. An expert may be trained enough to form a query while a

general web users don't. Furthermore, they may not have any special interest in such training. Belkin [10] has nicely illustrated the challenge the users face in text-based information retrieval: "How to guess what words to use for the query that will adequately represent the person's problem and be the same as those used by the system in its representation." So to make user query more expressive some more terms may be added or removed on the basis of its importance known as query expansion. These terms can be extracted from user-context relevant resources. Finding thematically rich terms can be seen as an optimizing problem. Searching space, sub-optimal solution etc. makes genetic algorithm the best suited tool as per the problem characteristics.

1.3 Aim of the Dissertation

The goal of this research is to address the fundamental issues related to these shortcomings of the current Web search engines by establishing a new paradigm for interactive query formation by suggesting terms for query expansion. These terms will not only help the user to formulate his problem but also to retrieve effective and relevant results as all these terms are thematically rich. This research acknowledges that human decision making is fundamentally important as users attempt to craft queries that capture their information needs, and as they attempt to locate documents that are relevant and useful. The development of next-generation Web search engines (of which this research is a part) will have a significant impact on how people will search for and find information in the future. In order to help the user to craft the queries, we have used GA for finding query expansion terms.

Overall aims of for this research work is given as follows:

1. We provide a method to deduce a set of related Categories for each user query based on the retrieval history of the user.
2. To design a system of user modeling that is used to identify a user's interests and preferences based on its browsing history.
3. To find thematically rich good query terms for query expansion in each predefined category using GA.

4. To help the user to craft the query by suggesting personalized inline query expansion terms.

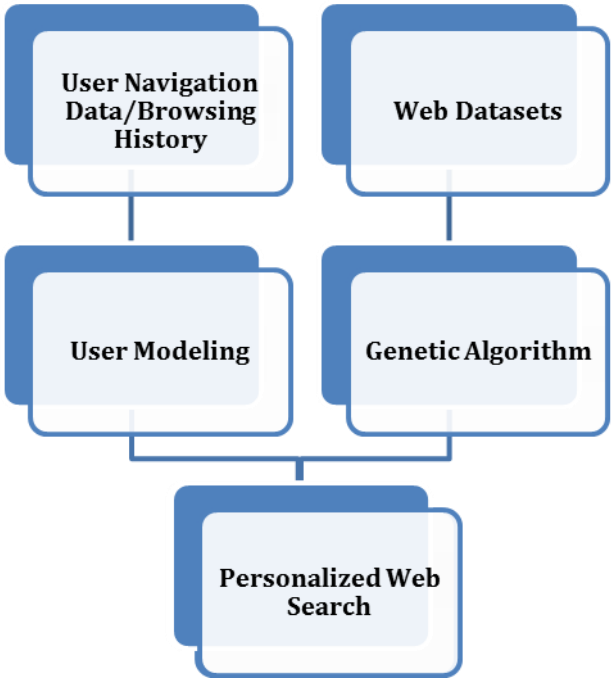

Figure 1.1 Overall Architecture of Research

1.4 Outline of the Dissertation

This dissertation is organized as follows. In Chapter 2 we have discussed Evolutionary Techniques (EA), Information Retrieval System (IRS), EA's appropriateness for Information Retrieval (IR) and thereafter EA's applications in IR. In Chapter 3 Personalized Web Search (PWS), its foundation, background, challenges and literature survey have been reviewed. How to build a user's interest and preference models based on the user's navigational data is discussed in this chapter. Proposed Personalized Web Search has been discussed in chapter 4 along with the Experiments and Results. In chapter 5 we have concluded with pointing future work.

CHAPTER 2

INFORMATION RETRIEVAL AND EVOLUTIONARY TECHNIQUES

2.1 Introduction

We are actually living in the information age. Most of the knowledge intensive organizations are having their information in large databases and text repositories. Unfortunately the large size of these databases has made the required effort to retrieve useful information increase significantly in the last few years. Information retrieval tries to make a suitable use of databases allowing the user to access the information which is really relevant in an appropriate time interval [11]. The goal of an information retrieval system is to satisfy user needs. Information retrieval (IR) is a complete process consisting of four main steps: problem identification, need articulation, query formation and result evaluation [12].

One of the computational intelligence areas with a considerable growth in the last decade is evolutionary computation. Different models have been proposed in evolutionary computations are named in the generic way as evolutionary algorithms (EA) [13]. In the IR field also researchers have worked with EA to improve efficiency of the IRS.

This chapter is structured as follows. In section 2.2 and 2.3 we review the basics of IRS and EA`s respectively. Further in section 2.4 we justify the appropriateness of EA`s along with analyzing the different kind of IR problems that have been solved using EA`s.

2.2 Information Retrieval Systems

Information retrieval is a field of study that helps the user to find needed information from a collection of large documents. Retrieving information simply means finding a set

of documents that is relevant to the user query. A ranking of these documents is also performed according to their relevance scores to the query. The user with an information need issues a query to the retrieval system through the query operational module. IRS deals with documentary bases containing textual, pictorial or vocal information and processes user queries trying to allow the users to access the relevant information in an appropriate time interval.

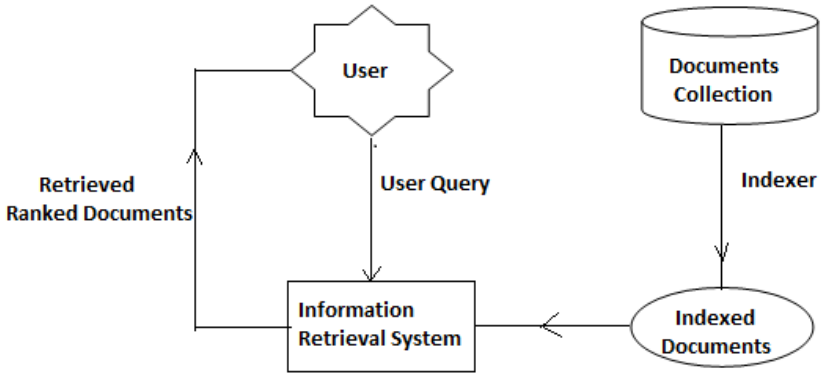

Figure 2.1 Basic Information Retrieval System

2.2.1 Components of IRS

An IRS consists of three basic components: Documentary Database, Query Subsystem, Matching mechanism [14].

1) ***The documentary database:*** This document database stores document along with the representation of their information content. It is associated with the indexer module which automatically generates a representation of each document by extracting the document contents.
2) ***The Query Subsystem:*** It allows the user to specify their information needs and presents the relevant documents retrieved by the system to them. The efficiency of an IRS system significantly depends upon the query formation.

3) ***The Matching Mechanism:*** It evaluates the degree to which documents are relevant to user query giving a retrieval status value (RSV) for each document. The relevant document is ranked on the basis of this value.

2.2.2 Information Retrieval Models

Information retrieval models govern how a document and a query are represented and how the relevance of a document to the user query is defined. The main IR models are [12]:

1) ***Boolean Model:*** The Boolean model is one of the earliest and simplest information retrieval model. A document is represented on the basis of indexed terms using a binary indexing technique. It uses the notion of exact matching to match document to the query.
2) ***Vector Space Model:*** VSM is perhaps the most widely used and well known model. A document in VSM is represented as a weight vector of terms and each term weight is computed based on some variation of TF or TF-IDF (Term Frequency-Inverse Document Frequency) scheme. The query is also represented in the same way as of documents.
3) ***Statistical Language Models:*** Statistical language models are based on probability and have some foundation in statistical theory. These probabilistic models compute the similarity coefficient between a query and a document as the probability that the document will be relevant to the query. There are two fundamental approaches: first relies on usage pattern to predict relevance and the other uses each term in the query as clues to whether or not a document is relevant [15].

2.2.3 Evaluation of IRS

1) ***Precision:*** Precision is a fraction of documents that are relevant among all the retrieved document.
2) ***Recall:*** Recall is a fraction of the documents that are retrieved and relevant among all relevant documents.

3) ***Precision-Recall Curve:*** This curve is based upon the value of precision and recall where the x-axis is recall and y-axis is precision. Instead of using precision and recall on at each rank position , the curve is commonly plotted using 11 standard recall level 0%, 10%, 20% ……..100%.

4) ***F-score:*** f-score is harmonic mean of precision and recall.

2.3 Evolutionary Algorithms

The field of search and optimization has changed over the last few years by the introduction of a number of non-classical and stochastic search and optimization algorithms. Of these, evolutionary algorithms use nature`s evolutionary principles to derive the search towards an optimal solution. Following well defined EA`s have served as the basis for much of the activity in the fields:

2.3.1 Genetic Algorithms

Genetic algorithms (1975) have been developed by John Holland, his colleagues and his students at the University of Michigan. GA`s are robust optimization techniques based on the principal of natural selection and survival of the fittest which claims "in each generation the stronger individual survive and weaker dies". To produce the new generation GA`s typically use selection together with genetic operators crossover and mutation. The fundamentally chromosome population is considered to be binary in nature.

1) ***Reproduction:*** also known as selection operator. Roulette wheel selection is among the well known methods used for selecting candidate chromosomes to crossover. In roulette wheel selection a chromosome is selected with the probability proportional to its fitness.

2) ***Cross Over:*** Once the reproduction is over the population is enriched with better individuals. Crossover is applied to the mating pool with a hope that it would create a better chromosome (solution).

3) ***Mutation**:* After crossover the chromosomes are subjected to mutation. Mutation of a bit involves flipping it, changing 0 to 1 and vice-versa with a small mutation probability.

The fundamental theory of genetic algorithm says that high-performance, short defining length low order schemata receive at least exponential increasing numbers of trials in successive generations [16]. This occurs because reallocation allocates more copies to the best schemata and because simple crossover does not disturb short-defining-length schemata with high frequency. Since mutation is fairly infrequent, it has little effect on these important schemata.

2.3.2 Multi-Objective Genetic Algorithm

When an optimization problem involves more than one objective, the task of finding one or more optimum solution is known as multi-objective optimization. Most real-world search and optimization problem naturally involves multiple objectives. So the simple GA mentioned above can`t be applied to such problems. Fonseca and Fleming (1993) first introduced Multi-Objective Genetic Algorithm (MOGA) which used the non-dominated classification of a GA population [17]. They also introduced niche among the solution of each rank. A MOGA is different from GA in the way fitness is assigned to each solution in the population.

First each solution is checked for its domination in the population. To a solution i, a rank equal to one plus the number of solutions n(i) that dominate solution i is assigned.

$$R(i) = 1+n(i) \qquad \text{……Equation (2.1)}$$

In this way non-dominated solutions are assigned a rank equal to 1 since no solution would dominate a non-dominated solution in the population. Once the ranking is performed, a raw fitness to a solution is assigned based on its rank. To perform this, first the ranks are sorted in ascending order of magnitude. Then the raw fitness is assigned to each solution of the linear (or any other) mapping function. Thereafter solution of each

rank is considered at a time and their raw fitness is averaged. The average fitness is called the assigned fitness to each solution in the rank.

In this way, non-dominated solutions are emphasized in a solution. In order to maintain the diversity among non-dominated solutions, Fonseca and Fleming have introduced niching among solution of each rank. Afterward a shared fitness value is calculated by dividing the fitness of a solution by its niche count. Then these fitness values are scaled and the whole procedure is continued until all ranks are processed. Thereafter the stochastic universal selection, the single point crossover, and the bitwise mutation operators are applied to create a new population.

2.4 Role of Evolutionary Algorithms for Solving IR Problems

The next two subsections deals with the justification of appropriateness of EA`s for information retrieval and its applications in IR.

2.4.1 EA`s Appropriateness for Information Retrieval

In classical methods of search and optimization we move from a single point in the decision space to the next point. It leads to the high possibility of locating false peaks in multi-model search space. While in case of EA`s, it climbs many peaks in parallel. Thus the probability of finding false peak reduced over other methods. Evolutionary algorithms are largely unconstrained (continuity, differentiability etc.) by limitation of many classical methods. Another advantage of using EA`s over other methods is that EA`s use probabilistic transition rules to guide their search. Following subsection discuss the appropriateness of EA`s for handling IR problems.

1) Information Retrieval as an Optimization Problem

We can see information retrieval as an optimization problem. The objective function to be optimized is based on the effectiveness of a query to retrieve relevant material when presents to a search engine. Depending upon the system goals a measure of query effectiveness can be defined using traditional IR notions such as precision and recall or other customized performance evaluation parameter.

2) **High Dimension Space**

In case of IR search space is of high dimension. And EA`s can naturally deal with such solution space rather than analytical methods.

3) **Multiple Solutions**

The ability of EA`s to find multiple optimal solutions in one single simulation run makes EA`s unique in solving multi-objective optimization problem. For example each one of the multiple sets of web pages can represent a satisfactory result. Therefore we may be interested to find more than a single one.

4) **Exploration and Exploitation**

Finding a good solution in IR requires exploration and exploitation in each direction of search space. Operators like crossover and mutation perform such operations really well.

2.4.2 EA`s Applications in IR

EA`s are vastly applicable in various domains. Here are some fields in IR where EA`s have been used successfully.

1) Document Clustering

Document clustering refers to put the similar documents in a group with the purpose to minimize the intra-cluster similarity and maximizing the inter-cluster similarity. In [18] Gordan designed a philosophy according to which it is possible to make a user oriented clustering of documents using any classical clustering technique. The basic idea is that the system adapts the document description throughout the time. Gordan proposes GA to derive document descriptions. He chooses a binary coding scheme where each description is a fixed length binary coded vector and the genetic population is composed of different description of the same document. Robertson and Willet`s [19] has proposed the idea which was based on that "look for group of terms appearing with similar frequencies in the documents of a collection. To do so, the author considered a GA grouping of terms without maintaining their initial order.

2) Automatic Document Indexing

Indexing is basically a data structure that's holds the key terms that represent a document. Taking a user query as input, IRS searches it through the indexes to find relevant documents. Gordon was the first to use genetic algorithm for document indexing. He has proposed the idea to attach more than one description with each document and then let them adapt throughout time as a good solution to the problem of the different forms that different user queries searching for the same document can present [15].

In [21] Vrajitoru presented a different approach to the same problem. In which each document is associated with just one description which leads to encode the whole collection in the single chromosome. The problem with this model is that the fitness function considers only one query and not the set of queries as the Gordon`s model. Fan et al. [22] proposed an algorithm for indexing function learning based on Genetic Programming, whose aim is to obtain an indexing function for the key term weighting for a documentary collection to improve the IR process.

3) Matching Function Adaption

The aim of using EA is to generate a similarity measure for a vector space IRS to improve its retrieval efficacy for a specific user. This constitutes a new relevance feedback philosophy since the matching function is adopted instead of queries. In [22] Pathak et al. have proposed the idea of combined similarity measure in which they have proposed a linear combination of various similarity measures and then optimize the weight of each similarity measure using GA.

A GP algorithm to automatically learn a matching function with relevance feedback is introduced in [23]. The similarity functions are represented as trees, and a classical generational scheme and the usual GP crossover are considered.

4) Query Optimization

There are mainly two methodologies that have been proposed for query optimization in a broad sense. One is based on relevance feedback techniques to modify the original query

by adding some other relevant terms [24]. And the other is based on IQBE (Inductive Query by Example), which is a process in which a searcher provide sample documents (examples) and the algorithm induce the key concept in order to find the other relevant documents. In [25], Chen et al. used a GA as an IQBE technique to learn query terms that better represent a relevant document set provided by the user. In [26] the author has used a GA to adapt the query term weights in order to get the closest query vector to the optimal one.

5) Context-based Search

Most of the IRS are not able to capture the context of user queries. Therefore they are unable to return the relevant documents fulfilling user`s needs. Finding the context of a user`s query itself a challenging research issue. Various attempts have been made to include the user context in IR. The system like WebWatcher [27] uses contextual information compiled from past browsing behavior search within the locus of currently viewed pages. Another system like CALVIN [28] is a context aware system that monitors the user`s web browsing activity to generate a model of user task to use it for retrieval of relevant resources indexed in similar context.

Once we are clear about user context then there are ways to deal with this problem. One is to redirect the user to a search engine specific to user context. And another approach is query reformulation means to add some new terms to make user query more expressive. The latter approach is having the advantages of huge data and functionalities of conventional search engines.

CHAPTER 3

PERSONALIZED WEB SEARCH

3.1 Definition

Personalized Web Search refers to search experiences that are tailored specifically to an individual's interests by incorporating information about the individual beyond specific query provided [65].For a given query, a personalized Web search can provide different search results for different users or organize search results differently for each user, based upon their interests, preferences, and information needs. Personalized web search differs from generic web search, which returns identical research results at all, regardless of varied user interests and information needs [29].

3.2 Historical Background

Web search engines have made enormous contributions to the web and society. They make finding information on the web quick and easy. However, they are far from optimal. A major deficiency of generic search engines is that they follow the ‘‘one size fits all’’ model and are not adaptable to individual users. This is typically shown in cases such as these:

1. Different users have different backgrounds and interests. They may have completely different information needs and goals when providing exactly the same query. For example, a biologist may issue ‘‘mouse’’ to get information about rodents, while programmers may use the same query to find information about computer peripherals. When such a query is issued, generic search engines will return a list of documents on different topics. It takes time for a user to choose which information he/she really wants, and this makes the user feel less satisfied. Queries like ‘‘mouse’’ are usually called ambiguous queries. Statistics have shown that the vast majority of queries are short and ambiguous. The generic web search usually fails to provide optimal results for ambiguous queries.

2. Users are not static. User information needs may change over time. Indeed, users will have different needs at different times based on current circumstances. For example, a user may use ‘‘apple’’ to find information about apple as a fruit when the user is looking for beverages recipes, but the same user would want to find information about apple computer products when purchasing a new computer. Generic search engines are unable to distinguish between such cases.

Personalized web search is considered a promising solution to address these problems, since it can provide different search results based upon the preferences and information needs of users. It exploits the user information and the search context in learning to which sense a query refers. Consider the query ‘‘mouse’’ mentioned above: Personalized web search can disambiguate the query by gathering the following user information:

- The user is a computer programmer, not a biologist.
- The user has just input a query ‘‘keyboard,’’ but not ‘‘biology’’ or ‘‘genome.’’ Before entering this query, the user had just viewed a web page with many words related to the computer mouse, such as ‘‘computing,’’ ‘‘input device,’’ and ‘‘keyboard.’’

3.3 Foundation

In general personalized search engines framework consists of user modeling based on user past browsing history or application he/she is using etc. And then use this context to make the web search experience more personalize. In the following section we have discussed different approaches to create a user profile to represent user preferences. Thereafter we have reviewed different strategies to personalize a web search based on user preference profiles along with their evaluation.

3.3.1 User Modeling

In personalized online services based on a user’s model , the content similarity between the user’s profile and the products (Web pages) or services are analyzed and used to

provide relevant products that satisfy the user's needs. In many personalized Web systems, a user's profile or model implies the user's attributes of interest and preference. The process of creating a user's profile or models is called user modeling. Different systems may have different structures of user models, since they have their own purposes, interfaces, user groups, etc. For example, a digital library may have a structure of user models consisting of gender, age, and education level, while an online shop needs to model a user's age, location, and purchase interests. Here we will discuss user profiling with respect to web search.

In order to collect the user information and create user models/profiles, there are two approaches that are widely used: the explicit approach and the implicit approach. For the purpose of user modeling, an explicit approach is an approach to obtain a user's model/profile through direct queries or surveys. However, using the explicit approach to create a user profile has several limitations: the profile cannot update itself when the user's interests and preferences change over time; a user may be tired of answering inquiries during the construction of a profile.

Any change in the structure of a profile needs the user's participation and many users are unwilling to participate in the time-consuming process of constructing user interest and preference models. Implicit approaches are increasingly applied in personalized online services to automatically build user models without interrupting Web user's navigation. The implicit approach to user modeling is a process to create and update user models by collecting and analyzing user data. A significant difference between explicit and implicit approaches is that implicit approaches need to calculate the collected user data for user modeling, while explicit approaches directly fit the collected user data into user models. Unlike explicit approaches which adopt the category (topic) structure for user profiles, implicit approaches can use either the structure of the category or the structure of bags of words to create user profiles.

In [30] Nicolaas and fillip have used the users' long term browsing history for personalizing web search. They have represented the user by a list of terms and weights associated with those terms, a list of visited URLs and a number of visits to each, and a

list of past search queries and the pages clicked for these queries. Next, this browsing history is processed into different summaries and finally the term weights are generated using different weighting algorithms.

First to capture the user browsing histories, a Firefox add-on called Alter Ego was used. To respect the users' privacy as much as possible, a random unique identifier is generated at installation time. The identifier is used for all data exchanges between the add-on and the server recording the data.

They have considered the following summaries of the content viewed by users in building the user profile:

Full Text Unigrams
The body text of each web page stripped of HTML tags.

Title Unigrams
The words inside any <title> tag on the html pages.

Metadata Description Unigrams
The content inside any <meta name=\description"> tag.

Metadata Keywords Unigrams
The content inside any <meta name=\keywords"> tag.

Extracted Terms
They have implemented the Term Extraction algorithm as presented in [30], running it on the full text of each visited web page. It attempts to summarize the web page's text into a set of important keywords.

Noun Phrases
Noun phrases were extracted by taking the text of each web page and splitting it into sentences using a sentence splitter from the OpenNLP Tools.

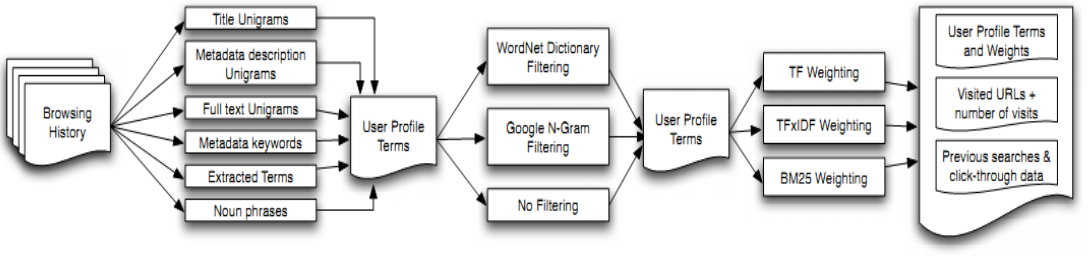

Figure 3.1 Methodology Used for User Profiling [30]

To reduce the number of noisy terms of user representation, they also tried filtering terms by removing infrequent words or words not in WordNet. However, neither of these was found to be beneficial. After the list of terms has been obtained then the weight has been computed for each of them. Any of following weighting strategies can be used to assign the weight.

TF Weighting

The most straightforward implementation considered is Term Frequency (TF) weighting. They used a frequency vector F that contains the frequency counts of a given term t for all of the input data sources, as shown in Equation. For example, ftitle is the number of times a given term t occurs in all of the titles in the user's browsing history. We calculate a term weight based on the dot product of these frequencies with a weight vector:

$$\vec{F}_{t_i} = \begin{bmatrix} f_{title_{t_i}} \\ f_{mdesc_{t_i}} \\ f_{text_{t_i}} \\ f_{mkeyw_{t_i}} \\ f_{terms_{t_i}} \\ f_{nphrases_{t_i}} \end{bmatrix}$$

.....Equation (3.1)

$$w_{TF}(t_i) = \vec{F}_{t_i} \cdot \vec{\alpha}$$

..... Equation (3.2)

For simplicity, we limit ourselves to three possible values for each weight i: 0, ignoring the particular field, 1, including the particular field, and 1/ Ni, where Ni is the total number of terms in the field i. This gives more weight to the terms in shorter fields.

TF-IDF Weighting

The second option can be taken into account is TF-IDF (or Term Frequency, Inverse Document Frequency) weighting. Here, words appearing in many documents are down-weighted by the inverse document frequency of the term:

$$w_{TFIDF}(t_i) = \frac{1}{log(DF_{t_i})} \times w_{TF}(t_i)$$

.....Equation (3.3)

Personalized BM25 Weighting

Another weight method can be considered was proposed by Teevan et al. [4], which is a modification to BM25 term weighting:

$$w_{pBM25}(t_i) = log\frac{(r_{t_i} + 0.5)(N - n_{t_i} + 0.5)}{(n_{t_i} + 0.5)(R - r_{t_i} + 0.5)}$$

.....Equation (3.4)

Where N represents the number of documents on the web (estimated from the Google N-Gram corpus, 220,680,773), n_{ti} is the number of documents in the corpus that contain the term t_i (estimated using the Google N-Gram corpus), R is the number of documents in the user's browsing history and r_{ti} is the number of documents in the browsing history that contains this term within the selected input data source.

In [32] fang liu , Clement Yu and Weiyi Meng have used mapping user query to categories for personalized web search. One way to associate a category with the user query is to let the user select one or more categories in the hierarchy before submitting his/her query. But unfortunately, the category hierarchy shown to the user is very large and as a result an ordinary user may have difficulty in finding the proper path leading to the suitable categories. So in this paper the author has proposed how to supply, for each

user, a small set of categories as a context for each query submitted by the user based on his/her search history. In this paper a user profile consists of a set of categories and for each category, a set of terms (keywords) with weights. Each category represents a user interest in that category. The weight of a term in a category reflects the significance of the term in representing the user's interest in that category. For example, if the term "apple" has a high weight in the category "cooking", then the occurrence of the word "apple" in a future query of the user has a tendency to indicate that the category "cooking" is of interest.

The Author has used matrices to represent user search histories and user profiles. It shows an example of the matrix representations of a search history and a profile for a particular user, who is interested in the categories "COOKING" and "SOCCER". This user's search history is represented by two matrices DT and DC. DT is a document-term matrix, which is constructed from the user queries and the relevant documents. DC is a document-category matrix, which is constructed from the relationships between the categories and the documents. A user profile is represented by a category-term matrix M. In this example, D1, D2 … are documents; lowercase words such as "football", "apple"… are terms; uppercase words such as "SOCCER", "COOKING" … are categories.

Doc\Term	**Apple**	**Recipe**	**Pudding**	**Football**	**Soccer**	**Fifa**
D 1	1	0	0	0	0	0
D2	0.58	0.58	0.58	0	0	0
D3	0	0	0	1	1	0
D4	0	0	0	0.58	0.58	0.58

Table 3.1 Document- Term matrix DT [32]

Doc\Category	Cooking	Soccer
D 1	1	0
D2	1	0
D3	0	1
D4	0	1

Table 3.2 Document- Category matrix DC [32]

Cate\Term	Apple	Recipe	Pudding	Football	Soccer	Fifa
COOKING	1	0.37	0.37	0	0	0
SOCCER	0	0	0	1	0.37	0.37

Table 3.3 Category- Term matrix M represents a user profile [32]

Another approach [33] given by feng qiu and junghoo cho uses topic preference vector of a user from his/her past browsing history. Given the billions of pages available on the web and their diverse subject areas, it is reasonable to assume that an average web user is interested in a limited subset of web pages .In addition, we often observe that a user typically has a small number of topics that she/he is primarily interested in and her/his preference to a page is often affected by her general interest in the topic of the page. For example, a physicist who is mainly interested in topics such as science may find a page on video games not very interesting, even if the page is considered to be of high quality by a video-game enthusiast. Given these observations, user's preference can be represented at the granularity of either topics or individual web pages.

Topic Preference Vector

A user's topic preference vector is defined as a m -tuple T = [T (1), . . . , T (m)],in which m is the number of topics in consideration and T (i) represents the user's degree of interest in the i^{th} topic(say, "Computers "). Example: Suppose there are only two topics : "Computers " and "News ," and a user is interested in "Computes "three times as much as she is interested in "News ," then the topic preference vector of the user is [0.75, 0.25]. Instead of the above, we may represent a user's interest at the level of web pages.

Page Preference Vector

A user's page preference vector is defined as a n-tuple P = [P (1), . . . , P (n)], in which n is the total number of web pages and P (i) represents the user's degree of interest in the i^{th} page.

In principle, the page preference vector may capture a user's interest better than the topic preference vector, because his/her interest is represented in more detail. But given a billion of pages available on the web, a user can click on only a small fraction of them, making the task of learning the page preference vector very difficult. Due to this practical reason, the author has used the topic preference vector as the presentation of user interest.

3.3.2 Personalized Search Strategies accommodating User-context

Once we are done with representing user profile then we use different approaches to personalize a web search based on these preference profiles. Following are the different approaches used recently for this purpose:-

1) Towards Context-based Search Engine Selection

While conventional search engines serve the population as a whole, specialized search resources, on the other hand, can provide coverage that is pre-focused. So an index of specialized search engines can be created and marked along with their specialized categories. Thus once we are able to detect the user or topic preference we can select the context relevant specialized search engine. For example if the interface can determine that the user is working on the paper in economics, it could use the information to

generate the context description. And select the context-relevant specialized search engine such as CNN financial. In [34] David B. Leake and Ryan Scherle described research on the source selection problem in the PRISM system. PRISM source selection approach relates to search on both distributed searching and "just in time" searching. Its central claim is that distributed searching can be more effective if it is guided by contextual information.

2) Re-ranking Strategies

Here user profile can be used to re-rank the top results returned by a search engine to bring up results that are more relevant to the user. This allows us to take advantage of the data search engines use to obtain their initial ranking, by starting with a small set of results that can then be personalized. In particular [35] noted that the chances are high that even for an ambiguous query the search engine will be quite successful in returning pages for the different meanings of the query.

Scoring Methods

When re-ranking, each candidate document can either be scored, or just the snippets shown on the search engine result page can be scored. Assigning scores to the search snippets as it was found to be more effective for re-ranking search results by Teevan et al. [35].

a) Matching

For each word in the search snippet's title and summary that is also in the user's profile, the weight associated with that term will be added to the snippet's score:

$$score_M(s_i) = \sum_{z=1}^{N_{s_i}} f_{t_z} \times w(t_z)$$

.....Equation (3.5)

Where N_{si} represents the total number of unique words within the snippet's title and summary, and f_{ti} represents the number of occurrences of t_i within the snippet. Words in the snippet title or summary but not in the user's profile do not contribute towards the

final score. This method is equivalent to taking the dot product between the user profile vector and the snippet vector.

b) Unique Matching

A second search snippet scoring option we consider involves counting each unique word just once:

$$score_{UM}(s_i) = \sum_{z=1}^{N_{s_i}} w(t_z)$$

.....Equation (3.6)

c) Language Model

The third score calculation method generates a unigram language model from the user profile in which the weights associated with the terms are used as the frequency counts for the language model:

$$score_{LM}(s_i) = \sum_{z=0}^{N_{s_i}} \log\left(\frac{w(t_z)+1}{w_{total}}\right)$$

.....Equation (3.7)

Where N_{si} is the total number of words in the snippet's title and summary, and w_{total} stands for the sum of all the weights within the user profile. The language model estimates the probability of a snippet given a user's profile.

d) PClick

As a final snippet scoring method, PClick algorithm was proposed by Dou et al. [35]. It assumes that for a query q submitted by a user u, the web pages frequently clicked by u in the past are more relevant to u. The personalized score for a snippet is:

$$score_{PC}(s_i) = \frac{|Clicks(q,p,u)|}{|Clicks(q,\bullet,u)|+\beta}$$

……....Equation (3.8)

Where Clicks (q; p; u) were the number of clicks on web page p by user u for query q in the past,

Clicks (q; • ; u) is the total click number on query q by u, and is a smoothing factor set to 0.5. Note that PClick makes no use of the terms and weights associated with the user's profile and are solely based on click-through data for a given query. As such, it only affects repeated queries.

3) Topic Sensitive Page Rank

The topic sensitive Page Rank scheme (TSPR) proposed in [36] is an interesting extension of PageRank that can potentially provide a different ranking for different queries, while essentially retaining the efficiency advantage of the standard PageRank. In the TSPR scheme, multiple scores, instead of just one, are computed for each page, one for each topic that one considers. Assuming that we consider m topics, m TSPR scores are computed for each page, which can be done offline. Then during online query processing, given a query, the search engines figures out the most appropriate TSPR score and use it to rank pages.

4) Query Expansion

This can be another approach for personalized web search. Some more terms can be added to the user query to make the context of the user clearer, with the goal of building a query that more accurately captures the user's information need. Query expansion is a well studied technique for improving information retrieval performance by improving the user's input.

With respect to the work that needs to be performed by the user, there are three options: manual query expansion, automatic query expansion, and interactive query expansion. Manual query expansion techniques are those which require the user to do the work of evaluating, selecting, and adding new terms to their query (i.e., without any additional system support). Automatic query expansion techniques choose and add new terms to the user's query, without the involvement of the user other than to submit an initial query.

Interactive query expansion techniques allow the user to interactively make choices which are then used to generate the query expansion.

	Automatic Query Expansion	Interactive Query Expansion
Harman, 1988 [30]		√
Salton & Buckley, 1990 [86]		√
Harman, 1992 [31]	√	
Qiu & Frei, 1993 [77]	√	
Voorhees, 1994 [108]	√	
Chang & Hsu, 1999 [15]		√
Mandala et al., 1999 [58]	√	
Xu & Croft, 2000 [114]	√	
Billerbeck et al., 2003 [11]	√	

Table 3.4 Classification of Query Expansion Based upon Work Performed by User

This methodology of interactive query expansion will be discussed in detail in our proposed framework. Much work has not been done on personalized web search using interactive query expansion using an evolutionary approach till now.

3.3.3 Evaluation Approach

Now let's consider potential evaluations for personalized search strategies:

1) Relevance Judgments

The first possible online evaluation approach (e.g. used by Teevan et al. [35]) is based on assembling a group of people that judge the relevance of the top k documents or search snippets for a set of queries. This approach has the advantage that once the relevance judgments are made, it allows for testing many different user profiles and re-ranking parameter configurations. However, due to the long time it takes to judge k documents, this can only be done for a small number of search queries. As volunteers need to be

found to sit through this slow and tedious evaluation process, it is also hard to gather a large group of evaluators.

2) Side-by-side evaluation

An alternative offline evaluation method, previously used for example by [37], consists of presenting users with two alternative rankings side-by-side and asking which they consider best. Judging two rankings next to each other is considerably faster than judging k documents per query, but it still requires a long offline evaluation exercise.

3) Click-through based evaluation

One common online evaluation approach involves looking at the query and click logs from a large search engine (e.g. used by [38]). The logs record which search results were clicked for each query, thus allowing the evaluator to check if the clicked result would be positioned higher in a personalized ranking. However, the method can have difficulties in assessing whether a search personalization strategy actually works. First, users are more likely to click a search result presented at a high rank, although these are not necessarily most or more relevant [39]. It is also unsuccessful in assessing whether lower results would have been clicked had they been shown at a higher rank. Finally, it's difficult to get access to such large scale usage and user profile data for this experiment.

4) Interleaved Evaluation:

Interleaved evaluation combines the results of two search rankings by alternating between results from the two rankings while omitting duplicates and the user is presented with this interleaved ranking [30]. The ranking that contributed the most clicks over many queries and users is considered better.

3.4 Challenges of Personalized Web Search

Despite the attractiveness of personalized search, there is no large-scale use of personalized search services currently. Personalized web search faces several challenges that retard its real-world large-scale applications:

1) Privacy is an issue. Personalized web search, especially server-side implement, requires collecting and aggregating a lot of user information including query and Clickthrough history. A user profile can reveal a large amount of private user information, such as hobbies, vocation, income level, and political inclination, which is clearly a serious concern for users [40]. This could make many people nervous and feel afraid to use personalized search engines. A personalized web search will be not well- received until it handles the privacy problem well.
2) It is really hard to infer user information needs accurately. Users are not static. They may randomly search for something which they are not interested in. They even search for other people sometimes. User search histories inevitably contain noise that is irrelevant or even harmful to the current search. This may make personalization strategies unstable.
3) Queries should not be handled in the same manner with regard to personalization. Personalized search may have little effect on some queries. Some work [41,42,43] investigates whether current web search ranking might be sufficient for clear/unambiguous queries and thus personalization is unnecessary. Dou et al. [43] reveal that personalized search has little effect on queries with high user selection consistency. A specific personalized search also has a different effectiveness for different queries.

   It even hurts search accuracy under some situations. For example, topical interest-based personalization, which leads to better performance for the query ‘‘mouse,’’ is ineffective for the query ‘‘free mp3 download.’’ Actually, relevant documents for query ‘‘free mp3 download’’ are mostly classified into the same topic categories and topical interest-based personalization has no way to filter out desired documents. Dou et al. [43] also reveal that topical interest-based personalized search methods are difficult to deploy in a real world search engine. They improve search performance for some queries, but they may hurt search performance for additional queries.

3.5 Literature Survey

Nauman et. al., [44] used machine common sense in conjunction with folksonomy based intelligent search systems for personalized web search. A folksonomy is a system of classification derived from the practice and method of collaboratively creating and managing tags to annotate and categorize content [45]. This practice is also known as collaborative tagging, social classification and social tagging. A huge division of the contemporary web is characterized by user generated content classified using collaborative tagging or folksonomy. It makes very tricky to search for appropriate content because of ambiguity in lexical illustration of concepts and variances in preferences of users. With additional services relying on tags for content classification, it is significant that search approaches progress to better suit the scenario. A promising technique in avoiding these difficulties is to use machine common sense in combination with folksonomy. A past effort to use this technique has shown encouraging results in obtaining relevant content but it does not deal with the issue of noise in search results. In this paper, the authors make use of the personalized web search approach of conventional web demographic information, etc., and their web activities search systems to concentrate on the issue of irrelevant search outcomes in common sense and folksonomy dependent search systems. In this new technique, search results are tailored according to the user's preferences implied by his/her search and access history. Outcomes are reflective of user's favorites, which are based on the search history and the kind of interest shown by the user. This contribution has thus succeeded in developing an approach that effectively addresses polysemy - a major problem in folksonomy based services. This paper proposes alterations to personalized web search approach. Using this personalized approach, the authors extend the fundamental machine common sense and folksonomy dependent search systems to deal with the problem of noise in search results.

Zhengyu Zhu et al., [46] proposed personalized web search model based on query expansion. It depends on a representation of personalized web search organization. The proposed novel system, as a middleware connecting a user and a Web search engine, is fixed on the client machine. It can study the user's favorite implicitly and then produce the user profile automatically. User profiles are used to represent users' interests and infer

their intentions for new queries. In this paper, a user profile consists of a set of categories and, for each category, a set of terms with weights. Each category represents a user interest in that category. The weight of a term in a category reflects the significance of the term representing the user's interest in that category. When the user enters query keywords, more personalized expansion words are produced by the proposed approach, and then these words in common with the query keywords are forwarded to a famous search engine such as Baidu or Google. These expansion words can facilitate search engine retrieval information for a user based on his/her implicit search objectives.

P. Palleti et al., [47] developed personalized web search using probabilistic query expansions. In this paper, the author proposed a personalized web search system implemented at the proxy which adapts to user interests implicitly by constructing a user profile with the help of collaborative filtering. A user profile essentially contains probabilistic correlations between query terms and document terms which is used for providing personalized search results. In this approach, the proposed web search system applied at proxy which changes to user interests perfectly by generating user profile with the use of collaborative filtering. Experimental outcomes prove that this proposed personalized Web search system is very effective and efficient.

Jie Yu et al., [48] suggested mining user context based on interactive computing for personalized Web search. Previous approaches focus more on constructing a user profile which depends on Web pages/documents which influences the effectiveness of search engine. Additionally, dynamics of user profile are frequently ignored. In this paper, the author has taken query context as the basis for building user context. Query context can be regarded as the semantic background of user's search behavior. It extracts not only concepts from snippet but also the relationship between them, which ensures the generated user context representing user's real interest more accurately and effectively. By interactive computing, user's each click behavior is reflected by a user context snap. By updating the weights between concepts in query context, user context can reflect user's interest with the single click.

Fang Liu et al., [49] recommended personalized Web search for improving retrieval effectiveness. A user profile and a general profile are learned from the user's search

history and a category hierarchy respectively. These two profiles are combined to map a user query into a set of categories, which represent the user's search intention and serve as a context to disambiguate the words in the user's query. A web search is conducted based on both the user query and the set of categories. In this paper, a user profile consists of a set of categories and for each category, a set of terms (keywords) with weights. Each category represents a user interest in that category. The weight of a term in a category reflects the significance of the term in representing the user's interest in that category.

Xuwei Pan et al., [50] proposed context-based adaptive personalized Web Search for improving information retrieval effectiveness. In this approach, the authors proposed a novel adaptive personalized technique based on context to adapting search outputs consistent with each user's requirement in different situations for relevant information with slight user effort. Following to the process in the context-based adaptive personalized search investigation, three important technologies to execute this method are provided, which are semantic indexing for Web resources, modeling and obtaining user context and semantic resemblance matching among Web resources and user context.

Kyung -Joong Kim et al., [51] developed a personalized Web search engine using a fuzzy concept network with link structure. Most of the famous search engines make use of the link structure to discover precision result. Typically, a link- based search engine provides superior -quality outputs than a text -based search engine. On the other hand, they have a complexity in providing the result that satisfies the specific user's preference. Personalization is necessary to maintain a more suitable result. Among the many approaches, the fuzzy concept network according to a user profile can characterize a user's subjective interest appropriately. The paper proposes a search engine that utilizes the fuzzy concept network to personalize the outputs from a link - based search technique. Depending on a user profile, the fuzzy concept network rearranges five outputs of the link -based search engine, and the system presents a personalized superior quality result.

Chen Ding et al., [52] suggested personalized Web search with self- organizing map. With the intention of minimizing the consumption of time on browsing irrelevant

documents, this paper suggests an intelligent Personal Agent for Web Search (PAWS). The PAWS intelligently utilizes the Self-Organizing Map (SOM) as the user's profile. The system learns and builds the user profile based on the user's search history using a SOM. The PAWS deduces the category, which is likely to be interested by the user for the specific query from the user's profile. The PAWS combines the user's search keywords and the related category into new queries. Through it, the user can get the search answer set in some specific area. The PAWS makes it easier to find the information on the web for the user. It makes the web search more efficient for the user.

Biancalana et al., [53] proposed a new way for personalized Web search using social tagging in query expansion. Social networks and collaborative tagging systems are rapidly gaining popularity as a primary means for sorting and sharing data: users tag their bookmarks in order to simplify information dissemination and later lookup. Social Bookmarking services are useful in two important respects: first, they can allow an individual to remember the visited URLs, and second, tags can be made by the community to guide users towards valuable content. In this paper author focused on the latter use: they presented a novel approach for personalized web search using query expansion, and further extended the family of well-known co-occurrence matrix technique models by using a new way of exploring social tagging services. This approach shows its strength particularly in the case of disambiguation of word contexts. This paper shows steps to plan and execute such a system in practice and performed numerous experiments on a real web- dataset. This is the first study focused on the use of social bookmarking and tagging approaches for personalization of web search and its performance in a real-world application.

Personalized Web search with location preferences is recommended by K. W. -T. Leung e t al,[54]. In this paper, the authors recommended a novel web search personalization technique that recognizes the user's interests and preferences with the help of concepts by mining search outputs and their Clickthroughs. Due to the important role location information plays in mobile search, it separates concepts into content concepts and location concepts, and organizes them into Ontologies to create an ontology-based, multi-facet (OMF) profile to precisely capture the user's content and location interests and

hence improve the search accuracy. Moreover, recognizing the fact that different users and queries may have different emphases on the content and location information, we introduce the notion of content and location entropies to measure the amount of content and location information associated with a query, and click content and location entropies to measure how much the user is interested in the content and location information in the results. Accordingly, they proposed to define personalization effectiveness based on the entropies and use it to balance the weights between the content and location facets. Finally, based on the derived Ontologies and personalization effectiveness, they trained an SVM to adapt a personalized ranking function for re-ranking of future search. They conducted extensive experiments to compare the precision produced by their OMF profiles and that of a baseline method. Experimental results show that OMF improves the precision significantly compared to the baseline.

J. Lai et al.,[55] compared personalized Web search results with user profile. It is vital to evaluate users' search and browsing activities based on searching keywords inputted by users, the clicking rate of each link in the output and the time they used on each site. To this end, the authors have proposed a technique to obtain user searching profiles. This paper also proposed a method to obtain document profiles, according to the similarity score of documents. In this paper, the authors discussed how to utilize this model to integrate the user searching profiles and the document profile, with the intention of presenting personalized search results to the users. For the customer profile component, customer behaviors - such as the searching keywords or phrases, a particular document the customer browsed and the time the customer spent on that document - are deemed as variables in the customer profile component. An average (generalized) customer profile will be derived from the overall customers currently profiled. This average profile will be used as the starting point for a new customer before their personalized profile is built up. For the document profile component, the keywords, phrases and weight of the keywords are adopted for calculating the similarity score of documents. The document profile is then derived using keywords and similarity score as variables. The document profile will be updated upon the arrival of any new document on the web.

B. Smyth [56] proposed a community-based approach to personalizing Web search. Researchers can influence the underlying knowledge produced within search communities by gathering users' search behaviors - the queries they enter and the results they choose - at the community level. They can make use of this data to construct a relevance model that provides the promotion of community-relevant results throughout standard Web search. This paper focuses on the collaborative Web search technique that encourages the suggestion that community search behaviors can offer a valuable form of search knowledge and sharing of this knowledge makes adapting conventional search-engine outputs possible.

In this paper [57], O. Shafiq et al., has described a novel ranking technique for personalized search services that combines content-based and community-based evidences. The community-based information is used in order to provide context for queries and is influenced by the current interaction of the user with the service. This paper has presented and evaluated a novel ranking technique that uses the combination of these evidential sources of relevance: the content of the objects being retrieved and the interest-based community of the user issuing the search. The framework proposed is general enough to allow the use of any of the classical models for content-based information retrieval and of most of community identification algorithms, as long as summaries for the communities can be produced. In the experiments conducted in this paper they were able to provide an improvement of 48% in term of average precision and even better results if they consider the position occupied in the ranking by the relevant objects for each query.

Han -joon Kim et al., [58] suggested building a concept network -based user profile for personalized Web search. The user profile is defined as a concept network, in which each concept is approximately represented by the formal concept analysis (FCA) theory. It has assumed that a concept, called ‘session interest concept’, subsumes a user’s query intention during a query session and it can reflect the user’s preference. Whenever a user issues his/her query, a session interest concept is generated. Then, new concepts are merged into the current concept network (i.e., a user profile) in which recent user preferences are accumulated. According to FCA, a session interest concept is defined as a

pair of the extent and intent where the extent covers a set of documents selected by the user among the search results and the intent covers a set of keyword features extracted from the selected documents. And, in order to make a concept network grow, they have calculated the similarity between a new concept and existing concepts, and to this end, they have used a reference concept hierarchy called the Open Directory Project. The user profile of concept network is eventually used to expand a user's initial query. The experimental result proves that this approach increases the accuracy of search results based on the personal preference.

Yan Chen et al., [59] recommended a personalized context-dependent Web search agent using semantic trees. In Web searching applications, contexts and users' preferences are two significant features for Internet searches in some way that outputs would be much more appropriate to users' requests than with existing search engines. Researchers had planned a concept-based search agent which utilizes a conceptual fuzzy set (CFS) for matching contexts-dependent keywords and concepts. In the CFS model, a word accurate meaning may be determined by other words in contexts. Owing to the fact that various combinations of words may become visible in queries and documents, it may be complicated to identify the relations between concepts in all possible combinations. To avoid this problem, the authors proposed a semantic tree (ST) model to identify the relations between concepts. Concepts are symbolized as nodes in the ST, and relations connecting these concepts are represented by the distances between nodes. Furthermore, this paper makes use of the users' preferences for personalizing search results. Finally, the fuzzy logic will be utilized for finding which factor, semantic relations or users' preferences will control results.

Wen -Chih Peng et al., [60] proposed re-ranking Web search results from personalized perspective. In this paper, the authors develop the approach of mining common access patterns from user browsing activities and developed an approach to automatically obtain user interests. Additionally, according to the user interests mined and feedbacks of users, a new approach is proposed by the plan of dynamically altering the ranking scores of Web pages. In particular, algorithm PPR stands for Personalized Page Rank, is segmented into four stages. The first stage allots the initial weights according to user

interests. In the second stage, the virtual links and hubs are generated based on user interests. By examining user click streams; this proposed algorithm will incrementally reproduce user favors for the personalized ranking in the third stage. To enhance the accuracy of the ranking, collaborative filtering is considered when the new query is entered. By carrying out simulation experiments, it is shown that algorithm PPR is not only very efficient but also very adaptive in offering personalized ranking to users.

B. Arzanian et al., [61] proposed a multi-agent based personalized meta-search engine using automatic fuzzy concept networks. Since the dynamic content of the web develops rapidly, the common purpose web search engines are becoming poor. Even though the meta-search engines can assist with raising the search coverage of the web, the vast number of unrelated results returned by a meta-search engine is still causing problems for the users. The personalization of meta-search engines avoids this problem by filtering results according to individual user's interests. In this paper, a multi-agent structural design is developed for personalizing meta-search engine by means of the fuzzy concept networks. The most important objective of this paper is to use automatic fuzzy concept networks to personalize outputs of a meta-search engine presented with a multi-agent architecture for searching and fast retrieving. Experimental outputs indicate that the personalized meta-search results of the system are more appropriate than the combined results of the search engines.

Dik Lun Lee et al., [62] put forth personalized concept-based clustering of search engine queries. The remarkable development of information on the Web has forced new challenges for the construction of effective search engines. The most important problem of Web search is that search queries are typically short and ambiguous, and thus are inadequate for identifying the precise user needs. To alleviate this difficulty, a few search engines recommend terms that are semantically connected to the specified query so that users can select from the suggestions the ones that return their information needs. In this paper, the author introduced an efficient technique that recognizes the user's conceptual preferences with the intention of providing personalized query suggestions. This objective can be realized with two new strategies. At first, develop online approaches that extract concepts from the Web-snippets of the search outputs returned from a query and

utilize the concepts to recognize related queries for that query. Then, propose a novel two - phase personalized agglomerative clustering approach that is capable of creating personalized query clusters. No earlier work has focused personalization for query suggestions, according to author's knowledge. To estimate the efficiency of this technique, a Google middleware was formulated for collecting Clickthrough data to perform an experimental evaluation. Experimental results show that this technique has enhanced precision than the existing query clustering approaches.

F. Akhlaghian et al., [63] proposed a personalized search engine using ontology- based fuzzy concept networks. Since the users may have various backgrounds and anticipations for a specified query, personalization of search engines outputs based on user's profile can assist to better match the overall interests of an individual user. In this paper the authors personalize the search engine outputs with the help of automatic fuzzy concept networks. The main objective is to make use of the concepts of ontology to improve the common fuzzy conceptual networks built according to the user's profile. Experimental output shows enhancement in personalized search engine outputs using enriched fuzzy concept networks contrast to common fuzzy concept networks.

CHAPTER 4

PROPOSED WORK

4.1 Objective

The fundamental objective of this work is to address the issues related to shortcomings of query crafting. We have proposed a simple but yet reasonable model which not only ensures to help user in query crafting but also promises for effective and relevant information retrieval. Following are the objectives of our work:-

- To do user profiling based on user browsing history. User profile helps us to map user query to a set of user related categories.
- To find thematically rich, good query terms using GA, so that these terms can be used for query expansion.
- To help user in crafting queries by giving personalized suggestions based on his/her profile.

4.2 Proposed Model

To improve search quality, we propose a novel framework of personalized web search system which takes individual interests into consideration for helping the search engine to retrieve the user's relevant information. With the development of information retrieval technology and personalized web search popular search engines can provide a fast response to user query and cover a huge amount of information and resources for users. We propose a simple yet reasonable search system which we can train on training datasets and which can make not only good use of the advantages of popular search engines, but helps to provide proper search results for people with different interests .

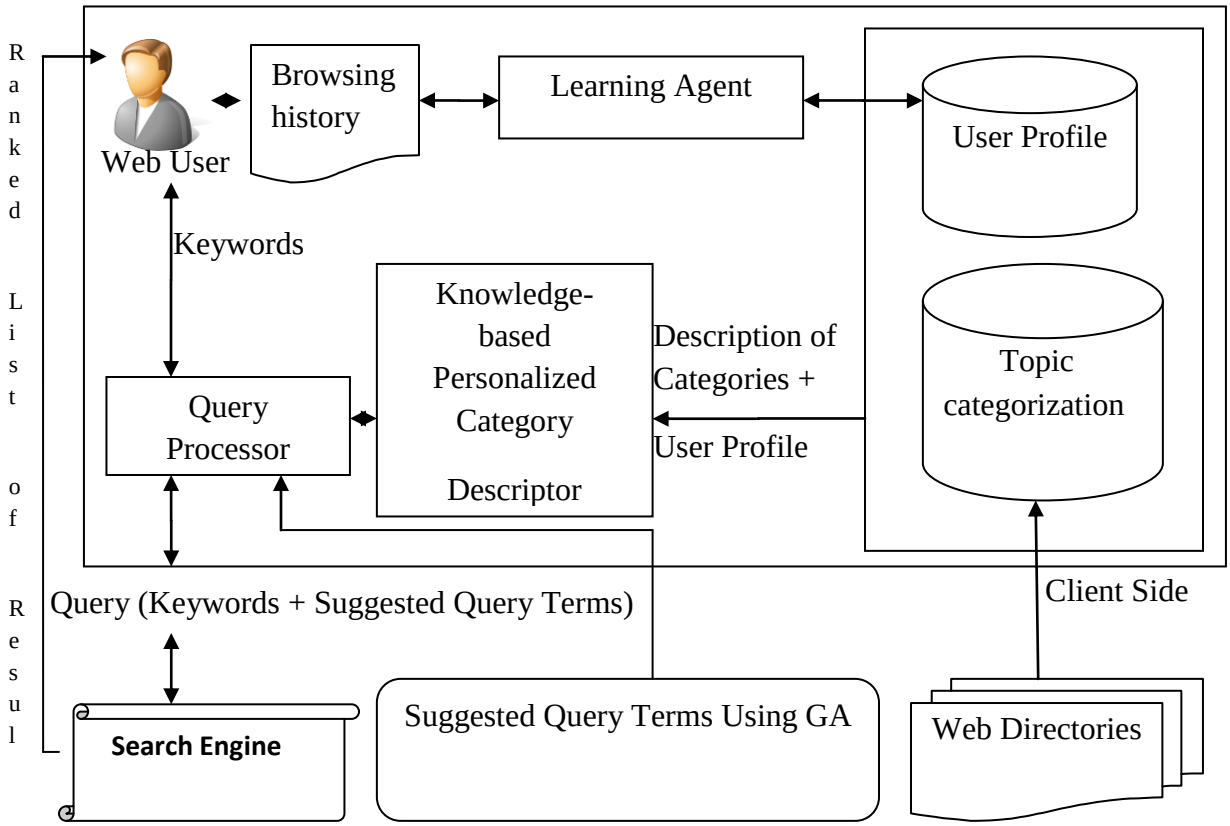

Figure 4.1 Proposed Generalized Framework of Personalized Web Search System

The system is set up at the client side. The learning agent can learn a user's preference automatically through analyzing user navigation/browsing history, and create/update user profile to adapt the user's most recent preference. After a user inputs query keywords, the query processor suggests more personalized thematically rich query expansion terms by computing the correlation between the query and user profiles.

4.2.1 User Profiling

User profile is used to reflect users' interest and infer their intentions for new queries. User profile also helps to deal with ambiguous queries. In our model a user profile is represented as a category preference vector, where weight of each category represents users' interest in that category. **Learning Agent** as shown in the figure above uses browsing history to build user profile. When the number of web pages browsed by the user grows above the specified threshold, the learning agent initializes/updates user

profile. User interest will thus be constituted by fix number of categories weight denoted by

$$U = \{cw_1, cw_2, cw_3 \ldots\ldots cw_m\} \qquad \text{……Equation (4.1)}$$

Where cw_i will be the number of web pages of category i visited by that user, normalized by maximum number of page visits among all categories.

To create the user profile we need to classify the web pages to a particular category. Alchemy API has been used for classifying web pages. As we are running our experiments on DMOZ, we have mapped these Alchemy categories to DMOZ categories as shown in the table 4.1 below.

Alchemy Categories	**DMOZ Categories**
Arts & Entertainment	Arts
Business	Business
Computers & Internet	Computers
Culture & Politics	Regional
Gaming	Games
Health	Health
Law & Crime	Society
Religion	
Recreation	Recreation
Science & Technology	Science
Sports	Sports
Weather	News

Table 4.1 Alchemy API to DMOZ Category Mapping

Alchemy API classifies a web page by giving it a particular category along with confidence (numerical value) which shows its probability of belonging to that particular category. Only if the web page is classified with confidence above the specified threshold level then only we have consider that page to contribute in user profile.

4.2.2 Mapping Queries to Most Relevant Interest Category

To start with, the user inputs his/her query to the search engine without specifying its category. The first step towards suggesting expansion terms is to map a user query to the most relevant interest category which represents users' search intentions and serves as a context to disambiguate user query. **Knowledge Based Personalized Category Descriptor** shown in the figure 4.1 above gives all possible categories for a given user query. **Query Processor** uses both the knowledge database and user profile to map the user query to related set of categories.

Let Q_{init} the query submitted by the user. First we find the weight of user submitted query in each category as identified by DMOZ. Let's say $Q(C_i)$ represents the weight of query in category C_i .

Now method of choosing most relevant category is as follows :

$$C_i = cw_i \times Q(C_i) \qquad \text{.....Equation(4.2)}$$

Thus we find C_i for each category. The category i, which has the highest weight will be assigned for that query for that user.

4.2.3 GA for Finding Good Query Terms

When a person use World Wide Web through search engine then it he/she is not familiar with query format notions, moreover there may be a term mismatch between the user query and the term used in relevant documents. This may decrease the retrieval efficiency of the system. An expert may be trained enough to form query while a general web user may not have expertise in query crafting. Furthermore, the general user may not have any special interest in such training. Research shows that even the domain expertise may not make significant impact on query formation. It's really hard to guess what words to use for the query that will adequately represent the person's problem and be the same as those used by the system in its representation.

Information retrieval by a search engine highly depends upon the match between vocabulary used to generate search queries and vocabulary used in document corpus.. In

this part of our proposed model we have used GA for finding good query expansion terms. In this scenario selecting good query expansion terms is treated as an optimization problem based on the effectiveness of a query to retrieve relevant terms. Some characteristics of this optimization problem are: (1) the high-dimensionality of the search space, where candidate solutions are queries and each term corresponds to a different dimension, (2) the existence of acceptable suboptimal solutions and (3) the possibility of finding multiple solutions. By suggesting these terms our aim is not only help the user to craft the query but also ensure effective information retrieval.

In order to accomplish our goal we start with initializing population of the queries formed by keywords and probable key phrases as suggestions extracted from web dataset of a particular category. Only those keywords and key phrases are considered who has the frequency above specified threshold.

Population and representation of chromosome

The search space S is constituted by all the possible queries that can be formulated to a search engine. Thus the population of chromosomes is a subset of such queries. Consequently, each chromosome is represented as a list consists of a keyword and along with key phrase suggestion, where both keyword and key phrase corresponds to a gene that can be manipulated by the genetic operators. The population is initialized with a fixed number of queries randomly generated. The number of terms in each of the initial queries will be random.

Fitness function

Lucene score [64] is associated with the search space as a fitness function which is a numeric value to evaluate quality/goodness of individual query. Different similarity measures, such as the standard cosine similarity or Jaccard similarity can be used in the implementation of the fitness function. Besides the standard cosine similarity, we have used Lucene score as it basically combines Boolean model and vector space model (VSM).

VSM is perhaps the best-known and most widely used information retrieval model. The model creates a space in which both documents and queries are represented by vectors. For a fixed collection of documents, a |v|- dimensional vector is generated for each document and each query from sets of terms with associated weights, where **|v|** is the number of unique terms in the document collection. Then, a vector similarity function, such as the inner product, can be used to compute the similarity between a document and a query. In VSM, documents and queries are represented as weighted vectors in a multi-dimensional space, where each distinct index term is a dimension, and weights are TF-IDF values.VSM does not require weights to be TF-IDF values, but TF-IDF values are believed to produce search results of high quality, and so Lucene is using TF-IDF in this scheme. Document is also represented as vector of weighted index terms.

$$\mathbf{W_{ij}=TF_{ij}*IDF_i} \qquad \text{……..Equation (4.3)}$$

Where the terms have following meaning.

W_{ij} is weight of i^{th} term in j^{th} document.

TF_{ij} is the frequency if i^{th} term in j^{th} document

$IDF_i=\log (N/df_i)$, IDF_i is inverse document frequency.

N is total number of documents.

df_i is number of document in which i^{th} term is present.

VSM score of document d for query q is the Cosine Similarity of the weighted query vector and document vector.

$$cosine(\mathbf{d}_j, \mathbf{q}) = \frac{\langle \mathbf{d}_j \bullet \mathbf{q} \rangle}{\| \mathbf{d}_j \| \times \| \mathbf{q} \|} = \frac{\sum_{i=1}^{|V|} w_{ij} \times w_{iq}}{\sqrt{\sum_{i=1}^{|V|} w_{ij}^2} \times \sqrt{\sum_{i=1}^{|V|} w_{iq}^2}} \quad \text{…Equation(4.4)}$$

Lucene refines VSM score for both search quality and usability:

- Normalizing document vector to the unit vector is known to be problematic in that it removes all document length information. For some documents removing this info is probably ok, e.g. a document made by duplicating a certain

paragraph *10* times, especially if that paragraph is made of distinct terms. But for a document which contains no duplicated paragraphs, this might be wrong. To avoid this problem, a different document length normalization factor is used, which normalizes to a vector equal to or larger than the unit vector: *doc-len-norm(d)*.

- At indexing, users can specify that certain documents are more important than others, by assigning a document boost. For this, the score of each document is also multiplied by its boost value *doc-boost (d)*.
- Lucene is field based, hence each query term applies to a single field, document length normalization is by the length of the certain field, and in addition to document boost there are also document fields' boosts.
- The same field can be added to a document during indexing several times, and so the boost of that field is the multiplication of the boosts of the separate additions (or parts) of that field within the document.
- At search time users can specify boosts to each query, sub-query, and each query term, hence the contribution of a query term to the score of a document is multiplied by the boost of that query term *query-boost (q)*.
- A document may match a multi term query without containing all the terms of that query (this is correct for some of the queries), and users can further reward documents matching more query terms through a coordination factor, which is usually larger when more terms are matched: *coord-factor(q,d)*.

$$Score(q,d) = coordfactor(q,d) \times queryboost(q) \times cosine(d,q) \times doclennorm(d) \times docbost(d)$$

…Equation (4.5)

Genetic Operators

The efficiency of genetic algorithm lies in its basic operators which are as follows:

a) ***Selection:*** Selection is basically to select the highly fit chromosome (query) among the population. The key idea behind this is to give preference to better individual allowing them to pass their genes to next generation. Probability of passing a query to the next generation is directly propotional to the fitness of the query and inversely propositional to the fitness of other queries. This method of selection is also known as roulette-wheel.

b) ***Crossover:*** Crossover is a prime factor that distinguish genetic algorithm over other optimization techniques. This is basically a method to merge the genetic information of two individual. Some of the queries pass to the next generation as it is, while some of them recombined with others to form new queries with a hope of better descendents.

c) ***Mutation:*** This operator help genetic algorithm not to converge in the local optima. This is basically a deformation of genetic information of individual to help in exploring search space.

Proposed system architecture

Figure 4.2 shown below shows the proposed system architecture for finding “good query term suggestion” for the query entered by the user. The basic mechanism enables the system to evolve a population of individuals consist of keywords and query term suggestion.

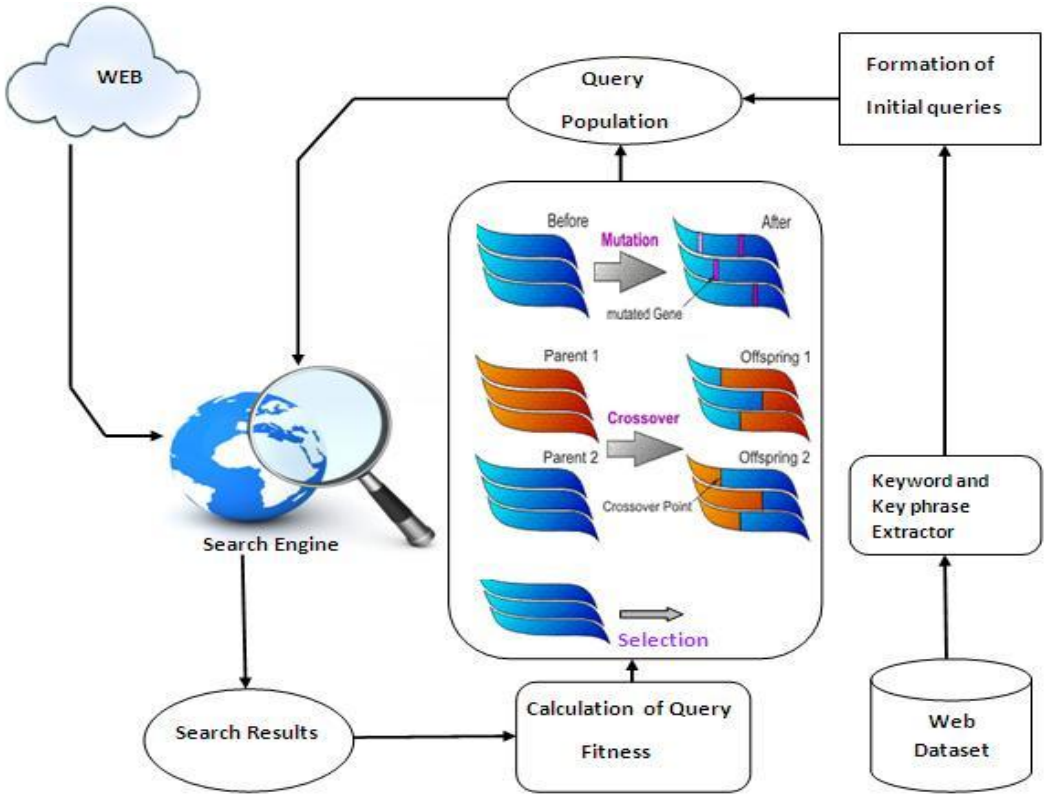

Figure 4.2 Proposed Framework for Finding Good Query Expansion Terms using GA

a) Formation of initial query

Initially the queries are formed using random selection of keywords and query term suggestions from respected term pools.

b) Calculation of query fitness

Fitness of query is estimated based on the relevance of result returned by the search engine after submitting the query. Fitness value is calculated using lucene score given to

each document for a given query. We have calculated the average score of the retrieved document to act as a fitness value.

c) Selection, Crossover and Mutation

These operations are performed in all iterations of GA until the termination condition achieved.

d) Mutation Pool

This term pool consists of the thematically rich terms extracted from web data set of specific category. When we perform changes in queries during mutation operation, terms are obtained from mutation pool. This procedure brings new terms to the query population, allowing a broader exploration of the search space. We have observed that in each upcoming generation, many high-quality queries are composed of terms that are not part of the previous population. These terms were extracted from mutation pool during mutation operation.

4.2.4 Suggesting Expansion Terms to Users

Once we are done with the user profiling and finding expansion terms using GA, then user profile can be used to give personalized query expansion from relevant category based on users' category interest vector. Category interest vector helped us to cop up with the ambiguous user queries by considering user's interest in these categories. For example in our experiments when the user entered the query 'genetic' then first we find out all possible categories of 'genetic' using Topic Categorization as shown in figure 4.1. Considering user's profile we mapped the user query to a set of related categories. Thus the user from life science department got query suggestions such as genetic disorder, genetic disease from science category while user from computer science department got suggestions like genetic programming, genetic algorithm etc. Some sample results have been attached in appendix.

4.3 Experiments and Results

In the absence of standard benchmark datasets which is suitable for our problem, we have designed our own dataset. In order to perform our experiment we have collected twelve different users' histories (by using chrome history view). Five of them were from computer science department while rests of them were from life science department. Further we have selected some topics from DMOZ. We have conducted our tests by choosing selected topics (say computers, science etc.). We have crawled the datasets from DMOZ for selected topics using Apache Nutch, while Apache Solr has been used for indexing crawled pages. By setting Crawling parameters of nutch we have restricted the crawling to specific DMOZ topic. Initial populations for each topic were created by extracting probable queries (keywords) and suggestions (key phrases) from respective DMOZ topic. Each keyword may have more than one occurrence by associating with different query suggestions in initial query population. Lucene score is assigned as a fitness of the individual (keyword + suggestion) in the population.

Once the data set was designed GA was applied. Each of our initial tests consisted in running the GA five times. Each run consisted in 20 generations, with a population of 650 queries. Fitness of the queries was normalized in all iterations by using maximum fitness among same keyword queries. The crossover and mutation probabilities were selected empirically. Experiments were conducted with different crossover probabilities and we got best results at P_c=0.6. Thus P_c was fixed to 0.6. Similarly experiments were repeated with varying probability as P_m=0 (no mutation), P_m=0.1(classic mutation) and P_m=0.3(hyper mutation). It has been observed that hyper mutation is giving best result amongst all three mutations probabilities.

Figure 4.3 below shows the average fitness of queries in each generation with different mutation rates where Quality-Avg (P) , as given in equation 4.6, was used to find average quality of a generation.

$$Quality\_avg(p) = \frac{\sum_{i=1}^{m} quality(q_i)}{m}$$

.....Equation (4.6)

The following figure 4.3 shows average quality of the population in all iterations with different mutation rates. In the case of hyper mutation, initially the average quality of the population increased significantly while toward reaching termination condition classic and hyper mutation performance was almost equal while no mutation was performing low.

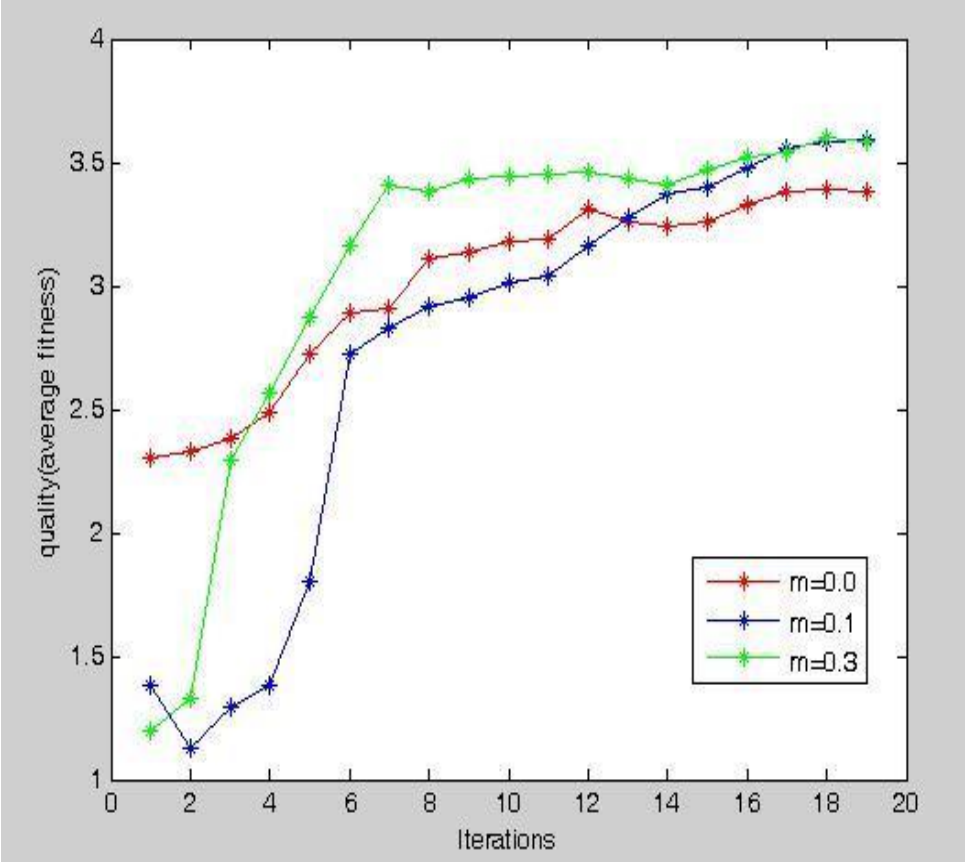

Figure 4.3 Average Quality of queries in each generation

The figure 4.4 shows the quality of best query in each generation where Quality -Max (P) has been used to find fittest query in a generation given in equation 4.7. The comparative analysis of different mutation rate has shown that in the case also hyper mutation was outperforming others as shown in figure 4.4.

$$Quality\_\max(p) = max_{i\in(1\ldots m)}(Quality(q_i))$$

……...Equation (4.7)

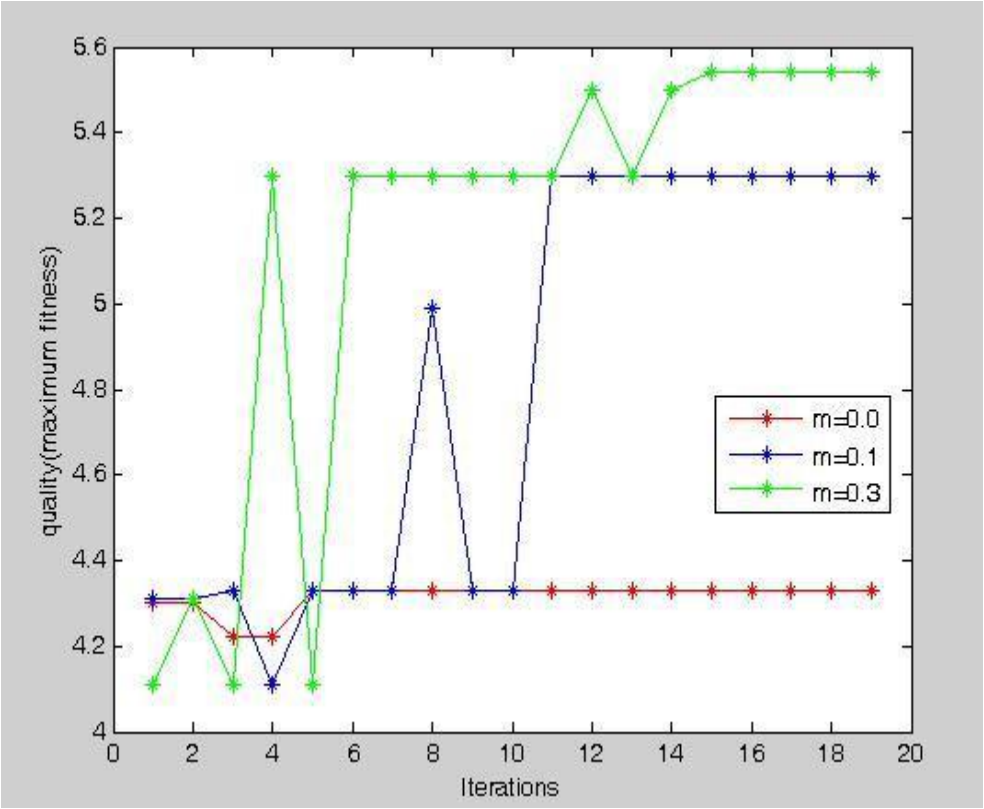

Figure 4.4 Quality of best query in each generation

We have also analyzed our results by taking queries itself as a dimension along with average fitness of the query starting with same keyword. The figure 4.5 below shows keyword wise Average quality of queries for all generations. These figures also show the impact of no mutation on the fitness of the query. The following figure shows that there is not much difference in quality of query grouped on the basis of starting keyword and due to no mutation the average quality of query group remained low as no new term was introduced on any iteration. Figure 4.6 has shown the same result with different quality parameter. In this figure we have considered quality of best query among all queries starting from same keyword in all iterations. Not much variation had been noticed in this case.

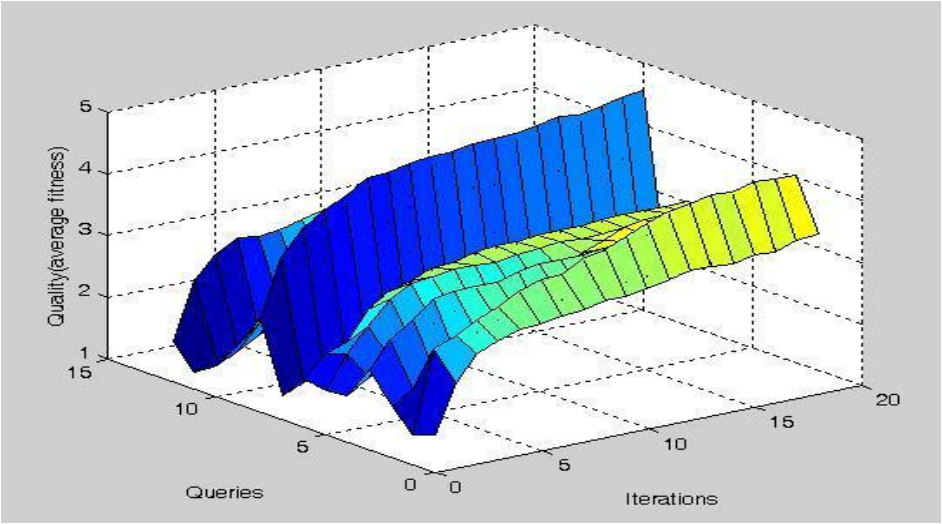

Figure 4.5 Average query quality in each generation for m=0

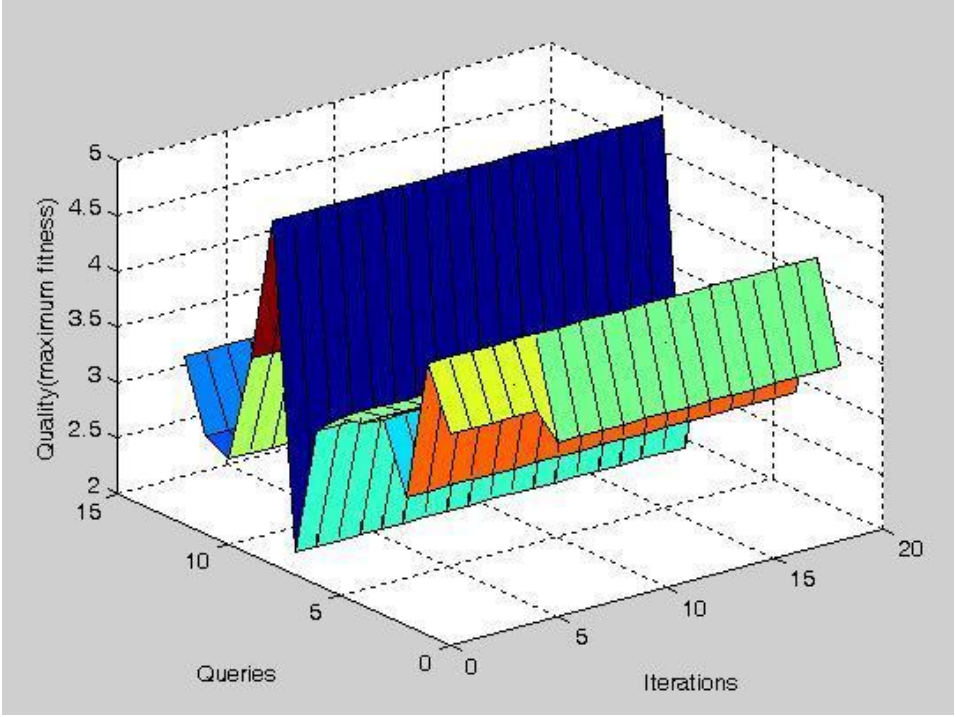

Figure 4.6 Quality of best query over all generation for m=0

The same thing has been shown in figure 4.7, but with classic mutation. In the case of classic mutation and no mutation, a comparative analysis shown that in the case of 'no mutation' the highly fit solution began to dominate. Even though we may get a good average quality but significant chances are there that resultant population would have

repeated individuals (queries). Figure 4.8 has shown the same result as shown in figure 4.6, but with classic mutation. Qualities of best query among queries starting from same keyword have been considered as quality parameter. A consistent growth has been noticed in max quality in all generations.

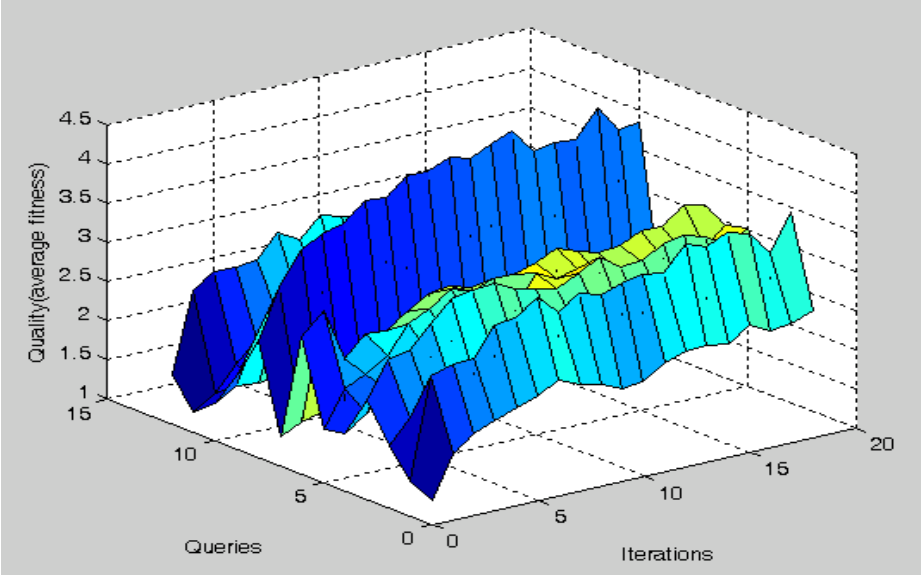

Figure 4.7 Average query quality in each generation for m=0.1

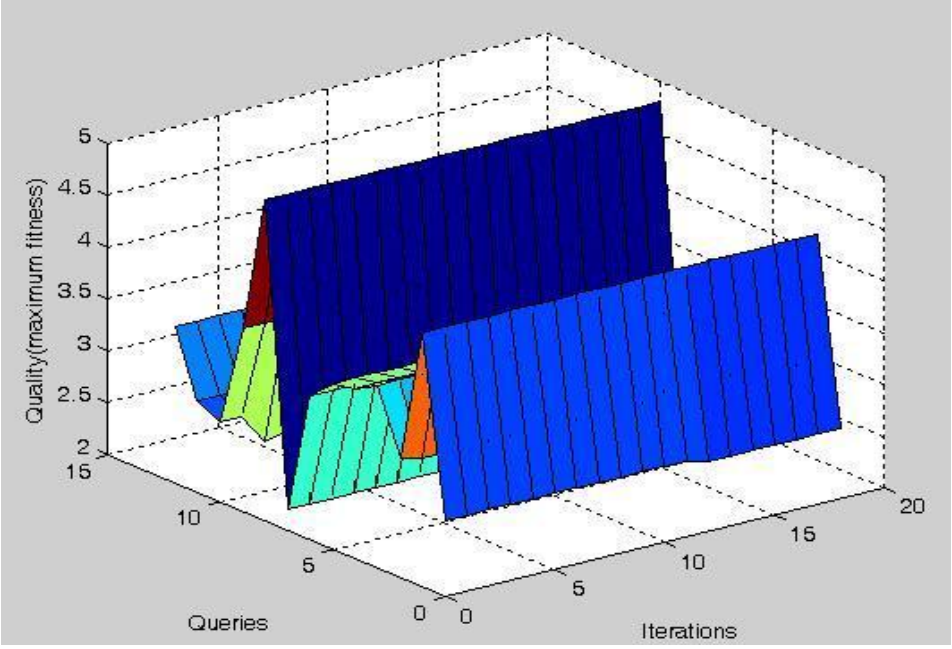

Figure 4.8 Quality of best query over all generation for m=0.1

The figure 4.9 also shows the results along with same dimensions but with hyper mutations. As we can clearly see that the rapid growth had been noticed in the query quality in all iteration. It has been observed that in case of hyper mutation we have better chances to get diverse queries. Figure 4.10 has shown the plotted graph on the same dimensions as discussed in figure 4.6 and figure 4.8. A variation has been noticed in max quality of the query but the performance was almost same as classic mutation.

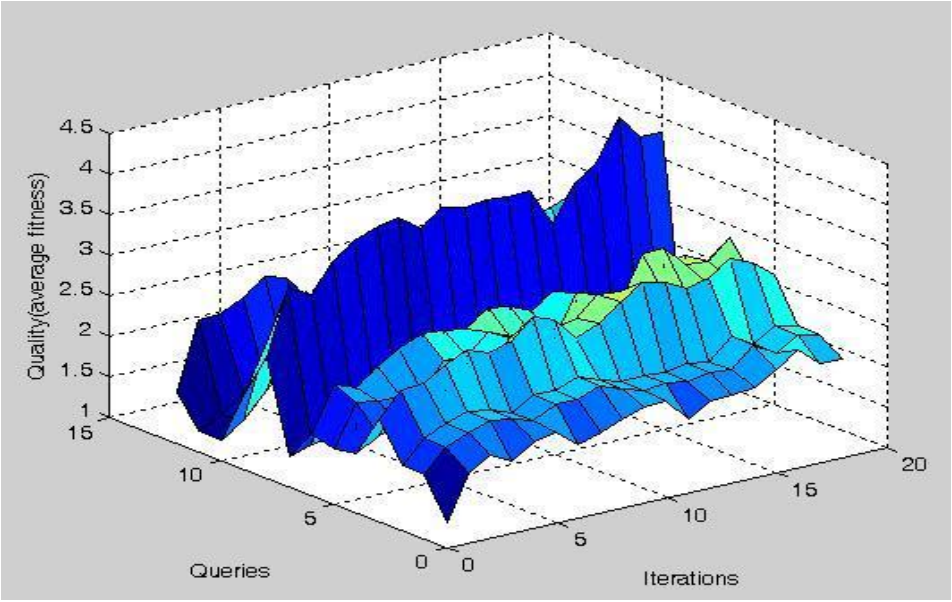

Figure 4.9 Average query quality in each generation for m=0.3

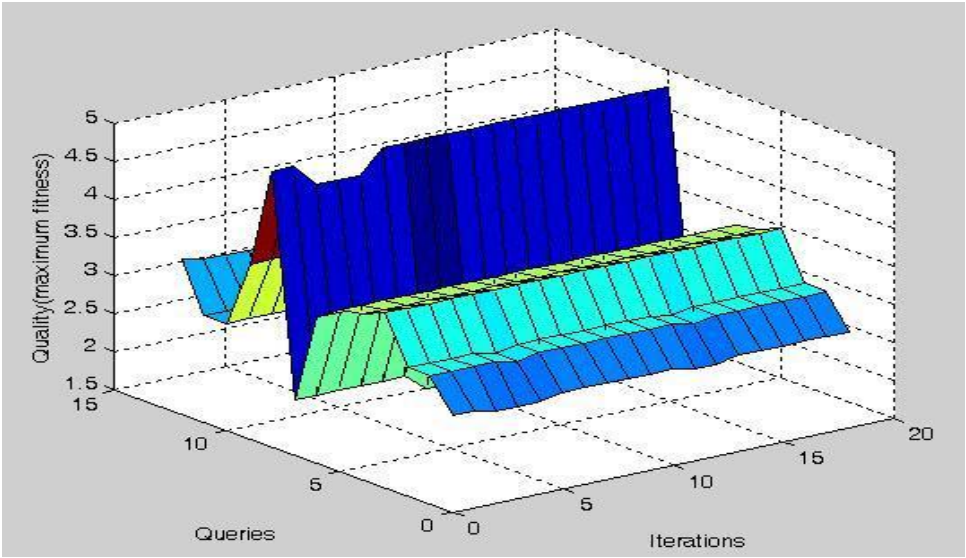

Figure 4.10 Quality of best query over all generation for m=0.3

4.4 Analysis of Results

In this section we have analyzed the result along with different dimensions as role of GA, significance of mutation and mutation pool etc.

4.4.1 General Role of GA

In the whole scenario selecting good query expansion terms is treated as an optimization problem based on the effectiveness of a query to retrieve relevant terms. As we have seen in our experiments and results we were getting better quality results in all iterations, the reason for such improvements is the suitability of problem characteristics with GA such as: (1) the high-dimensionality of the search space S which is constituted by all possible queries formulated to search engine. These queries are candidate solutions and each term corresponds to a different dimension, (2) the existence of acceptable suboptimal solutions and (3) the possibility of finding multiple solutions.

4.4.2 Significance of Mutation

Mutation is vital for evolution. It helps to explore the search space. Mutation can produce small random changes to the new population of queries. These changes consist in replacing a randomly selected query term T_q by another term T_m. The term Tm is obtained from a mutation pool. In our proposed model term T_m is obtained from mutation pool .As seen in the results in case of hyper mutation the query population rapidly approaches towards maximum fitness. Mutation has played a vital role in this scenario. It has been observed that many high quality queries were composed of terms that were not part of initial population. These terms came from mutation pool during mutation operation. Different mutation rates were selected empirically. In most of our runs hyper-mutation has given quick convergence of population towards optimal fitness.

4.4.3 Evaluating Thematic Richness

Experiments and results given in the last section show that the query population emerged after applying GA was able to fetch relevant documents in terms of Lucene score. To

evaluate the query in terms of thematic richness we have to establish suitable criteria suitable for this task. Each query in the population is substituted to the DMOZ directory and category result of DMOZ is used to assign the thematic richness to that query. Out of all the results only 100 are consider to derive thematic quality of the query.

The figure 4.11 below shows the average of thematic quality of all the queries starting with same keyword (probable user query). The equation 4.8 has been used for finding average thematic quality for each query starting with same keyword.

$$Thematic\_Quality(avg) = \frac{\sum_{i=1}^{q} \frac{total\ number\ of\ document\ of\ relevant\ category}{total\ number\ of\ document\ retrieved}}{|q|}$$

……Equation (4.8)

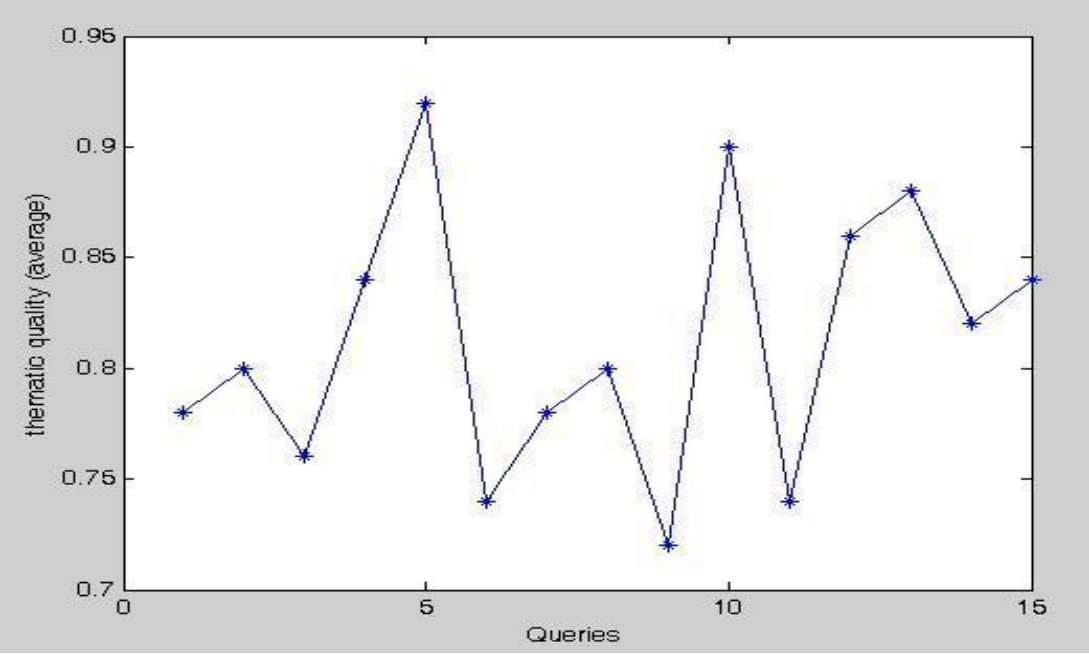

Figure 4.11 Averages Thematic Quality of Queries in Each Generation

The figure 4.11 shows that the resultant queries were thematically rich. Even the lowest average quality queries had more than 0.7 precision, while highly fit queries had up to 0.92.

The figure 4.12 below shows the quality of best thematic query starting with same keyword (probable user query) in each generation. The equation 4.9 has been used for finding average thematic quality for each query starting with same keyword

$$Thematic\_Quality(max) = max_{i\in(1,\ldots q)}(Thematic\_Quality(q_i))$$

……Equation (4.9)

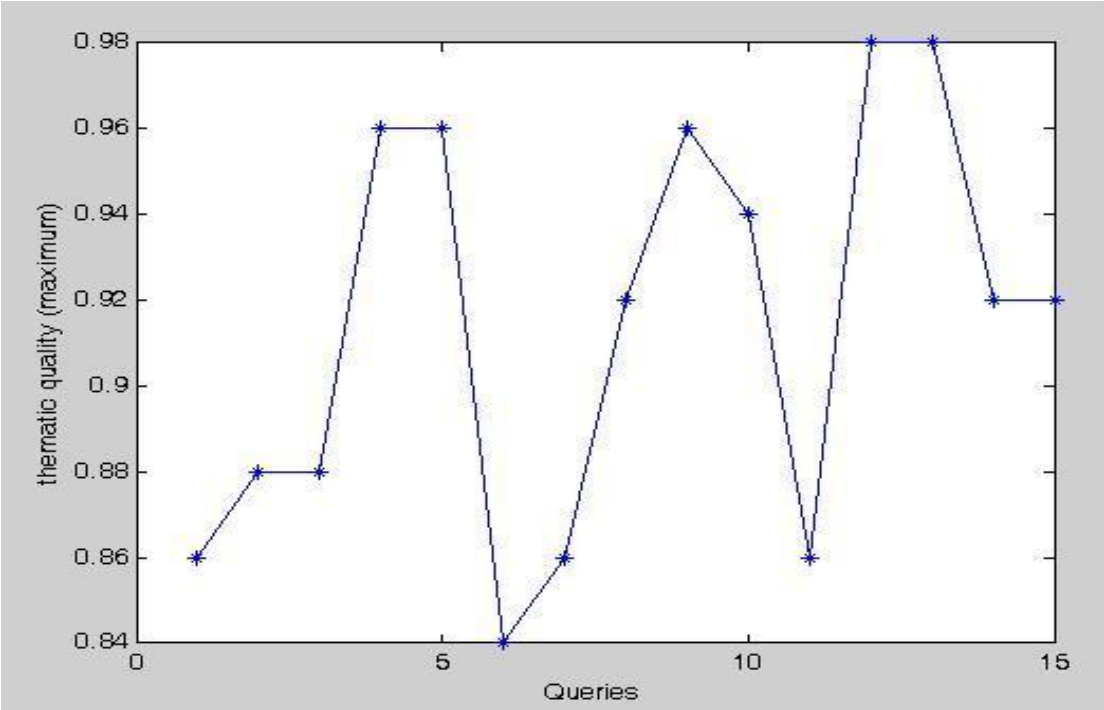

Figure 4.12 Quality of best thematic queries in each generation

The above figure 4.11 and 4.12 shows that there is not much variation in queries in terms of thematic richness. Maximum precision of the query is at least 0.84 and at most 0.98.

4.4.4 User Profile

Creating user profiles have shown significant improvement to disambiguate user queries and helped to provide personalized query expansion suggestions which results in high quality of relevant retrieval results.

CHAPTER 5

CONCLUSIONS AND FUTURE WORK

Query formulation is an essential part of successful information retrieval. There are challenges in formulating effective queries in web information search, because the web is used by a diverse population varying in their levels of expertise. Coping with ambiguous queries has long been an important part of research in information retrieval. Standard web searchers consider very little information about the user. It is increasingly difficult to let the search engine know what we want. Even if the user have domain expertise it is difficult to guess what words to use for the query that will adequately represent his/her problem and be the same as those used by the system in its representation. This work proposes a novel GA approach to personalized Web search based on thematic contexts. We have given a simple yet reasonable approach to learn user profile from his/her browsing history. The user profiles have been used to provide a method to deduce a set of related Categories for each user query. We have addressed the fundamental issues related to these shortcomings of the current Web search engines by establishing a new paradigm for interactive query formation by suggesting terms for query expansion. GA has been used to find good query terms for suggestions. Differently from most of the existing GA proposals to document retrieval, which attempt to tune the weights of the individual terms, our methods take each query as an individual. We have observed that many high-quality queries are composed of terms that are not part of the initial population.

The experiment and results show significant improvement in the search quality and demonstrate the potential of EA. But we have observed some limitations in this research, a large scale experiments are needed to prove the efficiency of proposed model. Another is if a model is proposed to assist the web users then it is meant to be evaluated by them as well. As a future scope of this research a comparative study can be done between other evolutionary and hybrid algorithms.

References

[1]A. Aula," *Query Formulation in Web Information Search*." In Isaias, P. & Karmakar, N. (Eds.) Proc. IADIS International Conference WWW/Internet,2003.

[2]F. Qiu and J. Cho. *"Automatic identification of user interest for personalized search."* In Proc. of WWW,22-26Edinburg,UK, 2006.

[3] "Google Official History",

http://www.comscore.com

[4] Sergey Brin and Lawrence Page, *" The anatomy of a large-scale hyper-textual web search engine."* In Proceedings of the Seventh International World Wide Web Conference, 1998.

[5] Sanjay Ghemawat, Howard Gobioff, and Shun-Tak Leung, *"The google file system."* InProceedings of the ACM Symposium on Operating System Principles, 2003.

[6] Bernard J. Jansen and Udo Pooch, *"A review of Web searching studies and a framework for future research. "* Journal of the American Society for Information Science and Technology, 52(3):235–246, 2001.

[7] Amanda Spink, Dietmar Wolfram, B. J. Jansen, and Tefko Saracevic, *"Searching the Web: The public and their queries."* Journal of the American Society for Information Science and Technology, 52(3):226–234, 2001.

[8] C. Silverstein, M. Henzinger, H. Marais, and M. Moricz, *" Analysis of a very large web search engine query log."* SIGIR Forum, 33(1):6–12, 1999.

[9] Keith Andrews, Christian Gutl, Josef Moser, and Vedran Sabol, *"Search result visualization with xFind. "* In Proceedings of the Second International Workshop on User Interfaces to Data Intensive Systems, 2001.

[10] "Alexa Internet, Inc. Alexa developer's corner",

http://www.alexa.com/site/devcorner, 2006.

[11] G. Salton, M.H. McGill, *"Introduction to Modern Information Retrieval,"* USA: McGraw-Hill, 1983.

[12] Bing Liu,Web *Data Mining :Exploring Hyperlinks, Contents and Usage Data* , Chicago, USA: Springer-Verlag Berlin Heidelberg 2007.

[13] *Handbook of Evolutionary Computation*, T.Bach, D.B. Fogel, Z. Michalewicz, New York, IOP Publishing and Oxford University Press, 1997.

[14] David, Grossman and Frieder, *Information Retrieval: Algorithm and Heuristic*, USA: Kulwar Academic Press, 1998.

[15] Gordan, M., *"Probabilistic and Genetic Algorithms for Document Retrieval,"* Communication of ACM 31(120) 1208-1218, 1998.

[16]David e. Goldberg, *"Genetic Algorithm in Search, Optimization & Machine Learning,"* USA: Pearson Education, 1989.

[17]Kalyanmoy Deb. "*Multi-Objective Optimization using Evolutionary Algorithms*," USA: John Wiley & Sons, Ltd, 2001.

[18] M. Gordon, "*User-based document clustering by re-describing subject description with a genetic algorithm,"* in Journal of the American Society for Information Science 42 (5), pp. 311–322, 1991.

[19] A.M. Robertson, P. Willet, *"Generation of equifrequent groups of words using a genetic algorithm."* Journal of Documentation 50 (3) ,pp.213–232, 1994.

[20] D. Vrajitoru, *"Crossover improvement for genetic algorithm in information retrieval,"* Information Processing and Management 34 (4) pp. 405–415, 1998.

[21] W. Fan, M.D. Gordon, P. Pathak, *"Personalization of search engine services for effective retrieval and knowledge management,"* in: Proc. 2000 International Conference on Information Systems (ICIS), Brisbane, Australia, 2000.

[22] P. Pathak, M. Gordon, W. Fan, *"Effective information retrieval using genetic algorithms based matching functions adaption,"* in: Proc. 33rd Hawaii International Conference on Science (HICS), Hawaii, USA, 2000.

[23] W. Fan, M. Gordon, P. Pathak, *"Automatic generation of a matching function by genetic programming for effective information retrieval,"* in: America's Conference on Information System, Milwaukee, USA, 1999.

[24] C.J. Van Rijsbergen, *Information Retrieval*, second ed., USA: Butterworth, 1979.

[25] H. Chen et al., *"A machine learning approach to inductive query by examples: an experiment using relevance feedback, ID3, genetic algorithms, and simulated annealing,"* Journal of the American Society for Information Science 49 (8) pp. 693–705,1998.

[26] J. Horng, C. Yeh, *"Applying genetic algorithms to query optimization in document retrieval,"* Information Processing and Management 36 pp. 737–759, 2000.

[27] Armstrong, R.; Freitag, D.; Joachims, T.; and Mitchell, *"Web Watcher: A learning apprentice for the World Wide Web,"* In AAAI Spring Symposium on Information Gathering, pp. 6-12, 1995.

[28] Bauer, T., and Leake, *" WordSieve: A method for real-time context extraction. In Modeling and Using Context"* in Proceedings of the Third International and Interdisciplinary Conference, Berlin: Springer-Verlag, 2001.

[29] Pitokow, James; Hinrich Schütze, Todd Cass, Rob Cooley, Don Turnbull, Andy Edmonds, Eytan Adar, Thomas Breuel (2002). *"Personalized search"*. Communications of the ACM (CACM) 45 (9): 50–55.

[30] Nicolaas Matthijs, Filip Radlinski, *"Personalized web Search using long term browsing history."* In WSDM'11 ACM 9-12 Hong Kong, China,2011.

[31] "Dmoz directory",

http://www.dmoz.org/

[32]F. Liu, C. Yu, and W. Meng, *" Personalized web search by mapping user queries to categories.'* In Proc. Of CIKM, 2002.

[33] F. Qiu and J. Cho, *"Automatic identification of user interest for personalized search."* In Proc. of WWW.2006.

[34]David B. Leake and Ryan Scherle, *" Towards Context-based Search Engine Selection."* In IUI'01 january 14-17,2001.Maxico.usa.

[35]J. Teevan, S. T. Dumais, and E. Horvitz, *"Personalizing search via automated analysis of interests and activities."* In Proc. of SIGIR, pages 449-456, 2005.

[36]T . Haveliwala, "*Topic-sensitive pagerank."* In Proceedings of the Eleventh Int 'l WorldWideWeb Conf ., 2002.

[37]P. Thomas and D. Hawking, *"Evaluation by comparing result sets in context."* In Proc. of CIKM, 2006.

[38]Z. Dou, R. Song, and J.-R. Wen, *" A large-scale evaluation and analysis of personalized search strategies."* In Proc. of WWW, 2007.

[39]T. Joachims, L. Granka, B. Pan, H. Hembrooke,F. Radlinski, and G. Gay, *"Evaluating the accuracy of implicit feedback from clicks and query reformulations in web search."* ACM Trans. on Info. Sys. (TOIS),25(2), April 2007.

[40] Shen X., Tan B., and Zhai C, *"Privacy protection in personalized search. SIGIR Forum,"* 41(1):4–17, 2007

[41] Chirita P.A., Firan C., and Ne jdl W, *" Summarizing local context to personalize g lobal web search."* In Proc. Int. Conf. on Information and Knowledge Management, 2006.

[42] Chirita P.A., Ne jdl W., Paiu R., and Kohlschu¨ tter C, *"Using ODP metadata to personalize search."* In Proc. 31st Annual Int. ACM SIGIR Conf. on Research and Development in Information Retrieval, 2005, pp. 178–185.

[43] Dou Z., Song R., and Wen J, *"A large-scale evaluation and analysis of personalized search strategies."* In Proc. 33rd Annual Int. ACM SIGIR Conf. on Research and Development in Information Retrieval, 2007

[44] Nauman, Mohammad Khan and Shahbaz, *"Using Personalized Web Search for Enhancing Common Sense and Folksonomy Based Intelligent Search Systems,"* International Conference on Web Intelligence (IEEE/WIC/ACM), Pp. 423 –426, 2007.

[45] "Folksonomy definition"

http://en.wikipedia.org/wiki/Folksonomy

[46] Zhengyu Zhu, Jingqiu Xu, Xiang Ren, Yunyan Tian and Lipei Li, *"Query Expansion Based on a Personalized Web Search Model,"* Third International Conference on Semantics, Knowledge and Grid, Pp. 128 – 133, 2007.

[47] P. Palleti, H. Karnick and P. Mitra, *"Personalized Web Search Using Probabilistic Query Expansion,"* International Conferences on Web Intelligence and Intelligent Agent Technology Workshops (IEEE/WIC/ACM), Pp. 83 – 86, 2007.

[48] Jie Yu and Fangfang Liu, *"Mining user context based on interactive computing for personalized Web search,"* 2nd International Conference on Computer Engineering and Technology (ICCET), Vol. 2, Pp. 209-214, 2010.

[49] Fang Liu, C. Yu and Weiyi Meng, *"Personalized Web Transactions on Knowledge and Data Engineering,"* Vol. 16, No. 1, Pp. 28 – 40, 2004.

[50] Xuwei Pan, Zhengcheng Wang and Xinjian Gu, *"Context-Based Adaptive Personalized Web Search for Improving Information Retrieval Effectiveness,"* International Conference on Wireless Communications, Networking and Mobile Computing, Pp. 5427 – 5430, 2007.

[51] Kyung-Joong Kim and Sung-Bae Cho, *"A personalized Web search engine using fuzzy concept network with link structure,"* Joint 9th IFSA World Congress and 20th NAFIPS International Conference, Vol. 1, Pp. 81 – 86, 2001.

[52] Chen Ding, J.C. Patra and Fu Cheng Peng, *"Personalized Web search with self-organizing map,"* The 2005 IEEE International Conference on e-Technology, e-Commerce and e-Service (IEEE'05), Pp. 144 – 147, 2005.

[53] C. Biancalana and A. Micarelli, *"Social Tagging in Query Expansion: A New Way for Personalized Web Search,"* International Conference on Computational Science and Engineering (CSE '09), Vol. 4, Pp. 1060 – 1065, 2009.

[54] K.W.-T. Leung, D.L. Lee and Wang-Chien Lee, *"Personalized Web search with location preferences,"* IEEE 26th International Conference on Data Engineering (ICDE), Pp. 701 – 712, 2010.

[55] J. Lai and B. Soh, *"Personalized Web search results with profile comparisons,"* Third International Conference on Information Technology and Applications (ICITA 2005), Vol. 1, Pp. 573 – 576, 2005.

[56] B. Smyth, *"A Community-Based Approach to Personalizing Web Search,"* IEEE Journals, Computer, Vol. 40, No. 8, Pp. 42 – 50, 2007.

[57] O. Shafiq, R. Alhajj and J. G. Rokne, "Community Aware Personalized Web search", International Conference on Advances in Social Networks Analysis and Mining (ASONAM), Pp. 3351 – 355, 2010

[58] Han-joon Kim, Sungjick Lee, Byungjeong Lee and Sooyong Kang, *"Building Concept Network-Based User Profile for Personalized Web Search,"* 9th International Conference on Computer and Information Science (ICIS), Pp. 567 – 572, 2010.

[59] Yan Chen, Hai Long Hou and Yan-Qing Zhang, *"A personalized context-dependent Web search agent using Semantic Trees,"* Annual Meeting of the North American Fuzzy Information Processing Society (NAFIPS), Pp. 1 – 4, 2008.

[60] Wen-Chih Peng and Yu-Chin Lin, *"Ranking Web Search Results from Personalized Perspective,"* The 8th IEEE International Conference on and Enterprise Computing, E-Commerce, and E-Services, The 3rd IEEE International Conference on E-Commerce Technology, Pp. 12, 2006.

[61] B. Arzanian, F. Akhlaghian and P. Moradi, *"A Multi-Agent Based Personalized Meta-Search Engine Using Automatic Fuzzy Concept Networks,"* Third International Conference on Knowledge Discovery and Data Mining, Pp. 208 – 211, 2010.

[62] Dik Lun Lee, W. Ng and K.W.-T. Leung, "*Personalized Concept-Based Clustering of Search Engine Queries,"* IEEE Transactions on Knowledge and Data Engineering, Vol. 20, No. 11, Pp. 1505 – 1518, 2008.

[63] F. Akhlaghian, B. Arzanian and P. Moradi, *"APersonalized Search Engine Using Ontology-BasedFuzzy Concept Networks,"* International Conference on Data Storage and Data Engineering (DSDE), Pp.137 –141, 2010.

[64] "Lucene Documentation",

http://lucene.apache.org/core/old_versioned_docs/versions/3_0_3/

Appendix

User query	**Suggested queries**
Bio-inspired	Bio inspired genetic algorithm
	Bio inspired learning algorithm
	Bio inspired artificial intelligence
	Bio inspired genes expression programming
Neural network	Artificial neural network
	Neural network artificial intelligence
Genetic	Genetic engineering
	Genetic programming
	Genetic evolutionary programming
Supervised learning	Supervised learning algorithm
	Supervised learning artificial neurons
Mining	Pattern mining
	Data mining
	Business patterns mining
	Mining learning algorithm
	Data mining and Data warehousing
	Data mining v/s Data warehousing
	Web mining

Database	Database management
	Database design
	Database programming language
	Database query language
	Data query optimization
IE	Information extraction
	Content extraction
	Linear regression

Table 2 List of Few Sample Results