A Major Project Report On

METAXPLORER: AN INTELLIGENT AND ADAPTABLE METASEARCH ENGINE USING A NOVEL OWA OPERATOR

Submitted in partial fulfilment of the requirements

for the award of the degree of

MASTER OF TECHNOLOGY

IN

SOFTWARE ENGINEERING

By

Neha Dimri

(Roll No. 2K13/SWE/08)

Under the guidance of

Dr. Daya Gupta

Professor

Department of Computer Engineering

Delhi Technological University, Delhi

Department of Computer Engineering

Delhi Technological University, Delhi

2013-2015

DELHI TECHNOLOGICAL UNIVERSITY

CERTIFICATE

This is to certify that the project report entitled **METAXPLORER: AN INTELLIGENT AND ADAPTABLE METASEARCH ENGINE USING A NOVEL OWA OPERATOR** is a bona fide record of work carried out by Neha Dimri (2K13/SWE/08) under my guidance and supervision, during the academic session 2013-2015 in partial fulfilment of the requirement for the degree of Master of Technology in Software Engineering from Delhi Technological University, Delhi.

To the best of my knowledge, the matter embodied in the thesis has not been submitted to any other University/Institute for the award of any Degree or Diploma.

Dr. Daya Gupta
Professor
Department of Computer Engineering
Delhi Technological University
Delhi

DELHI TECHNOLOGICAL UNIVERSITY

ACKNOWLEDGEMENTS

I feel immense pleasure to express my heartfelt gratitude to **Dr. Daya Gupta** for her constant and consistent inspiring guidance and utmost co-operation at every stage which culminated in successful completion of my research work.

I also would like to thank the faculty of Computer Engineering Department, DTU for their kind advice and help from time to time.

I owe my profound gratitude to my family which has been a constant source of inspiration and support.

Neha Dimri

Roll No. 2K13/SWE/08

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

World Wide Web has become the main place for searching information on any topic. This makes searching a key activity and thus, search engines the most widely used tools on the Web. However, as the Web continues to expand, the portion of the Web covered by each search engine is decreasing constantly. Metasearch engines address this issue by combining the results of multiple search engines and thereby increasing the search effectiveness. This research work proposes a new model for metasearch, MetaXplorer, which is both intelligent and adaptable. This research work also proposes a novel Ordered Weighted Averaging operator named Intelligent OWA operator, which is capable of handling the dynamic nature of decision making environment. The proposed Intelligent OWA operator is used for result aggregation in MetaXplorer, along with Fuzzy Analytical Hierarchy Process. Furthermore, MetaXplorer analyses the documents returned by individual search engines instead of considering their ranks in search engine result lists alone in the aggregation process, and thus is intelligent. Subjective evaluation of MetaXplorer is provided by comparing it with previously proposed models. This research work also performs the performance evaluation of MetaXplorer in terms of precision. The precision values for MetaXplorer are compared with three existing metasearch engines on the Web. The results indicate that MetaXplorer performs better than the existing metasearch engines and has several features which were not present in the previous models.

Keywords: MetaXplorer, Intelligent OWA, Metasearch, FAHP, MCDM, Information Retrieval

Chapter 1: INTRODUCTION

This chapter provides introduction to metasearch, Ordered Weighted Averaging (OWA) operators and Fuzzy Analytical Hierarchy Process (FAHP). This chapter also presents the motivation, scope and problem statement of the project. This chapter ends with a concise description of how this thesis is organised.

1.1 Basic Concepts

This section describes the fundamental concepts of metasearch, Ordered Weighted Averaging (OWA) operator and Fuzzy Analytical Hierarchy Process (FAHP).

1.1.1 Metasearch Concepts

This section provides the basic notions related to metasearch engines.

1.1.1.1. What is Metasearch?

A Metasearch Engine (MSE) is a system that supports parallel access to multiple search engines and aggregates the results from them in order to provide a single consolidated result list to the user. Metasearch engines, in essence, have two key functions: query dispatching and result aggregation [1]. The user enters a query to the MSE, which in turn dispatches it to the underlying search engines. The underlying search engines retrieve their respective results and return in to the MSE. The MSE then aggregates these results using some algorithm and return the aggregated result to the user (Figure 1).

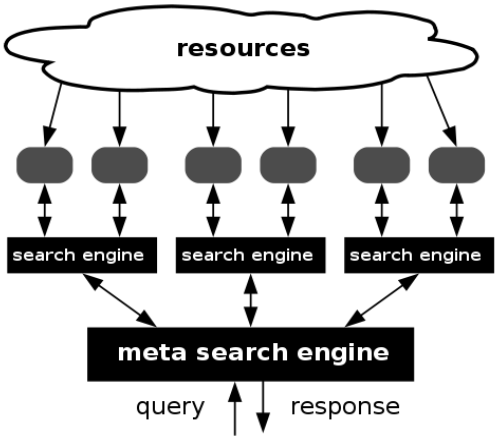

Fig. 1: Metasearch Engine Functioning [2]

Thus, MSEs are tools built on top of third party search engines and provide the user with a unified access to multiple search engines at once.

1.1.1.2. Why are Metasearch Engines Used?

With the growth of internet, World Wide Web (WWW) has become one of the main places to find information on any topic. WWW comprises of billions of web documents distributed over multiple web servers throughout the world. It is an environment which is growing rapidly and changing continuously. In order to locate specific information on the vast expanse of WWW, search engines such as Google, Bing, Yahoo, etc are the key tools used. However, the coverage of the Web by each of the search engines is constantly decreasing as the WWW continues to grow and expand. Several researches have demonstrated that no single web search engine has exhaustive coverage and it is implausible that any single search engine ever will [3]. A metasearch engine is a solution to address this limitation.

It is evident from many researches that metasearch engines, which combine the results of multiple search engines, can increase the search effectiveness significantly [4,5,6,7]. Also, the results by different search engines depict that only 45% of the relevant results are likely to be returned by single search engine and thus results quality can be significantly improved by combining the results of different search engines [8].

Thus, a metasearch engine extends the search coverage of the topic by forwarding queries to several search engines and combining their result lists. A metasearch engine is an improvement over a single search engine since it allows more relevant results to be extracted with the same amount of effort. The advantages of using a metasearch engine can be summed up as following [9]:

i. Expanding the search coverage of the Web.
ii. Addressing the scalability of searching the entire Web.
iii. Facilitating the invocation of numerous search engines in parallel.
iv. Increasing the information retrieval effectiveness.

As a result, metasearch is emerging as an interesting area of research and various researchers are proposing new and innovative techniques for the design of efficient metasearch engines.

1.1.1.3. Architecture of Metasearch Engine

Figure 2 describes the basic architecture of a metasearch engine. MSEs perform the following tasks sequentially:

i. Accepting a query from user.
ii. Processing of the submitted query.
iii. Passing query to underlying web search engines.
iv. Collecting and merging search results.
v. Presenting post-processed results to the user.

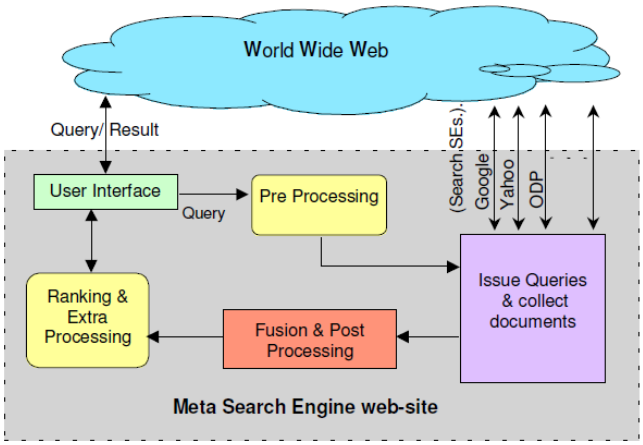

Fig. 2: Typical Metasearch Engine Architecture [10]

A typical session of a metasearch engine commences with a user submitting a query to the metasearch engine through an end-user interface. The metasearch engine next performs query pre-processing and further refines the submitted query. The processed query is then dispatched to numerous underlying search engines and result lists are obtained from each of them. These result lists are then merged or fused into a single ranked list by the metasearch engine using certain aggregation techniques. Post processing such as duplicate detection and removal or other advanced processing is performed over the results before presenting them to the user. Usually the number of results to be displayed is specified by the user and only the desired number of results is presented in the final ranked list.

1.1.1.4. Metasearch Engine Evaluation Criteria

The effectiveness of an information retrieval system is determined in terms of precision and recall [49]. Assume that the set of relevant documents with respect to a given query can be established. The two measures precision and recall can be defined as follows.

$$Precision = \frac{The\ number\ of\ retrieved\ relevant\ documents}{The\ number\ of\ retrieved\ documents} \quad (1)$$

$$Recall = \frac{The\ number\ of\ retrieved\ relevant\ documents}{The\ number\ of\ relevant\ documents} \quad (2)$$

Thus, precision is a measure of the fraction of retrieved instances that are relevant and defines the ability of the information retrieval system to filter out irrelevant hits and focus on potentially useful information. Whereas, recall is a measure of the fraction of relevant instances that are retrieved and defines how well a search finds every possible document that could be of interest.

For measuring the performance of an MSE, recall cannot be used as the total number of relevant documents in the underlying search engines' databases cannot be determined. *Thus, precision is used as a metric for evaluating the quality of results retrieved by the MSE.*

A set of test queries is considered for the evaluation of MSE's performance. For each test query, the documents in the result list are inspected in order to determine their relevance. Relevance of a document is usually considered to be binary, i.e. a document can be either 'relevant' or 'not relevant'. Precision is then calculated for the test query using Equation 1. The values of precision for each test query are averaged over the entire set of test queries to obtain the average precision value which is used as a performance indicator.

Another metric used by a few researchers is based on the MSE's response time. However this metric does not establish the quality of results returned by the MSE and thus is used less often for evaluation.

1.1.2. Ordered Weighted Averaging Operator Concepts

This section presents a basic overview of the Ordered Weighted Averaging (OWA) operators and their application in result aggregation phase of MSEs.

1.1.2.1. What are Ordered Weighted Averaging operators?

The problem of evaluating a cumulative decision function is crucial in many fields. In classical binary logic there are two extremes; "and", where all the criteria should be met, and "or", where at least one of the criteria should be met. In 1988, Yager[12] proposed OWA operator for aggregation in MCDM to form an overall decision function, which lies between the two extremes.

OWA operator of dimension n is defined as a function F: $I^n \rightarrow I$ (where I = [0,1]), with associated weighing vector W , $W = [W_1\ W_2\ \ldots\ W_n]$, such that

1. $W_i \in [0,1]$
2. $\Sigma\ W_i = 1$
3. $F(a_1,a_2, \ldots ,a_n) = w_1*b_1 + w_2*b_2 + \ldots + w_n*b_n$

where b_i is the i^{th} largest value in $a_1, a_2, \ldots ,a_n$

For example, assume F is an OWA operator with dimension, n = 4 and associated weighting vector, $W = [0.2\ 0.3\ 0.1\ 0.4]$. For the evaluation of F(0.2, 0.7, 0.4, 1.0), the ordered arguments are given by vector $B = [1.0\ 0.7\ 0.4\ 0.2]$.

F(0.2,0.7,0.4,1.0) = W'B
= (0.2)(1) + (0.3)(0.7) + (0.1)(0.4) + (0.4)(0.2)
= 0.53

Till date, OWA operator has been used to solve many real life Multi Criteria Decision Making (MCDM) problems. Some of these include the Doctoral Student Selection problem [13], using OWA operator in Minkowski distance [14], in data mining as data modeling and re-identification [15], information retrieval using metasearch [16], sports management [17], etc.

1.1.2.2. Evolution of OWA operators

The classical OWA operator [12,18], however, has evolved to overcome various limitations in it, as discussed by DK Tayal, Neha Dimri, Shuchi Gupta, 2012 [19]. In 2000, Ordered Weighted Geometric (OWG) operator was introduced for ratio-scale measurements

by Chiclana[20] since geometric mean is better suited compared to arithmetic mean for ratio-scale measurements [21,22].

In 2007, Chiclana[23] proposed Induced OWA (IOWA) operators, which extended the functionality of OWA operator by introducing an order inducing variable for re-ordering the arguments. Importance IOWA (I-IOWA) operator introduces the concept of criteria having different importances and performs arguments re-ordering on the basis of criteria importance degrees. A consistency value can be assigned to the experts' evaluation instead of explicit importance degrees being assigned to each expert. Consistency IOWA (C-IOWA) operator performs arguments re-ordering on the basis of the consistency index value of the experts. Preference IOWA (P-IOWA) operator performs arguments re-ordering by calculating relative preference values associated with each of the arguments. Thus, P-IOWA operator is somewhat like OWA operator only, where the arguments are ordered based on their values.

Since the inputs to OWA operator are numerical values, it is unable to handle linguistic data. In 2008, Zarghami[24] introduced EOWA operator, which extends OWA operator to incorporate the concept of linguistic inputs. In EOWA operator, the linguistic inputs are represented by their equivalent triangular fuzzy numbers, and then converted into crisp numbers by using the max-membership method. Although, EOWA operator handles linguistic preferences well, it cannot deal with the uncertain inputs whose values are known only under pessimistic and optimistic conditions. Therefore, in 2012, Suo[25] incorporated the concepts of interval theory to represent the uncertain arguments, and COG (Center Of Gravity) method was used for defuzzying, in conventional OWA operator. This was named AOWA (Advanced OWA) operator. The discussed OWA operators have been successfully applied in solving several MCDM problems.

1.1.2.3. Application of OWA operators to Metasearch

Diaz [26,27] applied the OWA operator for result aggregation in a metasearch model. In the OWA model, first each document in the result list is assigned a 'positional value' based on its rank in the list. The positional value of a document d_i in the result list l_k returned

by a search engine s_k is defined as $(n - r_{ik} + 1)$ where, r_{ik} is the rank of d_i in search engine s_k and n is the total number of documents in the result. Thus, a document with higher rank in a result list will have a greater positional value. Positional Values are a measure of the degree to which a document (analogous to a MCDM 'alternative') satisfies a search engine's (analogous to MCDM 'criteria') criteria for retrieval.

Most documents are considered relevant to a certain degree by some but not all search engines and thus, they appear in one or more ranked result lists but not in all. Thus each ranked result list has a unique composition in terms of the documents they contain. However, to apply the OWA operator, all documents need to be present in all lists, i.e. homogeneous composition. In order to achieve this Diaz [27] comes up with two heuristics to compute the virtual position of a document missing in a search engine result list. The next step is to apply one of the two heuristic. This creates a set of result lists where all documents are present and ranked.

Now in terms of MCDM, every document to a certain extent satisfies every search engines criteria for retrieval, i.e. homogeneous composition. The application of OWA operator to compute the value of decision function F, for each document is now straightforward. The final merged ranked list is obtained by sorting the documents in descending order on the basis of the value of function F.

Similar to the application of the classical OWA operator, I-IOWA operator has also been used in metasearch to overcome the classical OWA operator's inability to consider heterogeneous search environment. In case of heterogeneous search environment, the underlying search engines vary in terms of performance and this is reflected in their assigned importance degrees. Thus, a search engine with higher quality of retrieved results is assigned a higher importance degree.

1.1.3. Fuzzy Analytical Hierarchy Process

In 1980, Satty [28] formulated Analytical Hierarchy Process (AHP) for solving multi-criteria decision making problems. AHP is a multi-criteria decision making method, which is based on pair-wise comparison on a ratio scale. The AHP is based on the innate human

ability to make sound judgments about small problems. The benefits of pair-wise comparison in MCDM problems are demonstrated by Saaty in [29]. AHP, however, is unable to deal with the imprecision and uncertainty associated with decision makers' perception.

Therefore, Fuzzy AHP was developed in 1998 [30]. FAHP reflects the human thinking, by using linguistic quantifiers while making the comparisons instead of crisp numbers. Thus, crisp judgments are transformed into fuzzy judgments. The steps involved in FAHP computation are described next.

A matrix of size *nxn* is constructed, where *n* is the number of alternatives. Each entry in the matrix is a linguistic variable (Important, more important, etc.), thus incorporating fuzzy logic. The linguistic variables are represented by triangular fuzzy numbers.

$$\check{A} = \left(\check{a}_{ij}\right)_{nxn} = \begin{bmatrix} (1,1,1) & (l_{12}, m_{12}, u_{12}) & \dots & (l_{1n}, m_{1n}, u_{1n}) \\ (l_{21}, m_{21}, u_{21}) & (1,1,1) & \dots & (l_{2n}, m_{2n}, u_{2n}) \\ \dots & \dots & \dots & \dots \\ (l_{n1}, m_{n1}, u_{n1}) & (l_{n2}, m_{n2}, u_{n2}) & \dots & (1,1,1) \end{bmatrix} \quad (3)$$

These triangular fuzzy numbers are then converted to fuzzy interval using α-cut based method:

$$\alpha\text{left} = [\alpha^*(m\text{-}l)]+l, \quad \alpha\text{right} = u\text{-}[\alpha^*(u - m)] \quad (4)$$

where α ϵ [0,1] is confidence factor and (l,m,u) is triangular fuzzy number.

$$\check{p}_\alpha = \begin{bmatrix} (\alpha\text{left}_1 \alpha\text{right}_1) \\ (\alpha\text{left}_2 \alpha\text{right}_2) \\ \dots \\ (\alpha\text{left}_n \alpha\text{right}_n) \end{bmatrix} \quad (5)$$

Crisp Judgment Matrix, $C_{\boldsymbol{\lambda}}$, is next obtained to get a single, crisp value of one alternative over every other alternative, using the formula:

$$C_\lambda = \lambda * \alpha right + (1-\lambda) * \alpha left \quad (6)$$

where $\lambda \in [0,1]$ and is called the Optimism Index of Decision Maker

$$C_{\boldsymbol{\lambda}} = \begin{bmatrix} C_{\lambda 1} \\ C_{\lambda 2} \\ \dots \\ C_{\lambda n} \end{bmatrix} \quad (7)$$

FAHP has been applied in the following areas:

- GIS Application, 2008 [31]
- Project Risk Assessment, 2010 [32]
- Remote Sensed Data, 1998 [30]
- Evaluation Of Green Products Design, 2012 [33]
- Evaluate Success Factors Of E-commerce, 2005 [34]
- Capital Investment, 2005 [35]
- Evaluation and Selection of Construction Project Contractor, 2001 [36]

1.2. Motivation

Individual general purpose search engines providing search services have been unable to keep up with the fast growing and rapidly changing environment of the World Wide Web. The coverage of Web by each of the major search engines has been constantly reducing despite their efforts to comprehend larger portions of web space [3]. A metasearch engine is an improvement over search engines since it increases the search effectiveness by covering a wider span of the Web. Result aggregation techniques are used in a metasearch engine to combine the results from various underlying web search engines and presenting a single consolidated ranked list to the user. Most of the researches involving MSEs vary by the result aggregation mechanisms proposed and incorporated by them.

Recent researches on metasearch engines are based on Ordered Weighted Averaging (OWA) operators for result aggregation [1,26,37,38,39]. The usage of OWA operators for

result aggregation provides an efficient mechanism for merging the result lists from multiple search engines.

However, *the previous researches were unable to take into account the rapidly changing nature of the World Wide Web* and thus, could not handle the changes in underlying search engines' performance due to several factors such as modifications in indexing and ranking algorithms, updates to databases, etc. *Therefore, there is a need for the development of an MSE capable of adapting as the environment changes*.

Furthermore, *Ordered Weighted Averaging operators developed till now are highly dependent on the judgment of decision maker in assigning importance degrees to criteria*. This makes the decision making process prone to biased opinion of an individual. Thus, *a scheme is required which allows for unbiased assignment of importance degrees*. Some other facts that motivated this research are mentioned below:

- Most of the developed MSEs perform pre-processing of query and post-processing of results, but do not perform any internal processing over the returned results. They simply work on the returned results considering just their ranks in individual search engines. *This motivated us to incorporate an internal processing mechanism which analyses the different results returned rather than merely considering their ranks*.

- The most recent researches evaluate the proposed metasearch models using precision metric and evaluating it over 14 queries [1]. Here we have used a much wider test query set consisting of 30 queries selected from different areas.

From the above discussion it is evident that there is a need for the development of an intelligent metasearch engine which is capable of adapting with the environment and also an unbiased mechanism needs to be devised which assigns importance degrees to search engines. This motivated us to pursue research in the area of metasearch in order to address the issues present and design an intelligent MSE.

1.3. Problem Statement

Since a single web search engine cannot cover all the documents in the entire Web space, metasearch engines is emerging as an interesting area to the researchers. However, existing metasearch models have various limitations such as: inability to handle dynamic environment of the Web, absence of analysis of returned results, dependence over a biased mechanism for result aggregation. Although, the most recent metasearch model, MetaSurfer [1], is based on Fuzzy Analytical Hierarchy Process and a modified Extended Ordered Weighted Averaging Operator to deal with shortcomings of previous researches, it is unable to address any of the issues specified above. Therefore, there is a need for a result aggregation mechanism which can address these issues.

This research is aimed at developing an adaptable and intelligent metasearch engine, *MetaXplorer*. *MetaXplorer* is designed to consider the constantly evolving environment, hence '*adaptable*'. Also, it performs internal processing involving analysis of the returned results rather than merely considering their ranks, hence '*intelligent*'. Also, this research proposes the In-OWA (Intelligent Ordered Weighted Averaging) operator which provides *a mechanism for unbiased assignment of importance degrees to criteria*. Therefore, problem of the thesis can be stated as:

Development of an intelligent and adaptable metasearch engine, MetaXplorer, capable of handling the dynamic environment of the Web and incorporating analysis of returned results to compute their ranks in final result list.

1.4. Scope of Work

For the development of an adaptable metasearch engine, the changes in environment need to be reflected in the proposed metasearch model. The changes in environment basically refer to the variations in underlying search engines' result lists due to modifications in ranking and indexing algorithms, changes in databases, etc. These variations in result lists affect the performance of the underlying search engines and thus, may change their importance degrees with respect to each other. For example, suppose a metasearch engine

aggregates results from Yahoo and Bing. Initially Bing performed better compared to Yahoo and was thus assigned a higher importance degree. However, sometime soon, Yahoo may update its searching algorithm and give better quality results compared to Bing. The importance degrees at this stage would need to be updated accordingly.

Here we have developed an adaptable metasearch engine, MetaXplorer, using the proposed In-OWA (Intelligent-OWA) operator which learns the search engine importance degrees through examples. The training algorithm can be run periodically or according to user feedback to allow MetaXplorer to adapt as the environment changes. Another point to be noted is that the proposed In-OWA operator makes the decision making process unbiased as expert judgements do not assign importance degrees.

Also, MetaXplorer performs Uniform Resource Locator (URL) analysis of the results returned by multiple search engines, which assigns relevance scores to each document in the result lists. These relevance scores along with the documents' scores based on their ranks are used in the result aggregation process.

A set of 30 test queries from several areas is considered for evaluation of MetaXplorer. The average precision value over test query set is compared with those of three popular existing metasearch engines, namely Dogpile, Excite and Webcrawler. Therefore, scope of work can be summarized as:

- Design user interface of MetaXplorer which accepts the user queries and forwards them to underlying search engines.
- Incorporate adaptability with respect to changing environment of the Web through training with examples using the proposed In-OWA operator.
- Perform URL analysis over the set of documents retrieved in the result lists of underlying search engines to determine a measure of documents' relevance.
- Evaluate the quality of results retrieved by MetaXplorer over a set of 30 test queries by calculating precision.
- Compare the obtained precision value with three popular existing MSEs: Dogpile, Excite and Webcrawler.

1.5. Thesis Organisation

The remaining sections of the thesis are organised as follows:

Chapter 2 provides a detailed description of various metasearch engine models. It gives an insight to the advantages as well as disadvantages of the available techniques.

Chapter 3 presents the definition and a detailed explanation of the proposed novel Ordered Weighted Averaging operator, i.e. In-OWA operator.

Chapter 4 presents the proposed metasearch model, MetaXplorer, in detail.

Chapter 5 describes the implementation aspect of this research work.

Chapter 6 shows the evaluation of the proposed MSE, MetaXplorer. It also compares the performance of MetaXplorer with three popular MSEs: Dogpile, Excite and Webcrawler.

Chapter 7 concludes the thesis and depicts the possible improvements in this research work in future.

Chapter 8 exhibits the publications from this research.

Chapter 2: LITERATURE REVIEW

In this chapter we present a literature survey on the existing literature about metasearch models proposed by researchers in the past few years. A detailed description is provided for each metasearch model surveyed, including the improvements with respect to previous models, basic technique incorporated and evaluation of the model.

This chapter also presents a summarization of the discussed metasearch models in tabular form depicting several aspects such as: the underlying techniques used, year of establishment and comparisons based on certain evaluation criteria.

2.1 MetaCrawler

MetaCrawler [40,41] is one of the first metasearch engines developed and provided the users with a single interface to search simultaneously across several search engines. It was developed by Erik Selberg and Oren Etzioni at the University of Washington, Seattle and has been available since 1995. It uses a relatively straightforward mechanism for combining the results from multiple search engines and incorporates techniques for eliminating duplicate URLs. The steps involved in a typical invocation of MetaCrawler are described next.

2.1.1 Control Flow

Figure 3 depicts the basic flow of control of MetaCrawler and outlines the major steps involved in final results ranking computation. The steps followed are expressed below:

i. The process initiates with a user submitting a search query to MetaCrawler through a user-friendly interface.
ii. The user query is then refined and formatted appropriately for each underlying search service.

iii. The queries formulated specific to search services are then forwarded to the respective search engines such as Lycos, Excite, Yahoo, etc.

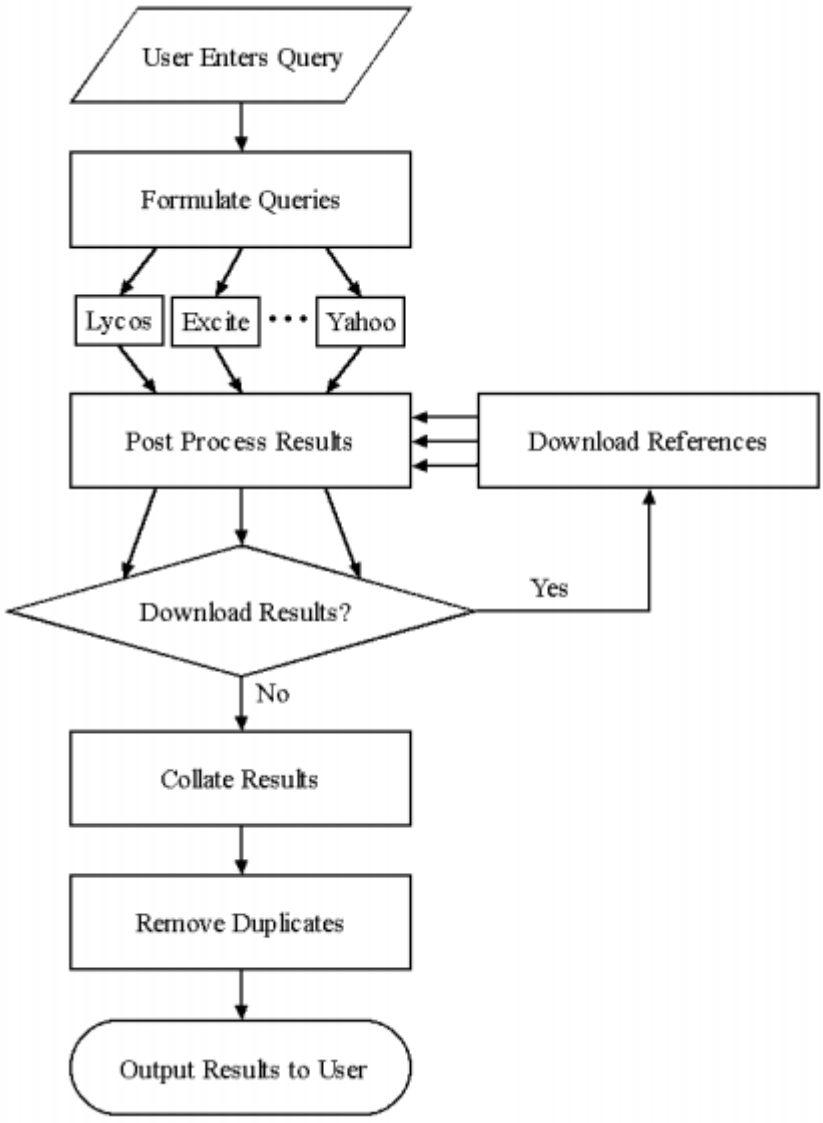

Fig. 3: MetaCrawler Control Flow [41]

iv. The results are parsed and aggregated with the results from all search services. MetaCrawler uses a '*confidence score*' to determine how closely a returned reference, i.e. document, matches the submitted query. A higher value of confidence score suggests a more relevant reference. For the calculation of confidence scores, first the values in the range 0 to 1000 are assigned to each reference returned by each search

service such that the topmost reference in each list is assigned a value of 1000. The values corresponding to each reference in different result lists (corresponding to different search services) are added to obtain its confidence score. Thus, the confidence scores are computed such that search services are allowed to vote for the best reference.

v. Finally, the results are analyzed to eliminate duplicate URLs in order to ensure quality and displayed as output to the user.

2.1.2 Model Evaluation

As discussed, the standard criteria for evaluating information retrieval systems are precision and recall. Manoj and Elizabeth Jacob [10] showed that MetaCrawler performed with a precision of 0.35 when evaluated by considering top 20 returned documents over 12 independent test queries.

Also, MetaCrawler has been published as the third most popular metasearch engine based on Alexa [42] Internet web-service, a subsidiary of Amazon.com [10]. Alexa ranks sites according to the number of visits by the users of Alexa Toolbar for various web browsers. Another measure for assessing popularity is by computing the number of pages that link to an MSE, i.e. in-links rank. Manoj and Elizabeth Jacob [10] considered the in-links ranks of MSEs provided by Google and Yahoo as in January 2008 and ranked MetaCrawler as second according to Google and fifth according to Yahoo.

2.2 Borda-Fuse Model

Borda-Fuse model was proposed by Aslam and Montague [43] in 2001 for result aggregation in metasearch. It is based on election strategies and allows for the underlying search engines to vote for the returned documents. Documents are ranked on the basis of the proposed Borda Count voting algorithm. The steps involved in the Borda Count algorithm are described next.

2.2.1 Model Description

Borda Count voting algorithm is used to assign '*Borda points*' to each document retrieved by each search engine. The calculation of Borda points is as follows:

i. Each search engine, i.e. voter, ranks a set of *n* documents in the order of preference.
ii. For each search engine, the top ranked document is assigned *n* points, the second ranked document is assigned *n* – 1 points, and so on.
iii. If some documents are missing in a search engine's result list, they are assigned the remaining points corresponding to that search engine evenly.
iv. For every document the points obtained from different search engines are added together to get the total Borda points for that document.
v. The documents are ranked in the order of Borda points and the document with highest Borda points wins the election.

2.2.2 Model Evaluation

Aslam and Montague [43] evaluated the performance of Borda-Fuse metasearch model using TREC 3, TREC 5, TREC 9 and Vogt datasets offered by Text REtrieval Conference (TREC). Each of the TREC datasets consists of 50 queries. The Vogt dataset comprises of a subset of TREC 5 dataset as defined by Vogt [44] and consists of 10 queries.

The performance is measured in terms of average precision over the test queries for each dataset. Figure 4 presents the graphical comparison of Borda-Fuse model with several other models in terms of average precision.

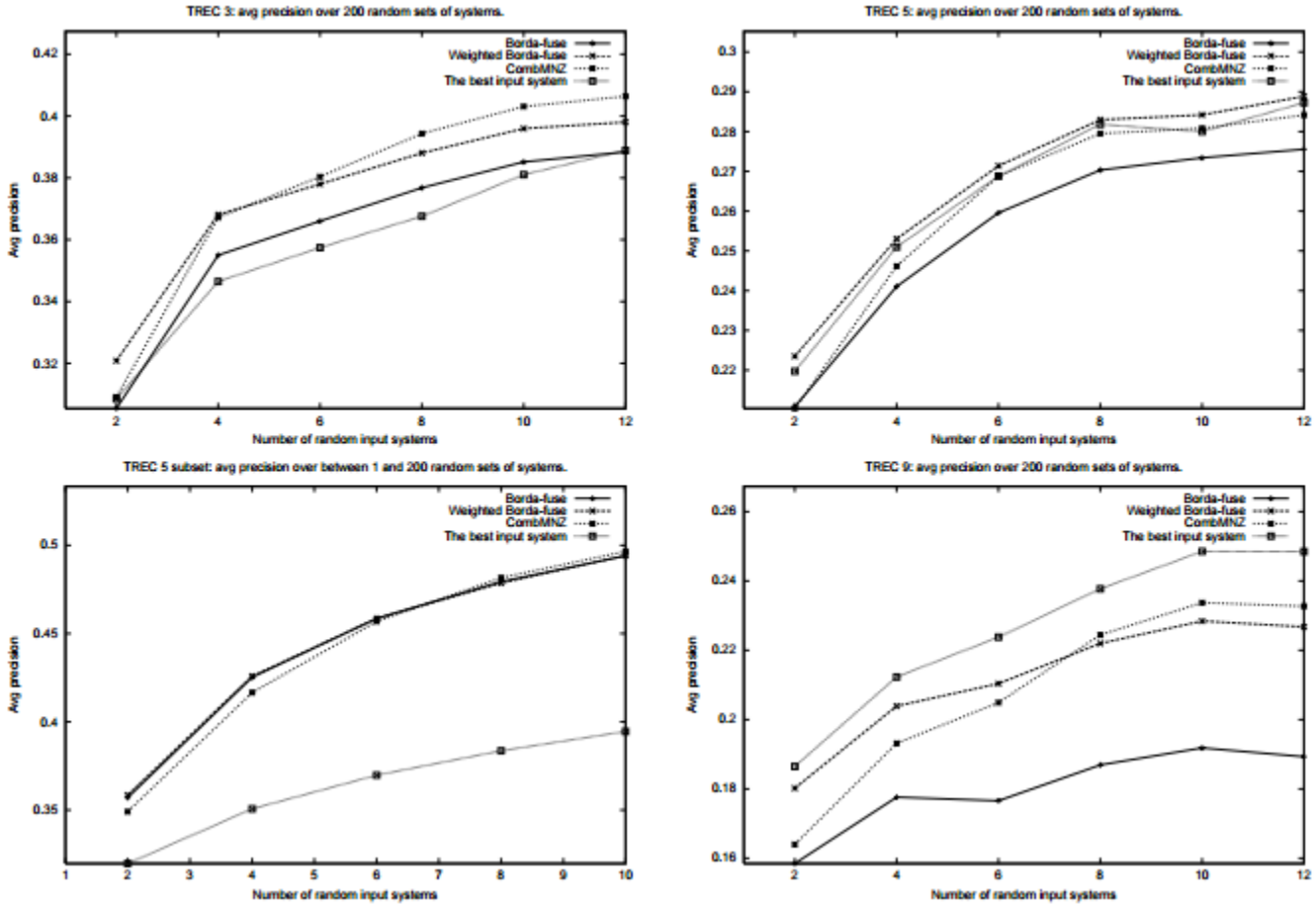

Fig. 4: Borda Fuse and Weighted Borda Fuse Evaluation [43]

2.3 Weighted Borda-Fuse Model

Although Borda-Fuse model is able to aggregate the results from different search engines, it considers all the search engines to be equally important in the election process, i.e. homogeneous situation. However, this ideal scenario does not exist in real world and different search engines should be assigned different importance values based on their performances. Thus, the aggregation scheme should consider this heterogeneous nature of the environment in the election process.

Weighted Borda-Fuse model [43] ranks the documents from different search engines considering search engine importance weights in the voting scheme. It was proposed by Aslam and Montague in 2001. The stepwise calculation of final document rankings is described next.

2.3.1 Model Description

Weighted Borda-Fuse model assigns '*importance weights*' to each search engine and provides a simple mechanism for obtaining the documents' ranking in the final list. The calculation of Borda-points in case of Weighted Borda-Fuse model is as follows:

i. Each search engine ranks a fixed set of *n* documents in the order of preference.
ii. For each search engine, the top ranked document is assigned *n* points, the second ranked document is assigned $n - 1$ points, and so on.
iii. If some documents are missing in a search engine's result list, they are assigned the remaining points corresponding to that search engine evenly.
iv. For each search engine, the Borda points assigned to the documents are multiplied with the search engine's importance weight.
v. For every document the products corresponding to different search engines, obtained in Step iv, are added together to get the total Borda points for that document. Thus, total Borda points correspond to the weighted sum of Borda-points assigned by different search engines.
vi. The documents are ranked in the order of Borda points and the document with highest Borda points is assigned topmost rank in the final merged list.

2.3.2 Model Evaluation

The performance of Weighted Borda-Fuse metasearch model was evaluated by Aslam and Montague [43] using TREC 3, TREC 5, TREC 9 and Vogt datasets offered by Text REtrieval Conference (TREC). The TREC datasets consist of 50 queries each and the Vogt dataset comprises of a subset of TREC 5 dataset [44] and consists of 10 queries.

The performance is measured by computing the average precision over the test queries for each dataset. Figure 4 presents the graphical comparison of Weighted Borda-Fuse model with several other models in terms of average precision. As can be observed, Weighted Borda-Fuse model performs better than Borda-Fuse model.

2.4 OWA Model

Borda-Fuse and Weighted Borda-Fuse models handle missing documents by distributing remaining points available evenly amongst them. This causes missing documents to be ranked at the bottom of the list. This results in less points being assigned to the missing documents. However, if a document is missing it does not mean it is less relevant. Missing documents appear since different search engines cover different portions of the Web as search space. This limitation concerning missing documents is addressed by the OWA model.

The OWA model is an application of Ordered Weighted Averaging operator in metasearch for multi-criteria decision making. It was proposed by Diaz et al. [26] in 2005. The description of OWA model is presented next.

2.4.1 Model Description

The OWA model proposes two heuristics for dealing with missing documents appropriately. Final document rankings are obtained by aggregating the '*positional value (PV)*' of each document using OWA operator. The method for result aggregation proposed by OWA model is as described below:

i. The positional value (PV) is computed for each document in different search engine result lists. Positional value of a document d_i in the result list l_k returned by a search engine s_k is defined as $(n - r_{ik} + 1)$ where, r_{ik} is the rank of d_i in search engine s_k and n is the total number of documents in the result.

ii. The missing documents are handled using one of the two heuristics: H1 and H2. In heuristic H1, the positional value for all the search engines where the document does not appear in the list (missing document) is denoted by the average of positional values in r search engines where it appears. This is denoted by Equation (8).

$$PV = \frac{\sum_{i=1}^{r} PV_i}{r} \tag{8}$$

In heuristic H2, the positional value for all the search engines where the document does not appear in the list (missing document) is denoted by the average of positional values in the total number of search engines, *m*. This is denoted by Equation (9).

$$PV = \frac{\sum_{i=1}^{r} PV_i}{m} \tag{9}$$

iii. The weights for OWA operator are generated using the linguistic quantifier based approach as shown in Equation 10. Let *Q* be the associated linguistic quantifier, *m* be the number of search engines (i.e. criteria) and $Q(r) = r^{\alpha}$ with $\alpha \geq 0$.

$$W_i = Q(\frac{i}{m}) - Q(\frac{i-1}{m}) \tag{10}$$

iv. The weights generated in Step iii and the positional values (PV) of document *d* are fed as input to the OWA operator to evaluate the function *F* as per Equation 11.

$$F(d) = \sum_{j=1}^{m} W_j \times PV_j \tag{11}$$

v. Final documents' ordering is obtained by sorting the documents in decreasing value of the function *F*.

2.4.2 Model Evaluation

The OWA model was evaluated by Diaz et al. [26] using TREC 3 dataset from Text REtrieval Conference (TREC). The dataset consists of 50 queries numbered from 151 to 200 and 40 search systems. The model is evaluated in terms of the standard measure of average precision.

Diaz performed the tests comparing the OWA model with different quantifier values of 0.5, 1, 2 and 2.5. Figure 5 shows the variation in average precision of the collated list for different values of α and compares the OWA model with the Borda-Fuse method. As can be viewed, OWA model produces higher precision values compared to Borda-Fuse method.

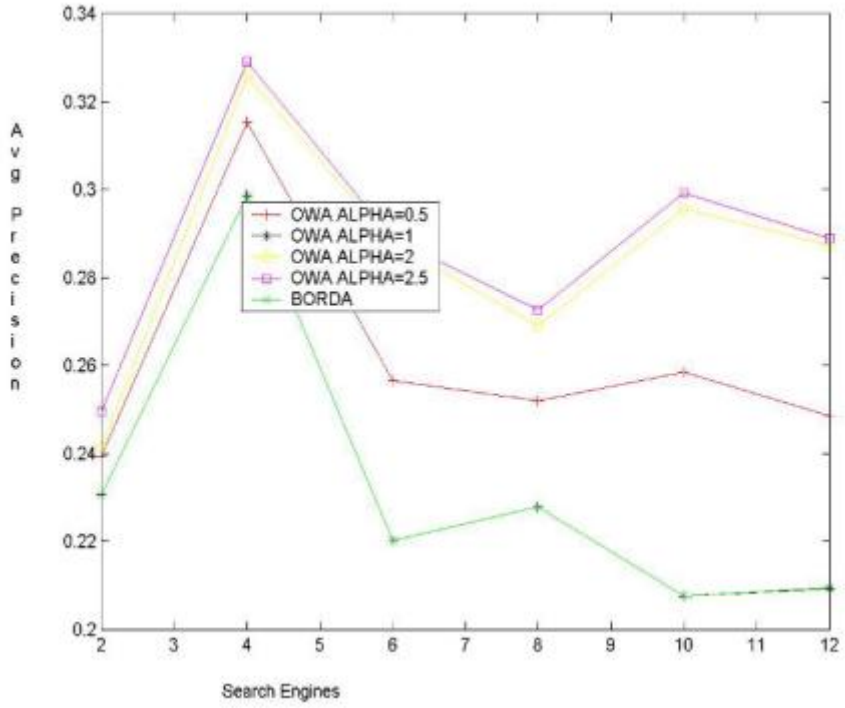

Fig 5: Evaluation of OWA Model [26]

2.5 Hybrid Fuzzy Model

Although the OWA model handles missing documents well, it does not consider search engine performances, suitability or user preference for search engines in the aggregation process. Also, relationships between documents and search engine correlation which might influence the search quality were not considered. To address these issues, De and Diaz [37] proposed the Hybrid Fuzzy model in 2009.

Hybrid Fuzzy model incorporates Analytical Hierarchy Process (AHP) in result aggregation, to compare documents as well as search engines pair-wise prior to aggregation using OWA operator. The description of Hybrid Fuzzy model is presented next.

2.5.1 Model Description

Hybrid Fuzzy model merges the result lists from different search engines in two stages. The first stage is associated with handling missing documents whereas the second stage comprises of application of AHP to perform result aggregation in metasearch. The two stages are described in detail below:

i. The missing documents are handled using the heuristic H1 proposed by Diaz in OWA model. After the missing documents have been assigned positional values, a set of homogeneous lists consisting of each document's positional value for each search engine are obtained.

ii. Search engines are ranked on the basis of their importance values and a search engine relationship matrix is created. Search engine scores are computed by applying AHP. Next, each search engine result list is analyzed to create a relationship matrix for documents. From this matrix, the document scores corresponding to each search engine are derived using AHP. The normalized document scores are aggregated using OWA operator in a similar fashion as in OWA model.

2.5.2 Model Evaluation

De and Diaz evaluated the Hybrid Fuzzy model using TREC datasets: TREC 3, TREC 5 and TREC 9 [37]. Each dataset consists of 50 queries and a set of search systems. The relevance information for the documents is also specified in the dataset.

De and Diaz compared the Hybrid Fuzzy model with OWA model with different values of α in the quantifier guided weights calculation. The value of α is varied as 0.25, 0.5, 1, 2, 2.5 and 5. Figure 6 shows the average precision values of Hybrid Fuzzy model compared with those of OWA model. It is clear from the average precision values that Hybrid Fuzzy model outperforms the OWA model.

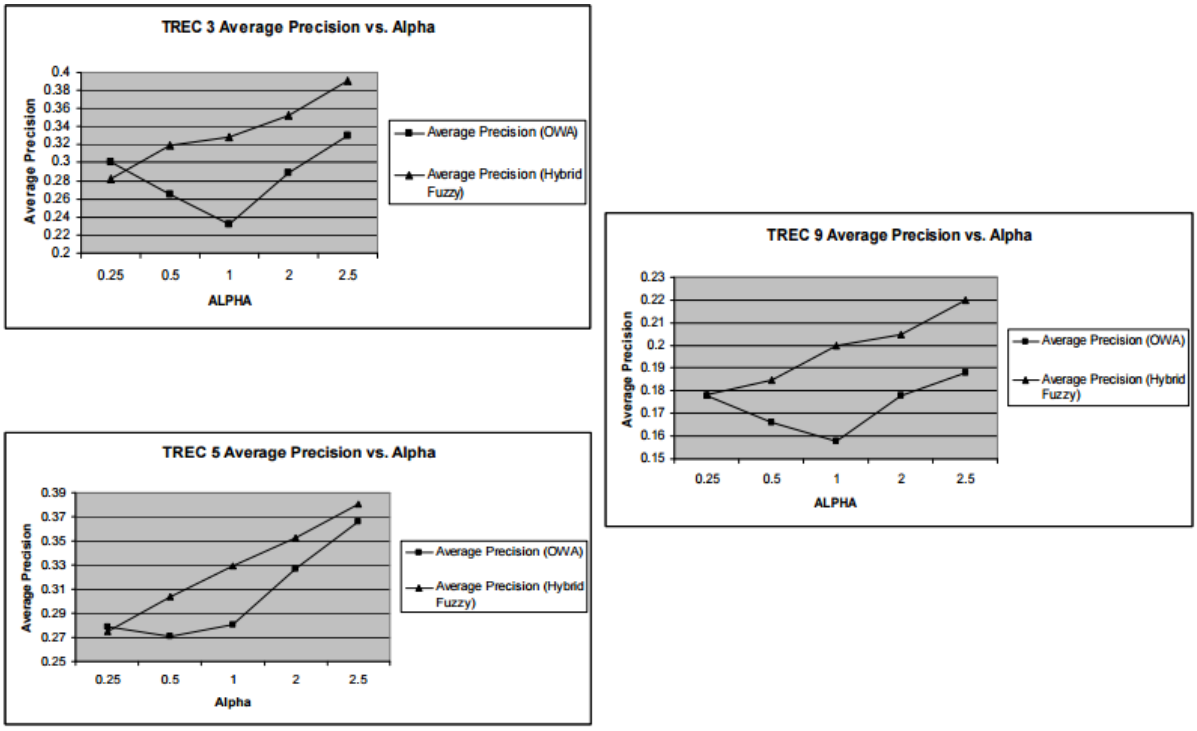

Fig 6: Hybrid Fuzzy Model Evaluation [37]

2.6 T-norm Hybrid Fuzzy Model

The Hybrid Fuzzy model uses OWA operator for aggregation of document scores in order to obtain a single merged result list. However, t-norm OWA aggregation operator has a superior performance compares to OWA operator in certain cases. Therefore, T- norm Hybrid Fuzzy model uses the T-norm (Triangular-norm) based OWA operator for result aggregation.

T-norm Hybrid Fuzzy model was proposed by De and Diaz [38] in 2010. It uses similar process as used in Hybrid Fuzzy Model for pair-wise comparisons of documents as well as search engines based on AHP. A brief description of the model is presented next.

2.6.1 Model Description

T-norm Hybrid Fuzzy model performs pair-wise comparison of documents and search engines on a scale of 1/9 to 9 as specified by AHP, similar to the method used in Hybrid

Fuzzy model. The document and search engine scores are also computed in the same fashion, the only difference being the aggregation operator used.

T-norm Hybrid Fuzzy model uses T-norm based OWA operator for aggregation of document scores in order to obtain the final ranking of documents. The decision function *F*, for document *d*, of T-norm OWA operator is computed as shown in Equation 12.

$$F(d) = \sum_{i=1}^{n} W_i \times T(B_j) \tag{11}$$

where *n* corresponds to the number of search engines, W_i correspond to the i[th] OWA weight obtained using quantifier guided approach, T is a T-norm function, $B_j = [b_1, \ldots\ldots\ b_j\ ]$ and b_j is the j[th] largest value of document scores. The T-norm function used by De and Diaz is product T-norm function, which multiplies all the elements in B_j.

2.6.2 Model Evaluation

De and Diaz [38] evaluated the T-norm Hybrid Fuzzy Model using the datasets TREC 3, TREC 5 and TREC 9 provided by TREC. Each dataset comprises of 50 queries and a set of search systems. Ad hoc tracks are considered for TREC 3 and TREC 5, whereas for TREC 9 web track is used.

The values of average precision obtained over the datasets are compared with Hybrid Fuzzy model for different values of α, used in quantifier guided weight computation, as depicted by Figure 7. As can be observed, T-norm Hybrid Fuzzy model has higher average precision and thus performs better than the Hybrid Fuzzy model and OWA model.

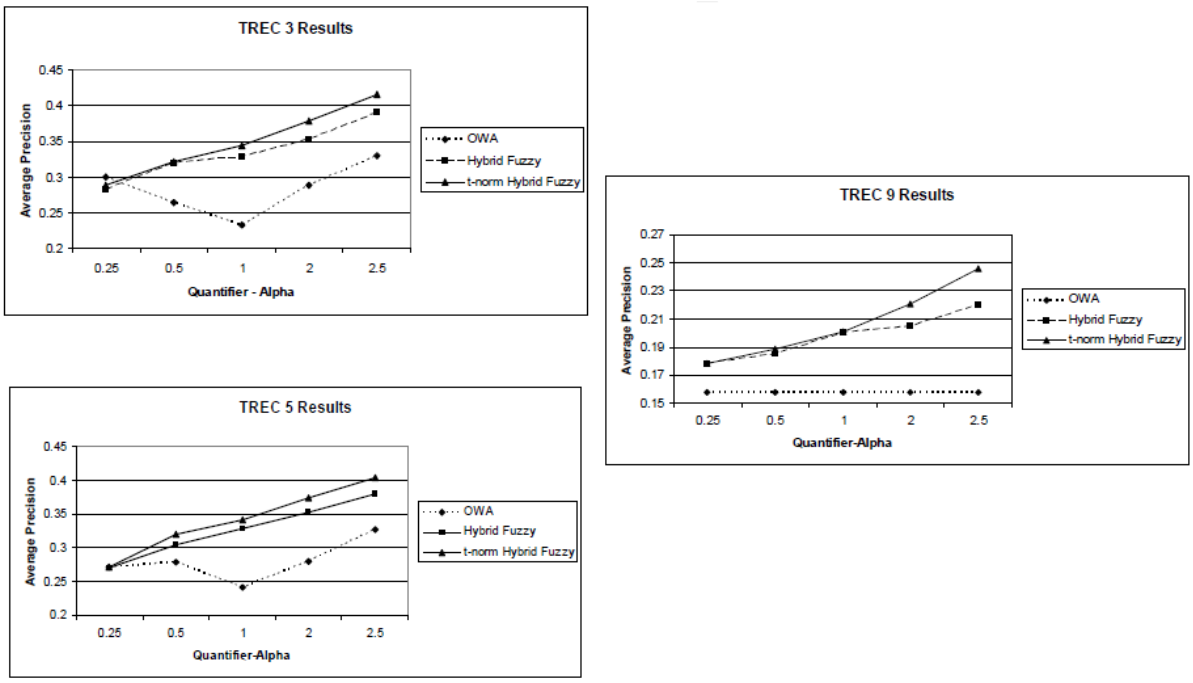

Fig 7: T-norm Hybrid Fuzzy Model Evaluation [38]

2.7 T-norm Importance Guided Hybrid Fuzzy Model

Even though T-norm Hybrid Fuzzy model allows for comparison of search engine pair-wise and considers the importance values of search engines relative to one another, it does not incorporate these importance values in the actual aggregation process. The actual aggregation process based on OWA operator is independent of the search engines' importance scores. To overcome this shortcoming De and Diaz proposed the T-norm Importance Guided Hybrid Fuzzy Model [39] in 2011.

The T-norm Importance Guided Hybrid Fuzzy Model extends the previous metasearch models proposed, by incorporating importance guided aggregation technique. This makes the aggregation process itself consider the search engines' importance scores while merging the individual result lists. The description of this model is provided next.

2.7.1 Model Description

The basic steps followed in the T-norm Importance Guided Hybrid Fuzzy Model are similar to those followed in the T-norm Hybrid Fuzzy Model. First the missing documents' scores are evaluated in similar fashion. The documents in each result list are compared pair-wise using a scale of 1/9 to 9 as per the AHP scale. The normalized document scores and search engines' scores also obtained using the same method as in T-norm Hybrid Fuzzy Model.

However, instead of multiplying each of the search engines' scores with the corresponding documents' scores, the search engines' scores are used to compute the importance weights for OWA operator using Equation 12.

$$W_i = Q\left(\frac{\sum_{k=1}^{j} u_k}{T}\right) - Q\left(\frac{\sum_{k=1}^{j-1} u_k}{T}\right) \quad (12)$$

where u_i denotes the importance of i[th] search engine, $T = \sum_{k=1}^{n} u_k$, and n is the number of search engines. The remainder of the aggregation process is similar to T-norm Hybrid Fuzzy model.

2.7.2 Model Evaluation

De and Diaz used Hersh [45] OHSUMED collection within the LETOR 2 (Learning TO Rank) dataset for evaluation of the model. The dataset consists of 106 queries for which a set of documents is pre-judged in terms of relevance. There are 25 features in the dataset analogous to search engines.

The T-norm Importance Guided Hybrid Fuzzy Model is compared with the Borda-Fuse, Weighted Borda-Fuse, OWA model, Hybrid Fuzzy model and the T-norm Hybrid Fuzzy Model in terms of average precision. Table 1 shows the average precision values for different values of α used in quantifier guided method for weight calculation. As can be observed, T-norm Importance Guided Hybrid Fuzzy Model performs better than the other models in most of the cases.

Table 1: T-norm Importance Guided Hybrid Fuzzy Model Evaluation [39]

Model Name	Quantifier Parameter α				
	0.25	0.5	1	2	2.5
OWA	0.4643	0.4417	0.3741	0.4479	0.4713
HFM	0.4012	0.4514	0.4522	0.4678	0.4792
tHFM	0.4218	0.4745	0.4754	0.4919	0.5038
tIGHFM-Score	0.4509	0.5073	0.5082	0.5258	0.5386
tIGHFM-Rank	0.4481	0.5041	0.5050	0.5225	0.5352

2.8 MetaSurfer

The T-norm Importance Guided Hybrid Fuzzy model uses AHP to assess the inter-document relationships and search engines' correlation, and thus is unable to handle the uncertainty associated with decision makers' perception. Fuzzy Analytical Hierarchy Process (FAHP) corresponds to human perception by using linguistic quantifiers instead of crisp numbers. MetaSurfer [1], a metasearch engine based on FAHP and modified EOWA operator, was proposed by Tayal et al. in 2014 to allow for imprecise and uncertain comparisons.

Also, modified EOWA operator allows the search engines' importance degrees to be represented in terms of linguistic quantifiers as well. Thus, MetaSurfer incorporates imprecision and uncertainty in the result aggregation in metasearch. The detailed explanation of the MetaSurfer MSE is presented next.

2.8.1 Model Description

MetaSurfer uses FAHP to compare the documents pair-wise, in each search engine's result list in order to compute the document scores. Search engines are assigned linguistic importance degrees through analysis of their performances by experts. The linguistic importance degrees are defuzzified using center of gravity (COG) method as opposed to max-

membership method used in EOWA [24] (hence modified EOWA) operator. The step by step procedure for obtaining final documents' ranking is as follows:

i. Missing documents are handled by a slightly different heuristic, in which missing documents' scores are computed by taking the weighted mean of that document's scores in the search engines where they appear. Thus, the concept of search engines' importance degrees is also incorporated in dealing with missing documents.
ii. The documents in each search engine's result list are compared pair-wise to create a pair-wise comparison matrix $D = [d_{ij}]$ where, each d_{ij} can take linguistic values represented by triangular fuzzy numbers.
iii. FAHP calculation, as described in [30], is followed for the computation of normalized document scores. Alpha-cut based method is applied on the triangular fuzzy numbers in *D*, to evaluate the interval performance matrix. Crisp Judgment matrix is computed to get a single numerical value for each document, which is then normalized to get the normalized document scores.
iv. Importance degrees are assigned to the search engines in terms of linguistic variables, represented by triangular fuzzy numbers. COG method is used to defuzzify the linguistic importance degrees to obtain crisp importance values.
v. '*Total preference*' of each document is calculated by multiplying the normalized document score with search engines' importance degree.
vi. The total preference values are fed as input to the OWA operator to calculated decision function *F*. Re-ordering weights for OWA are obtained by using quantifier guided approach.
vii. The final documents' ranking is obtained by re-ordering the documents in descending value of function *F*.

2.8.2 Model Evaluation

Tayal et al. [1] introduced a new concept for calculation of effectiveness of MSE, called weighted precision. Weighted precision gives a measure of topmost retrieved documents which are relevant. MetaSurfer's performance was assessed by calculating precison and weighted precision over a set of 14 test queries.

Table 2 shows the mean precision and weighted precision values for MetaSurfer and compares them with other MSEs available, namely Mamma, Webcrawler and Excite. MetaSurfer shows the highest precision (2.13) as well as weighted precision (19.97). Note that the precision values used here are not normalized.

Table 2: MetaSurfer Evaluation [1]

Metasearch Engine	Mean Weighted Precision	Mean Precision
MetaSurfer	19.97	2.13
Mamma	13.64	1.69
WebCrawler	17.63	1.88
Excite	19.00	1.97

2.9 Summarization of the Surveyed Models

Table 3 gives a brief comparison of some of the main metasearch result aggregation models.

Table 3: Metasearch Models Summarization

Metasearch model	Year of establishment	Underlying techniques used	Major Advantages	Main Shortcomings
MetaCrawler	1995	Confidence factor evaluation using a voting scheme	Simple method, Query formulation specific to search services, duplicate removal	Search engines considered to be equally important

Borda-Fuse Model	2001	Borda Count voting algorithm	Straightforward technique, Allows search engines to vote	Search engines considered to be equally important
Weighted Borda-Fuse	2001	Borda points along with search engine weights	Considers heterogeneous search environment	Missing documents are assigned lesser Borda points
OWA model	2005	OWA based aggregation	Proposed two heuristics for missing documents	Does not consider inter-document relationships or search engines' similarity
Hybrid Fuzzy Model	2009	AHP and OWA operator	Performs pair-wise comparison of documents as well as search engines	OWA sometimes performs worse than T-norm OWA operator
T-norm Hybrid Fuzzy model	2010	AHP and T-norm OWA operator	T-norm OWA operator performs better than OWA in certain cases	Search engines' importance degrees are not considered in actual aggregation
T-norm Importance Guided Hybrid Fuzzy model	2011	AHP and T-norm OWA operator with importance guided weights	Re-ordering weights calculated based on search engine importance values	Doesn't consider the inherent imprecision associated with decision maker's perspective

MetaSurfer	2014	FAHP and modified EOWA	Slightly different heuristic for missing documents, linguistic comparisons are made, linguistic importance degrees	Does not consider the dynamic nature of the Web, Documents are ranked just on the basis of search engine preferences

Chapter 3: PROPOSED INTELLIGENT OWA OPERATOR FOR MULTICRITERIA DECISION MAKING

The OWA operator proposed by Yager[12] has evolved over the years in order to overcome the limitations present in one another. The most recent development in OWA operators, named AOWA operator, was made by Suo[25] in 2012. The AOWA operator uses fixed linguistic importance degrees for the criteria in the MCDM problem, which are then defuzzified using COG method.

The importance degrees of the criteria, however, may change due to several changes in the environment with time. For example, as the cost of solar powered cells decreases due to advancements in technology, the importance of considering the cost of cells will decrease when selecting the type of cell for power provision. This study proposes an Intelligent OWA (In-OWA) operator to deal with such changes in importance degrees of criteria.

The In-OWA operator is based on the traditional OWA operator, and allows for the adaptation of the decision making process by considering the changing importance degrees. This chapter describes the operation of In-OWA operator and presents an example explaining its application to MCDM problems. This chapter also provides a comparison between In-OWA operator and the previous OWA operators, in terms of the way in which importance degrees are assigned.

3.1 Working of the In-OWA operator

The In-OWA operator allows for the decision making process to be adaptable and obtains the importance degrees of criteria through learning with examples. The importance degrees are thus, not assigned by experts but learned and are capable of reflecting the changes in environment. The learning algorithm for importance degree calculation in In-OWA operator is described next.

Training Examples:

A training example consists of the ranking according to each criteria and the optimal ranking of the alternatives for the given MCDM problem. A sufficient number of training examples are considered to form the training dataset, which is used to calculate the importance degrees of the criteria.

Learning Algorithm:

Let A_i be the i^{th} alternative in the optimal ranking of alternatives, i.e. A_1 is at rank 1 in the optimal ranking, n be the total number of alternatives, m be the total number of criteria, and $\sum(n)$ is the sum of all numbers through 1 to n.

Let Wj denote the weight of criteria Cj.
for j=1 to m
    Set W_j to initial value 0.
end
for i=1 to n
    Identify the criteria Cj with the highest ranking of the alternative A_i.
    Update $W_j \rightarrow W_j + (n - i + 1)$
end
for j=1 to m
    Normalize $W_j \rightarrow W_j / \sum(n)$
end

The learning algorithm presented above will be used to obtain the weights of all criteria for a single training example. The cumulative importance degree, Ij of the j^{th} criteria, Cj, is evaluated by taking the mean of the weights obtained by running the algorithm over the entire training set.

Note that, for each alternative, A_i, in the optimal ranking the criteria which ranks it the highest, i.e. the criteria which favours that alternative the most, is considered since the

assignment of higher weight to that criteria would cause the alternative A_i to achieve higher ranks. Also, since the alternatives are considered from top to bottom in the optimal ranking, higher weights will be assigned to criteria which favour an alternative at higher rank in optimal ranking.

3.2 Illustrative Example

An example is presented in this Section to elucidate the calculation of importance degrees in In-OWA operator. Consider three criteria C_1, C_2, C_3 and five alternatives a_1, a_2, a_3, a_4, a_5. Let a training example be specified as:

Optimal ranking – $[a_1,a_2,a_3,a_4,a_5]$,
Ranking according to C_1 alone – $[a_5,a_3,a_1,a_2,a_4]$,
Ranking according to C_2 alone – $[a_1,a_3,a_2,a_4,a_5]$, and
Ranking according to C_3 alone – $[a_2,a_3,a_1,a_5,a_4]$.

The weights of criteria, calculated using the learning algorithm, are –

$W_1 = 3 + 1 = 4$
$W_2 = 5 + 3 + 2 = 10$
$W_3 = 4 + 3 = 7$

The weights of criteria, calculated using the learning algorithm, are –

$W_1 = 4 / \sum(5) = 4/15 = 0.26$
$W_2 = 10 / \sum(5) = 10/15 = 0.66$
$W_3 = 7 / \sum(5) = 7/15 = 0.46$

As can be observed, the criteria C_2 has assigned the highest weight. This corresponds to the fact that the ranking according to C_2 is closest to the optimal ranking of alternatives. Thus, the highest importance should be assigned to C_2 for the specified training example.

Similarly, the weights are calculated for all the training examples in the training set, and the cumulative importance degree, I_j of the criteria C_j is calculated by taking the average of these evaluated weights for each training example.

3.3 Comparison of Learned with Linguistic Importance Degrees

The linguistic importance degrees used in the most recent OWA operator, AOWA operator and the learned importance degrees obtained by In-OWA operator are compared below:

i. The linguistic importance degrees are not capable of adaptation to accommodate the changes in environment. *The importance degrees calculated using the learning algorithm are adaptable and thus are better suited for rapidly changing environments*. Whenever such adaptation is needed, the algorithm is re-run using a new training dataset, in order to learn the cumulative importance degrees of criteria again.

ii. Another point to be noted is that the linguistic importance degrees are based on the intuition of an expert, whereas the proposed method uses a set of training examples to calculate the importance degrees. *Thus, even in the environments which rarely change, the In-OWA based approach is better than using linguistic importance degrees since it does not rely on the opinion of an expert, making the decision making process unbiased*.

Chapter 4: THE PROPOSED MODEL FOR METASEARCH

This chapter presents our proposed model for metasearch, MetaXplorer. This model is adaptable and free from biased expert opinion. *It uses FAHP and the proposed In-OWA operator for performing result aggregation*. Another important aspect of the model is it allows for URL analysis of retrieved documents instead of simply considering their ranks in search engines' result list. Therefore, MetaXplorer is an intelligent model as it performs analysis over the documents rather than just relying on different search engines to provide document preferences. This chapter also provides a summarization of improvements in MetaXplorer compared to the previous metasearch models in the end. The implementation details and evaluation of MetaXplorer will be discussed in chapters 5 and 6 respectively.

4.1 The Proposed Model: MetaXplorer

The proposed model for MetaXplorer consists of two phases: training phase and query execution phase. The training phase learns the cumulative importance degrees of underlying search engines through examples. Once the training phase is over and cumulative importance degrees of search engines are available, user can submit queries to MetaXplorer and obtain the consolidated result list in the query execution phase. The detailed explanation of the two phases is provided next.

4.1.1 Training Phase

In training phase, the training examples are fed as input to the training algorithm described in Chapter 3 considering search engines as criteria and documents as alternatives. Training algorithm calculates the cumulative importance degrees of underlying search engines as output. Each training example corresponds to a single query and consists of documents' ranking according to each search engine and optimal ranking of documents.

In the proposed model, we have considered two search engines, namely Google and Bing. Thus, the training phase will compute the cumulative importance degrees of Google, W_g and Bing, W_b as depicted in Figure 8.

Fig. 8: Training Phase

The training phase allows for the search engine importance degrees to be learned through examples and thus makes the proposed model adaptable to the changing environment. The training algorithm could be run periodically or as per user feedback in order to reflect the changes in the environment to result aggregation in MetaXplorer.

Note that the model can be scaled to incorporate more than two search engines for result merging and in that case the training phase will produce weights for those search engines as well.

4.1.2 Query Execution Phase

A typical invocation of MetaXplorer is depicted by the query execution phase. In this phase, a user submits a query to MetaXplorer. MetaXplorer then performs certain steps and returns a ranked list of documents pertaining to that query. The steps performed in query execution phase, as depicted in Figure 9, are:

i. *Preprocessing*:

The query submitted by user is preprocessed to create a refined query after removing redundant terms such as the articles, prepositions, sentence connectors, etc. A few words are replaced with better words in refined query, for example if the user query is "how to make tea", then in the refined query "how" is replaced such that it is translated to "make tea method".

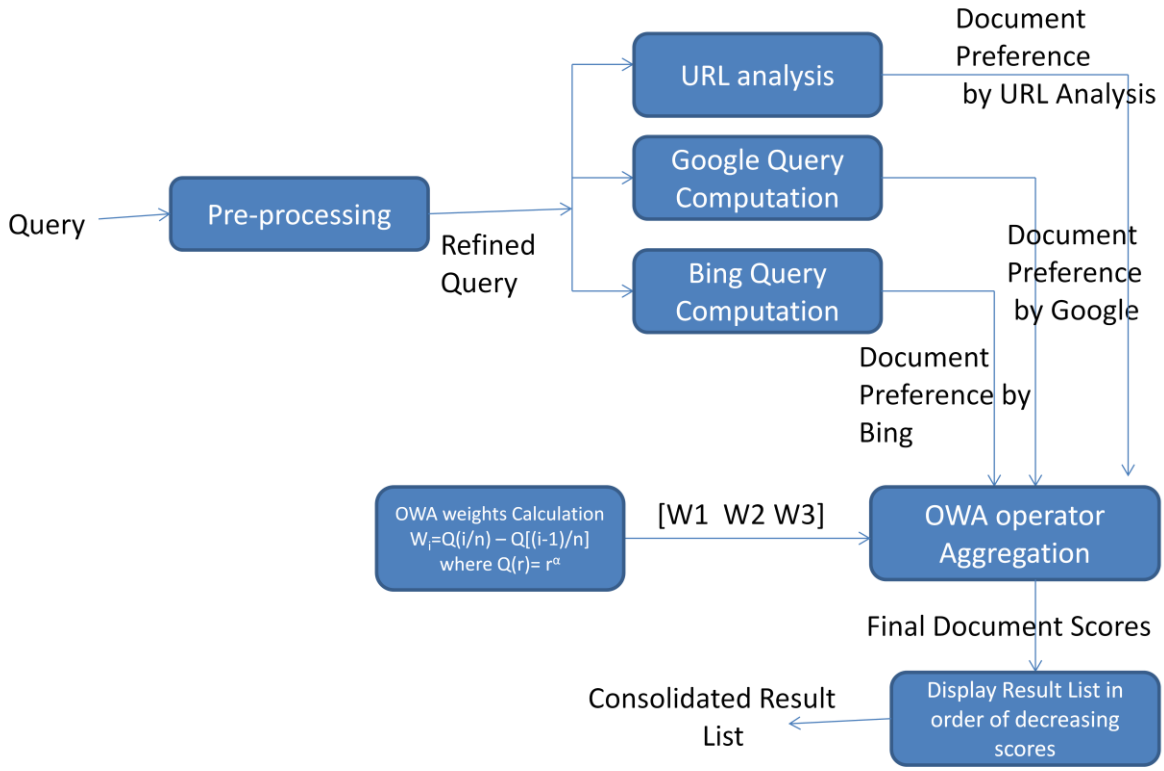

Fig. 9: Query Execution Phase

ii. *URL Analysis*:

URL analysis allows MetaSurfer to be intelligent since rather than just aggregating the returned result lists from different search engines, it analyses each document's URL in order to determine its relevance. URL analysis assigns weights, i.e. '*document preference*', to each document returned by Google and Bing. It works such that a higher weight is assigned to a more relevant document. Relevance of documents is determined by inspecting their URLs in order to deduce whether the URL belongs to a research paper or a patent or chapter in a text book. The contents of a research paper or a patent are

clearly more likely to be relevant compared to those found in a textbook. Similarly, a chapter in some textbook is likely to be more relevant than a dictionary website providing meaning to the query submitted. Also, by inspecting corresponding URLs we can infer the likelihood of a document referring to a research article, or a text book, etc.

For each document in the result lists, the document preference *W* is assigned by URL analysis, in the following manner:

- If the document most likely corresponds to a full text or abstract of a journal or conference paper or a patent, $W = 0.4$
- If the document most likely corresponds to a journal or conference homepage, $W = 0.3$
- If the document most likely corresponds to a book or database (such as Wikipedia) , $W = 0.2$
- If the document most likely represents other than above (dictionaries, company web pages, etc), $W = 0.1$

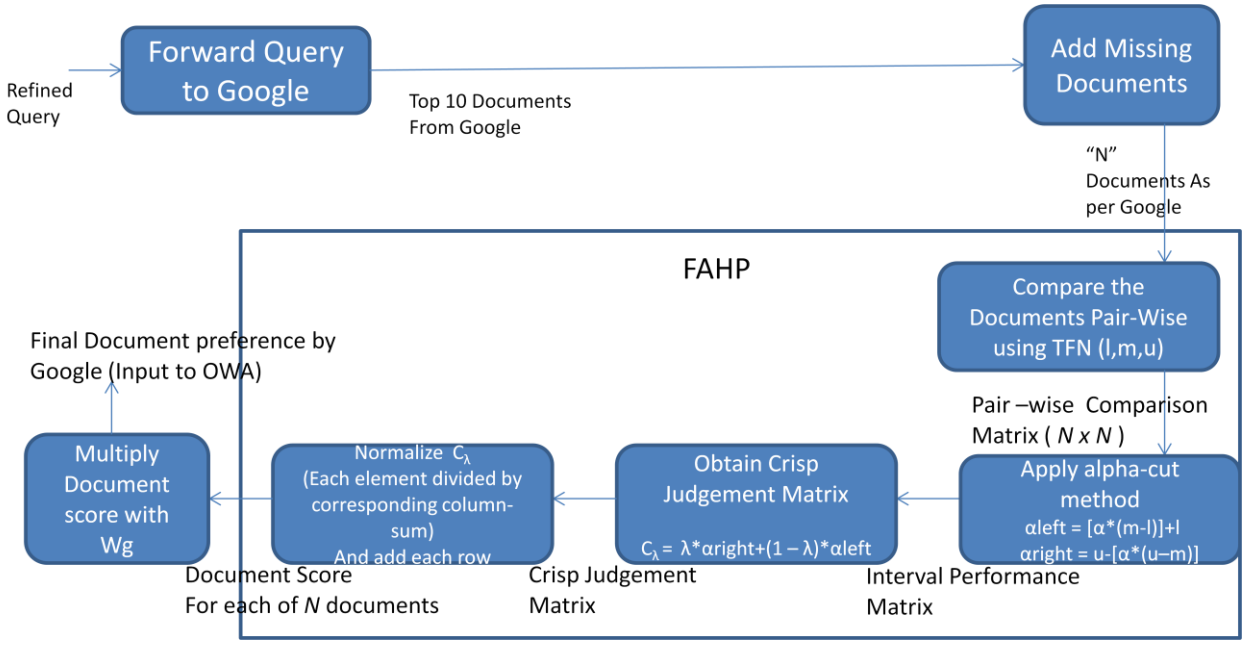

Fig 10: Google Query Computation

iii. *Google Query Computation*:

Google Query Computation consists of the following steps (Figure 10):

- The refined query is first forwarded to the Google search engine and top ten documents from the results are fetched.
- Missing documents in Google which are present in Bing are added to Google result list next to obtain a list of '*N*' documents as per Google, where *N* is the total number of unique documents in Google and Bing results taken together. Missing documents are added by taking weighted average of their positioning in each search engine result list. The weights used while taking weighted average are the corresponding search engine importance weights.

- Next step is to apply Fuzzy Analytical Hierarchy Process (FAHP) [30] in order to evaluate document scores. First, the documents are compared pair-wise by forming an $N \times N$ pair-wise comparison matrix using the linguistic variables below. The triangular fuzzy numbers (TFN) corresponding to the linguistic variables are depicted as (*l*,*m*,*u*), where *l* is left, *m* is middle and *u* is right component of the TFN.

Least Important (LTI)	(1,1,3)
Less Important (LSI)	(1,3,5)
Equally Important (EI)	(3,5,7)
More Important (MEI)	(5,7,9)
Most Important (MSI)	(7,9,9)

- Alpha-cut method is applied next to calculate the interval performance matrix, as per FAHP. Each element of interval performance matrix, [$\alpha left$, $\alpha right$] is evaluated using the following formula:

$$\alpha left = [\alpha * (m - l)] + l$$
$$\alpha right = u - [\alpha * (u - m)]$$

where, α is the confidence factor and $\alpha \in [0,1]$.

- Crisp judgment matrix, C_λ, is computed next as:

$$C_\lambda = \lambda * \alpha right + (1 - \lambda) * \alpha left$$

where, λ is the optimism index of the decision maker and $\lambda \in [0,1]$

- Next step is to normalize C_λ by dividing each element by corresponding column's sum. Following this, the document score, d_i, for each of the 'N' documents is determined by adding each row of the normalized matrix.

- The document scores are multiplied with importance degree of Google, W_g, to compute the final document preference, DP_i, by Google.

$$DP_i = W_g * d_i$$

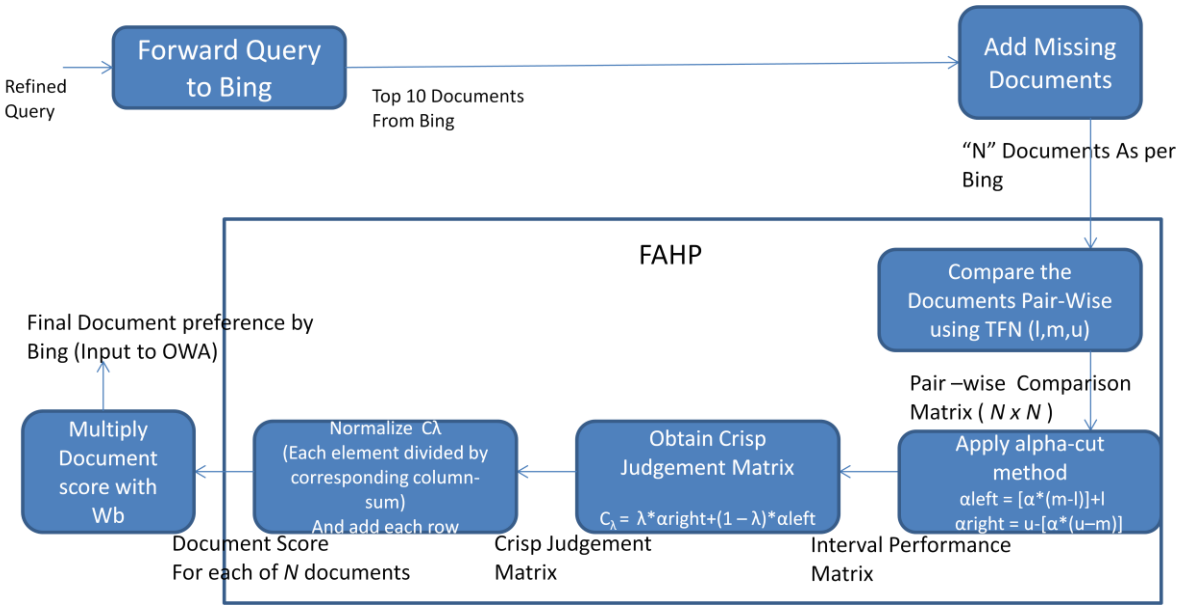

Fig. 11: Bing Query Computation

iv. *Bing Query Computation*:

Bing query computation also involves the use of FAHP to obtain final document preferences according to Bing search engine, and is similar to Google query computation as shown in Figure 11.

v. *OWA Weights Calculation*:

The re-ordering weights for OWA operator are calculated using quantifier-guided approach, using the following equation:

$$W_i = Q(\frac{i}{n}) - Q(\frac{i-1}{n}) \quad (12)$$

where, $Q(r) = r^{\alpha}$, n is the number of criteria i.e. 3 (two search engines and one for URL analysis) in this case, $\alpha \in [0,1]$.

vi. *OWA operator Aggregation*:

OWA operator is used next to obtain the final document scores, DS_i, and the documents result list is displayed to the user in the order of decreasing scores.

$$DS_i = W_1 * b_1 + W_2 * b_2 + W_3 * b_3$$

where, W_j is the j^{th} OWA weight and b_j is the j^{th} largest document preference, DP, value for the i^{th} document with j = 1,2,3 for document preferences by Google, Bing and URL analysis.

4.2 Advantages of MetaXplorer over Previous Models

The major advantages of the proposed model, MetaXplorer, over previous models described in Chapter 2 are presented below:

i. MetaXplorer handles the dynamic nature of the Web and adapts accordingly using the training algorithm in the proposed In-OWA operator.

ii. It is free from the biased opinion of a decision maker in the assignment of importance degrees to search engines.

iii. It performs URL analysis over the set of documents returned by search engines, rather than only taking into account their ranks, in search engine result lists, while result aggregation.

iv. MetaXplorer automates the calculation of search engine importance degrees, in the training phase, which had been a manual process involving search engines' performance inspection by experts till now.

v. User feedback is considered during adaptation. If the users feel results are not satisfying they can provide feedback, this would cause the training phase to be executed if a considerable number of user requests are received.

Chapter 5: IMPLEMENTATION

This chapter provides the implementation details of the proposed metasearch model, MetaXplorer. The detailed explanation pertaining to implementation can be divided into three sections. The first section provides a brief description of MetaXplorer's implementation platform, the second section discusses the implementation details of training phase and the third section presents the implementation details corresponding to query execution phase.

5.1 Brief Description

The proposed MSE, MetaXplorer, is implemented using JAVA EE 6 and MATLAB R2008b platform. Netbeans IDE is used for:

- Dispatching user query to the underlying search engines, namely Google and Bing, and retrieving the top ten documents from each.
- Designing the interface of MetaXplorer.
- Training the metasearch engine, MetaXplorer.
- Performing URL analysis.
- Adding missing documents in the underlying search engines' result lists.
- Displaying the final ranking of documents on the results page.

MATLAB R2008b is made to interact with the java program developed in Netbeans IDE, using MatlabControl Java API [46], and is used to evaluate the ranking of documents by applying FAHP and In-OWA operator.

The JAR (Java Archive) files used in the development of MetaXplorer are:

- *MatlabControl 4.1.0* [46] to allow MATLAB to be invoked through java program.

- *Google API Services Custom Search 1.20.0* [47] to allow queries to be forwarded to Google in order to get the results.
- *Azure Bing Search Java 0.12.0* [48] to send user queries to Bing and get the results.
- *HttpClient 4.1* [49] to resolve dependencies caused by Google Custom Search API.
- *HttpMime 4.1* [50] to resolve dependencies caused by Google Custom Search API.
- *HttpCore 4.1* [51] to resolve dependencies caused by Google Custom Search API.
- *Org Apache Commons codec* [52] to resolve dependencies caused by Bing Search API.
- *Org Apache Commons logging* [53] to resolve dependencies caused by Bing Search API.
- *Org Apache Commons net 3.3* [54] to resolve dependencies caused by Bing Search API.

5.2 Implementation of Training Phase

MetaXplorer is trained considering ten examples consisting of document rankings according to Google, Bing and optimal ranking. The ten example queries used for training the model are listed below:

i. Ontology
ii. Cryptography
iii. Data Mining Techniques
iv. Job Scheduling
v. Deep Learning
vi. Prediction Neural Network
vii. Information Retrieval
viii. Biogeography Based Optimization
ix. Genetic Algorithm, and
x. Remote Sensing

The working of training algorithm described in Chapter 3 can be demonstrated with the help of an example. Consider the example query '*ontology*'. The training example with ranking according to Google, Bing and optimal ranking is depicted in Figure 12. In this training example, 10 documents' ranking according to Google, 10 documents' ranking according to Bing and Optimal ranking of 14 documents (after considering missing documents) are provided.

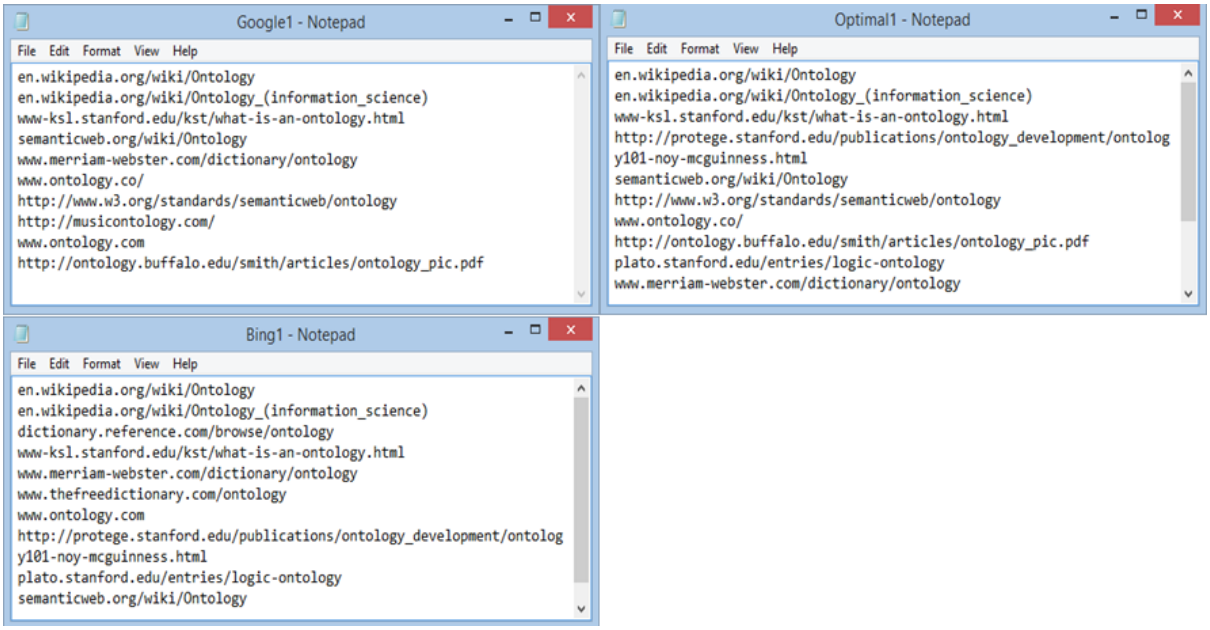

Fig. 12: Training Example - 'Ontology'

Initially, the search engine importance degrees for both Google and Bing are set to 0, i.e. $W_g = W_b = 0$. The training algorithm begins with the topmost document in optimal ranking, i.e. '*en.wikipedia.org/wiki/Ontology*'. As can be observed from Figure 13, the document is placed at rank 1 by both Google and Bing. Thus, the value 14, (14 – 1 + 1), is added to both W_g and W_b since both Google and Bing favour the document equally. Therefore, $W_g = W_b = 14$.

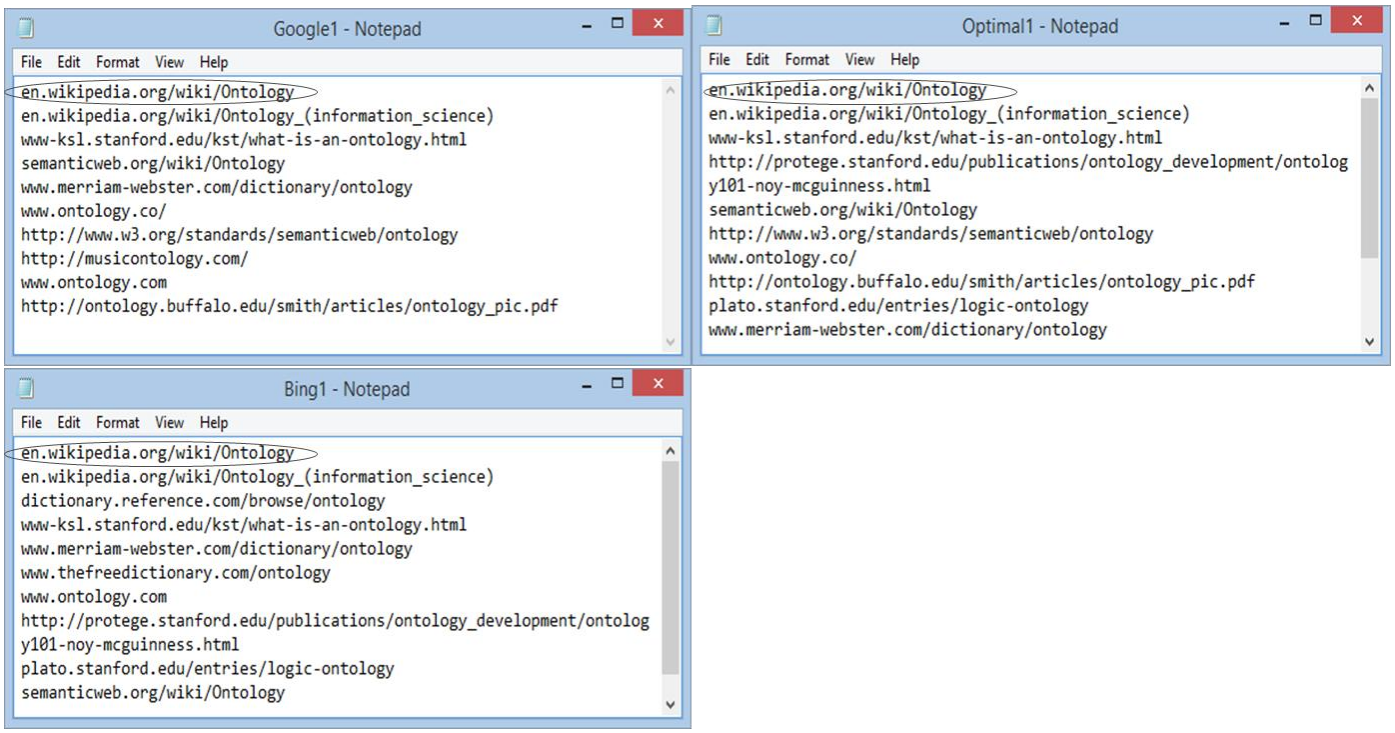

Fig. 13: Topmost Document Processing while Training

Next, the second document in the optimal ranking is considered and its rank in both Google and Bing is determined. Both Google and Bing have the second document at rank 2 and thus 13, (14 – 2 + 1), is added to both W_g and W_b. Therefore, $W_g = W_b = 14 + 13 = 27$. Similarly, third document in the optimal ranking is present at ranks 3 and 4 in Google and Bing respectively. Thus, the third document is favoured by Google, and the value 12, (14 – 3 + 1) is added to W_g. Therefore, W_g becomes 39 whereas W_b remains unchanged, i.e. 27.

The same process is repeated until all the 14 documents in the optimal ranking have been traversed. The final values of W_g and W_b obtained as a result, for the example '*ontology*' are 82 and 55 respectively. The values of W_g and W_b are then normalized. Therefore, the cumulative importance degrees for Google and Bing obtained through the training example '*ontology*' are:

$$W_g = W_g / \sum(14) = 82/105 = 0.78$$

$$W_b = W_b / \sum(14) = 55/105 = 0.52$$

where, $\sum(14)$ denotes sum of all numbers from 1 through 14.

The cumulative importance degrees, obtained as a result of training through all the ten examples, for Google and Bing are 0.6198741480397827 and 0.5236731251204936, respectively. These importance degrees are used in query execution phase to merge the result lists from Google and Bing effectively.

An important observation to be noted is that as we proceed from top to bottom along the optimal ranking, the magnitude by which importance degrees W_g and W_b are being updated is decreasing steadily. Consider the example provided above for '*ontology*'. Initially, for the topmost document the magnitude which is to be added is 14, for the second document it is 13, for the third document it is 12, and so on. *Therefore, if a document appears higher in the optimal ranking the importance degree of the search engine favouring it will be updated by a higher magnitude.*

5.3 Implementation of Query Execution Phase

This section describes the implementation work corresponding to the query execution phase. First we describe the implementation aspect of URL analysis and then an explanation of implementation details for the computations involved when user submits a query is provided.

5.3.1 URL Analysis Implementation Details

URL analysis assigns '*document preference*', *W*, to each document returned by forwarding the query to underlying search engines, Google and Bing. The value of document

preference, *W*, is a measure of how likely a document is to be relevant. By analyzing document URLs, MetaXplorer predicts whether that document corresponds to a research paper or a patent or a chapter from some textbook. As discussed before, if the document corresponds to research paper, it is more likely to be relevant compared to if it refers to a chapter in a textbook.

Consider an example of ScienceDirect [55] which is a leading resource for technical, scientific and medical research work. By closely inspecting the URLs from ScienceDirect, we can conclude that all the research paper articles contain the string "sciencedirect.com/science/article" as a part of their URLs. Similarily, all the references to journal homepages comprise of "sciencedirect.com/science/journal" and those corresponding to books contain "sciencedirect.com/science/book" as a part of their URLs. Therefore, we can deduce whether a URL from ScienceDirect belongs to a journal article, journal homepage or a book.

Similarly, we have constructed a checklist by inspecting several URLs' formation and used it to deduce whether the document in question corresponds to a research article or a patent or a textbook or none of them.

5.3.2 Query Computation Implementation Details

A user-friendly interface is designed to allow a user to submit a query to MetaXplorer. Figure 14 shows the interface of MetaXplorer where user enters the query. As soon as '*Search*' button is clicked, MetaXplorer refines the submitted query and forwards it to the search engines Google and Bing. Google Custom Search API [47] is used to send the refined query to Google and get the results. Similarly, Bing Search API [48], published by Microsoft, is used to get the results from Bing. Both the services are available according to different payment plans based on the number of search queries sent. Bing Search API is freely available for the limit of 5000 transactions/month and Google Custom Search API is free of charge allowing 100 search queries per day.

Fig. 14: MetaXplorer User Interface

The top ten results, i.e. web document URLs, from Google and Bing are stored in two files. The result aggregation is achieved by invoking MATLAB, using MatlabControl API, and an instance of it for the query '*Hepatology*' is shown in Figure 15. As can be seen, MATLAB R2008b is invoked and the final documents' ordering is computed using MATLAB.

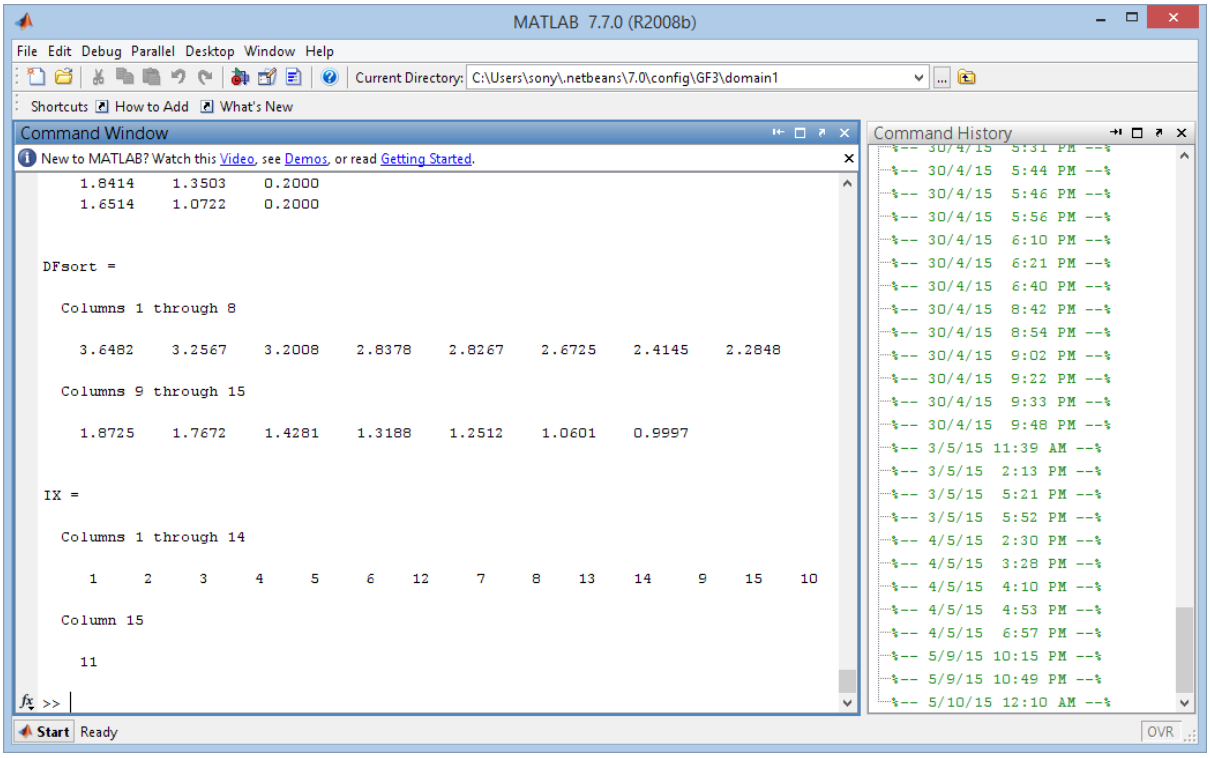

Fig. 15: Result Aggregation in MATLAB for 'Hepatology'

Finally, the consolidated result list for the same query, '*Hepatology*', is displayed in MetaXplorer's interface, as shown in Figure 16. The consolidated result list is presented as an ordered list of documents' titles, URLs and snippets, corresponding to the query submitted.

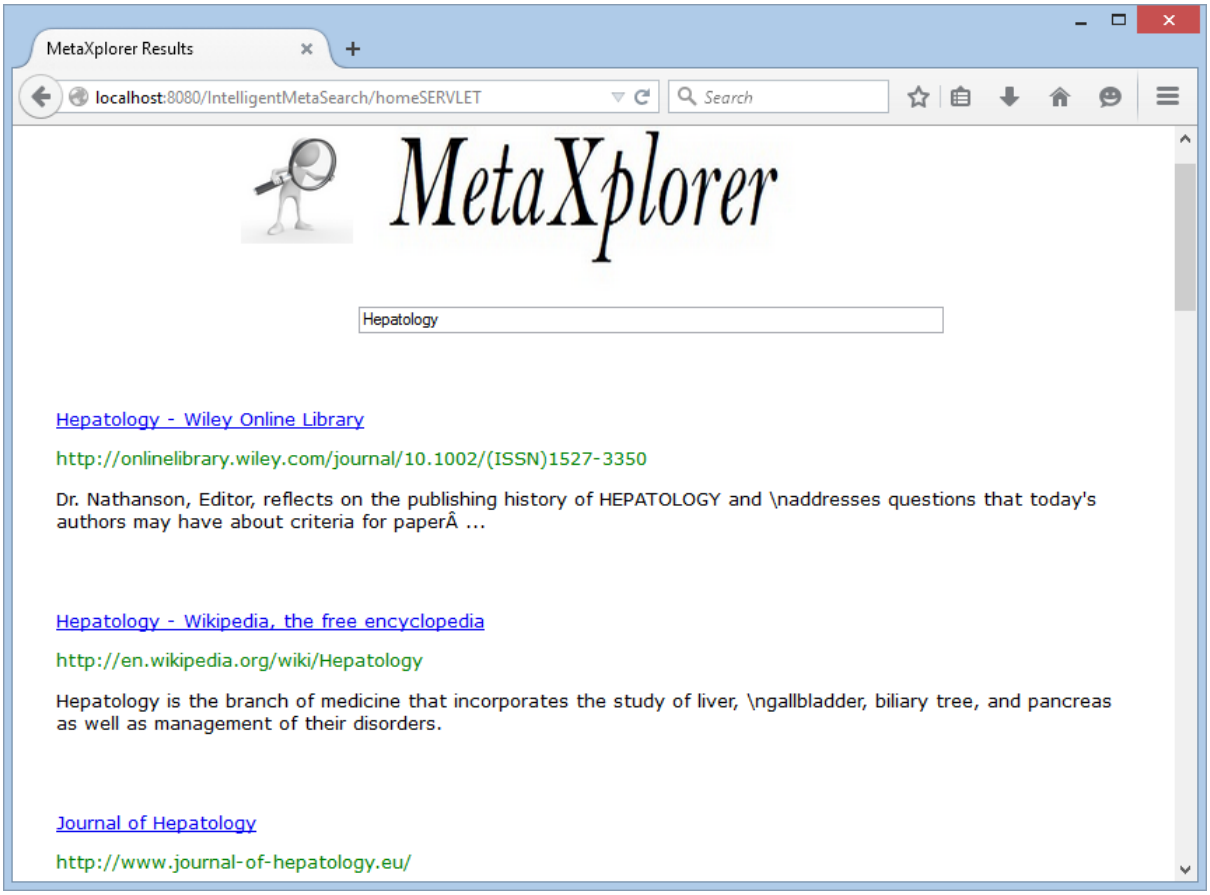

Fig. 16: Final Result List for 'Hepatology'

Chapter 6: EVALUATION AND RESULTS

This chapter discusses evaluation of the proposed model, MetaXplorer. This chapter is divided into three sections. The first section explains subjective evaluation of MetaXplorer by comparing it with previous models. The second section describes performance evaluation of MetaXplorer by comparing it with popular existing MSEs. The third section presents the results of evaluation.

6.1 Subjective Evaluation of MetaXplorer

In this section, a comparison of the proposed model, MetaXplorer, with other models for result aggregation in metasearch is presented. Table 4 compares MetaXplorer with MetaSurfer [1], T-norm Importance Guided Hybrid Fuzzy model [39] and OWA model [26]. As can be observed, MetaXplorer contains many new features which were not defined in the previous models.

Table 4: Comparison of MetaXplorer with Other Models

Evaluation Criteria	**MetaXplorer**	**MetaSurfer**	**T-norm Importance Guided Hybrid Fuzzy Model**	**OWA Model**
Adaptable to changing environment of the Web	√ (supports training)	×	×	×
Underlying techniques used	FAHP and proposed In-OWA operator	FAHP and modified EOWA operator	AHP and T-norm OWA operator	OWA operator

Unbiased decision making process	√ (importance degrees are learned and not assigned by experts)	×	×	×
Criteria other than search engine result ranking	√ (URL Analysis)	×	×	×
Missing Documents	Handled by taking weighted mean	Handled by weighted mean	Handled by computing mean	Handled by taking mean
Automatic Search Engine Importance Degrees Assignment	√	×	×	×
Performance Evaluated against real MSEs	√ (Dogpile, Excite, Webcrawler over 30 queries)	√ (Mamma, Webcrawler, Excite over 14 queries)	×	×

Note that the T-norm Importance Guided Hybrid Fuzzy model and OWA model were evaluated using TREC datasets which doesn't allow for them to be tested against real environment, with the MSEs available on the Web. MetaSurfer was evaluated against existing MSEs but over a smaller set of test queries compared to MetaXplorer.

6.2 Performance Evaluation of MetaXplorer

This section describes the performance evaluation of MetaXplorer. We measure the performance of MetaXplorer by calculating precision which is defined as ratio of retrieved relevant documents to the total number of retrieved documents (Equation 13).

$$Precision = \frac{The\ number\ of\ retrieved\ relevant\ documents}{The\ number\ of\ retrieved\ documents} \quad (13)$$

A set of test queries is considered. For each test query, the documents in the result list of the MSEs being evaluated are inspected in order to determine their relevance. We consider relevance of a document to be binary, i.e. a document can be either '*relevant*' or '*not relevant*'. A document is considered to be '*relevant*' if its contents provide some useful information on the test query; otherwise it is regarded as '*not relevant*'. Duplicate contents and documents which do not exist on servers, i.e. badly formed URLs, are considered to be '*not relevant*'.

Precision is then calculated for the test query using Equation 13. The values of precision for each test query are averaged over the entire set of test queries to obtain the average precision value which is used as a performance indicator. The performance of MetaXplorer is compared with three popular MSEs available on the Web namely, Dogpile [56], Excite [57] and WebCrawler [58].

A set of 30 test queries is considered for performance evaluation. The set of test queries is formulated from a wide range of research areas such as physics, computer science, biology, energy studies, chemistry, etc. The 30 test queries used are listed below:

- *Ad hoc Mobile Network Attacks*
- *Applied Nanoscience*
- *Hepatology*
- *Noise Filtering*

- *Nuclear Science and Technology*
- *Oncology*
- *Ordered Weighted Averaging operator*
- *Power System Modelling*
- *Web Crawling Techniques*
- *Microstrip Antenna*
- *Steganography Algorithms*
- *Machine Learning*
- *Pharmacogenetics*
- *Predicting Druggability using Machine Learning*
- *Text Mining*
- *Security in Cloud Computing*
- *Cellular Imaging Techniques*
- *Exergy Analysis of Solar Energy*
- *Protein Motion Simulation*
- *Homeostasis*
- *Protein-DNA Interaction*
- *Cosmochronology*
- *Nano-fabrication Techniques*
- *Quantum Hall Effect*
- *Photocatalysis*
- *Liquid Crystals*
- *Image Processing*
- *Amino Acids*
- *Fuzzy Inference System*
- *Metabolomics*

The top 10 results from MetaXplorer, Dogpile, Excite and WebCrawler are analyzed to determine their relevance to each test query. Average precision is then computed over the entire set of 30 test queries and comparisons are made.

6.3 Results

This section presents the results of evaluation of the proposed model, MetaXplorer. The evaluation was performed by comparing MetaXplorer with three popular MSEs namely, Dogpile, Excite and WebCrawler over a set of 30 test queries. Figure 17(a), 17(b), 17(c) and 17(d) show the relevance of some of the documents returned by MetaXplorer, Dogpile, Excite and WebCrawler respectively, for the test query '*Cosmochronology*'. Relevance of results is determined by inspecting each document and analysing whether its contents provide information on the query being considered.

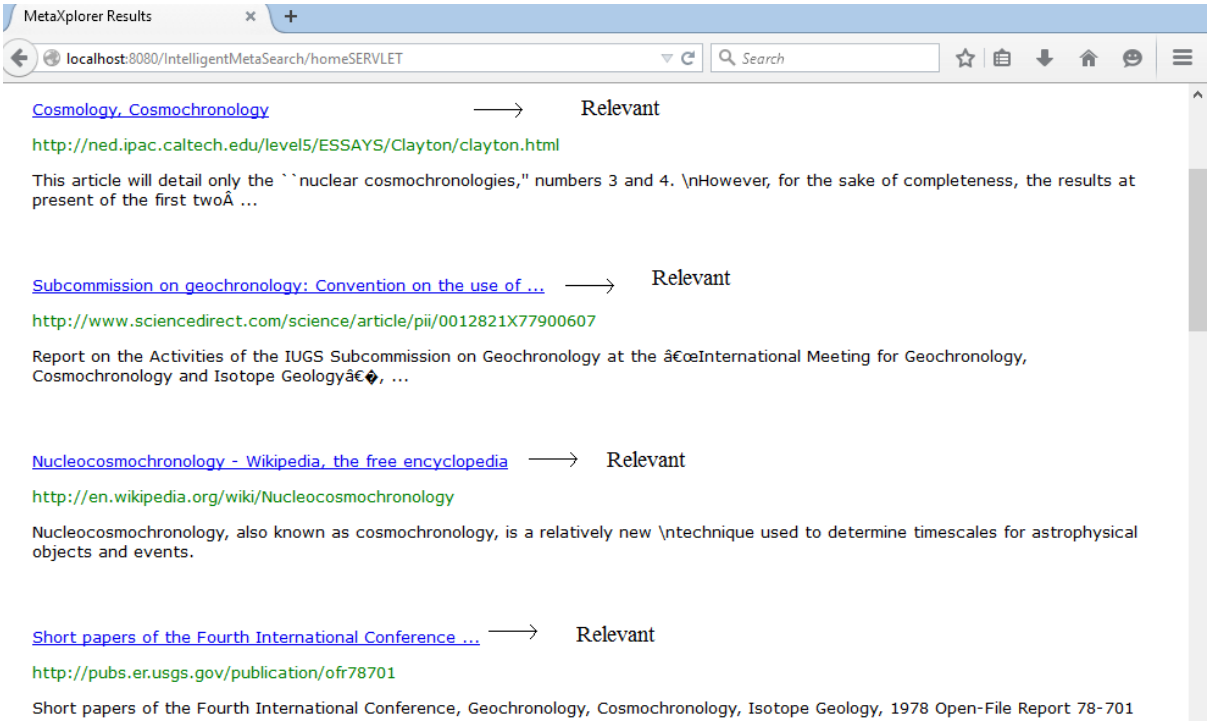

Fig. 17(a): Relevance of MetaXplorer Results for 'Cosmochronology'

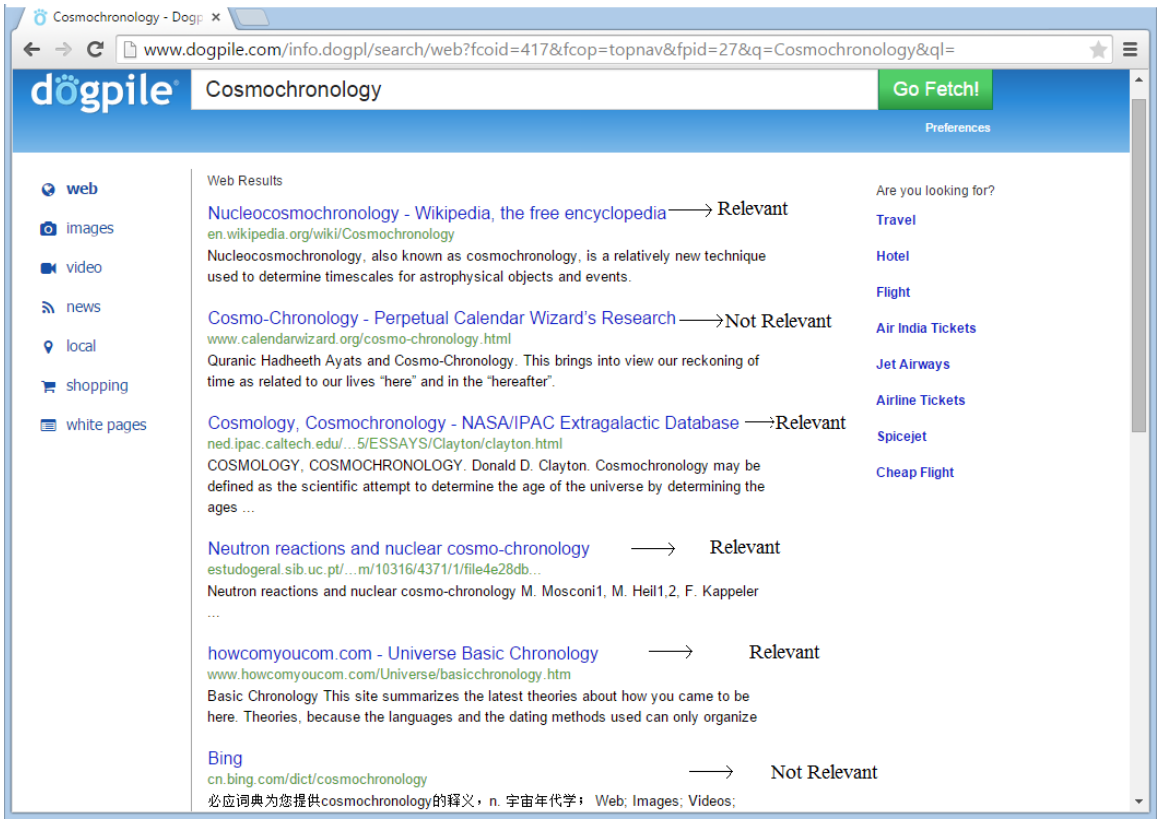

Fig. 17(b): Relevance of Dogpile Results for 'Cosmochronology'

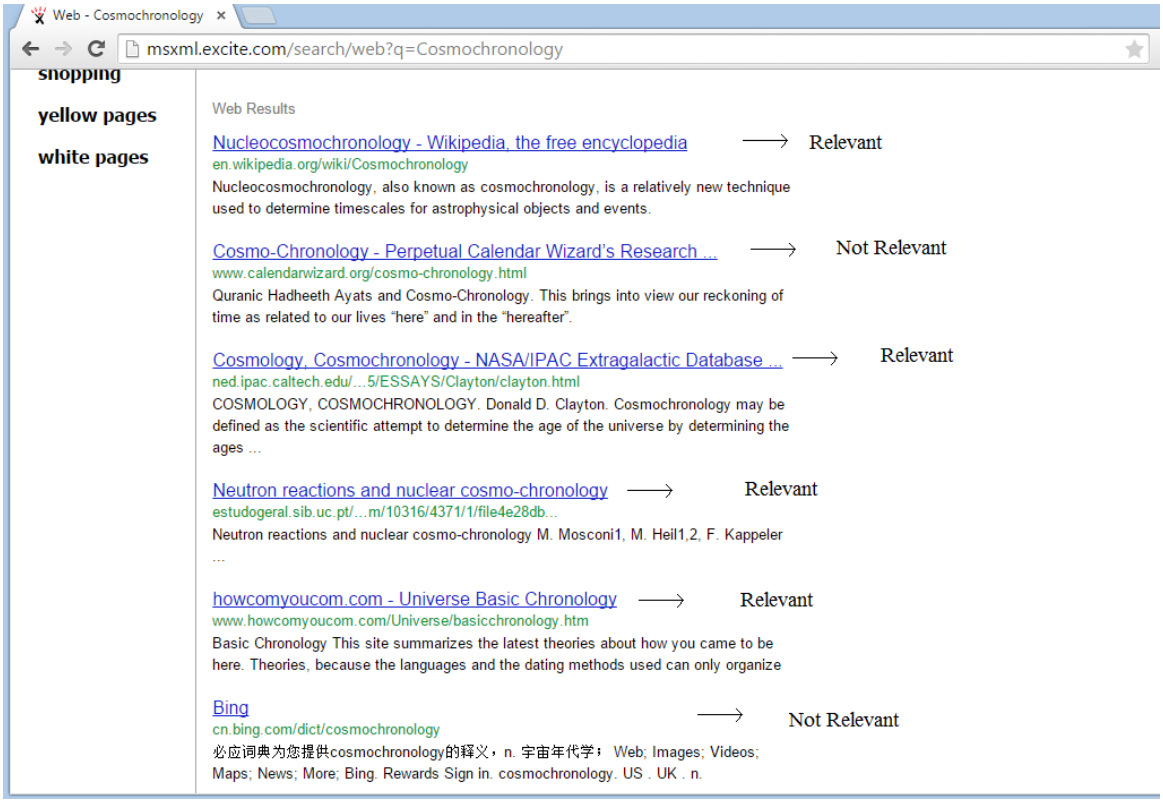

Fig. 17(c): Relevance of Excite Results for 'Cosmochronology'

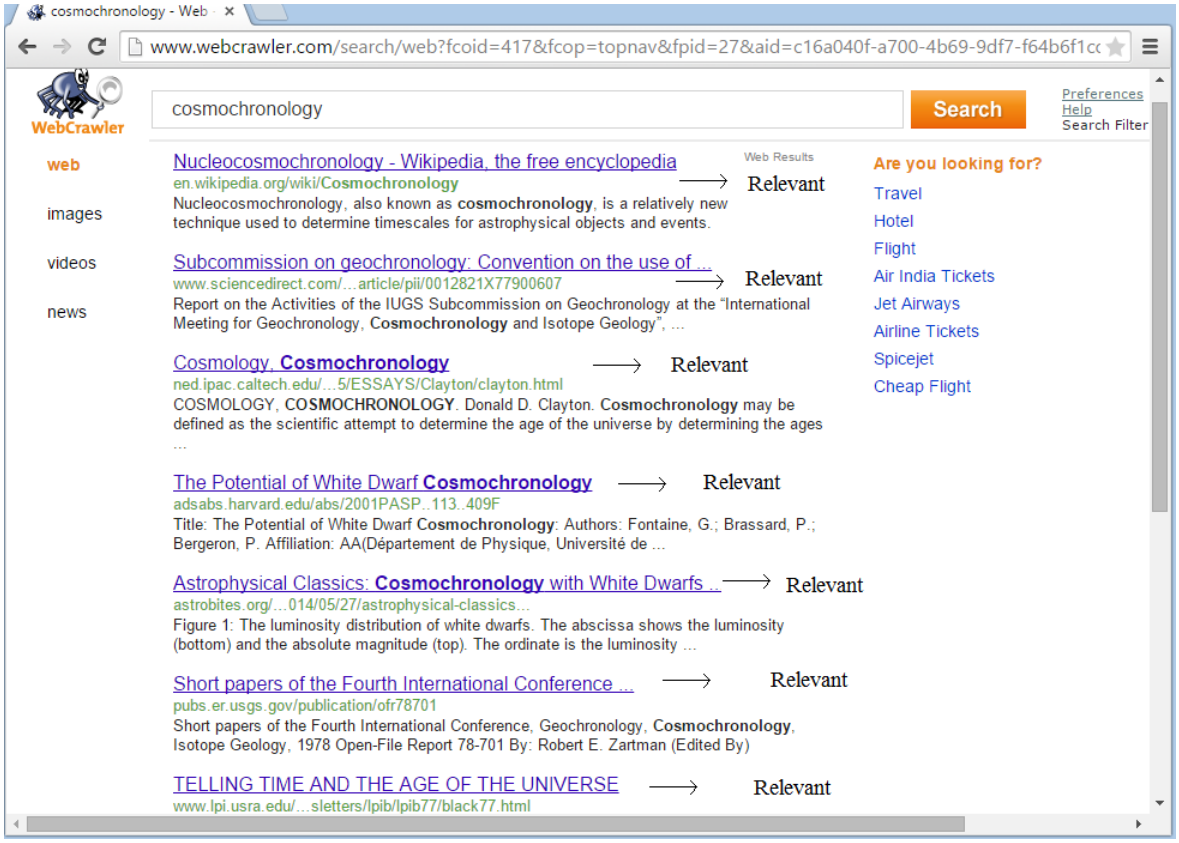

Fig 17(d): Relevance of WebCrawler Results for 'Cosmochronology'

Figure 18 shows the graphical comparison of the precision values obtained by analyzing the relevance of top 10 documents retrieved over the set of 30 test queries. The X-axis of the graph represents test queries and the Y-axis represents precision. The colours blue, red, green and purple represent MetaXplorer, Dogpile, Excite and WebCrawler respectively. It can be seen from the graph that MetaXplorer performs better than other MSEs in most of the cases.

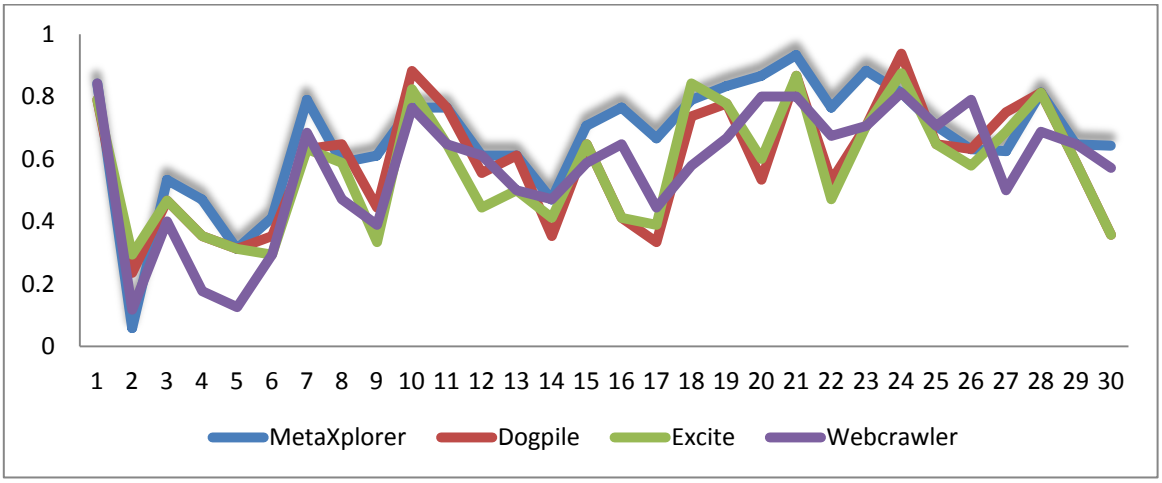

Fig. 18: Comparison of precision of MetaXplorer, Dogpile, Excite and WebCrawler over 30 test queries

The average precision values calculated over the set of 30 test queries for MetaXplorer, Dogpile, Excite and WebCrawler are depicted in Table 5.

Table 5: Average Precision over 30 Test Queries

Metasearch Engine	**Average Precision**
MetaXplorer	0.6641
Dogpile	0.5887
Excite	0.5723
WebCrawler	0.5694

As can be seen from the table, MetaXplorer has the highest average precision of 0.6641, followed by Dogpile (0.5887), Excite (0.5723) and WebCrawler (0.5694), respectively. Therefore, performance analysis shows that MetaXplorer has the best performance when compared with three popular MSEs on the Web namely, Dogpile, Excite and WebCrawler.

Chapter 7: CONCLUSIONS AND FUTURE WORK

This chapter discusses the conclusions inferred from this research and presents the possibilities of extension of this work in future.

This research work proposes a new model, *MetaXplorer*, for result aggregation in metasearch. This thesis also proposes a new Ordered Weighted Averaging operator, In-OWA (Intelligent OWA) operator, which can be applied to solve any multicriteria decision making problem. The use of In-OWA operator makes the decision making process free from the biased opinion of an expert and allows for adaptability in terms of changing environment. The proposed model *MetaXplorer*, based on the proposed In-OWA operator, is capable of adapting as the environment, i.e. the Web, changes and thus effectively handles the dynamic nature of the Web.

Also, *MetaXplorer* performs analysis over the set of documents returned by search engines, rather than simply working on the preferences given by underlying search engines, and hence is intelligent. It analyses the URLs of the documents retrieved in order to deduce their relevance in terms of the submitted query. A measure of this relevance is used in the result aggregation process to get final documents' ranking. *MetaXplorer* is designed, developed and successfully implemented.

A comparative analysis of *MetaXplorer* with previous metasearch models shows that it has many new features which were not presented in any of the previous models. The performance of *MetaXplorer* is evaluated by comparing its average precision over a set of 30 test queries with three popular MSEs on the Web namely, Dogpile, Excite and WebCrawler. The set of 30 test queries is formulated by considering topics evenly from different research areas. The results show that *MetaXplorer* has the highest average precision of 0.6641 when compared with the MSEs Dogpile, Excite and WebCrawler.

This research work can be extended in future by incorporating more search engines in result aggregation. Addition of more search engines will allow a greater number of documents to be retrieved. Therefore, for each query submitted the number of relevant documents returned is likely to increase. However, the number of irrelevant documents and duplicates will also increase as more search engines are considered. Thus, appropriate mechanisms for duplicate removal and limiting the number of documents displayed in final list would need to be established.

The relevance of documents, during performance evaluation, can be categorized as 'highly relevant', 'relevant' and 'not relevant', instead of binary classification used in this research work. This would allow the 'less relevant' documents as well to be considered in precision calculation. A more exact measure of precision will be derived since the 'less relevant' documents will also add a certain component to the precision, along with 'highly relevant' documents. Although note that the component added corresponding to 'highly relevant' documents should be larger than that corresponding to the 'less relevant' ones.

Chapter 8: PUBLICATIONS FROM THE RESEARCH

This chapter briefly states the communicated research paper from this research work, along with the details of the conference of publication.

1. Dimri N., Gupta D., "***MetaXplorer: An Intelligent and Adaptable Metasearch Engine using a Novel OWA operator***", 9th International Conference on Advanced Computing and Communication Technologies (ICACCT - 2015), Springer.

REFERENCES

[1] Devendra Tayal, Amita Jain, Neha Dimri, Shuchi Gupta, "MetaSurfer: a new metasearch engine based on FAHP and modified EOWA operator", International Journal of System Assurance Engineering and Management, Springer, 2014.

[2] Available Online at: http://en.wikipedia.org/wiki/Metasearch_engine

[3] R.M. Losee, "When information retrieval measures agree about the relative quality of document rankings", Journal of the American Society of Information Science, vol. 51, issue 9, pp. 834-840, 2000.

[4] Bar-Ilan, J., Mat-Hassan, M., and Levene, M., "Methods for comparing rankings of search engine results", Computer Networks, vol. 50, pp. 1448–1463, 2006.

[5] Spink, A. H., Jansen, B. J., Blakely, C., and Koshman, S., "A study of results overlap and uniqueness among major Web search engines", Information Processing & Management, vol. 42, issue 5, pp. 1379–1391, 2006. 32

[6] Spoerri, A., "Examining the authority and ranking effects as the result list depth used in data fusion is varied", Information Processing & Management, vol. 43, issue 4, pp. 1044–1058, 2007.

[7] Vaughan, L., "New measurements for search engine evaluation proposed and tested", Information Processing & Management, vol. 40, issue 4, pp. 677–691, 2004.

[8] Keyhanipour, A. H., Moshiri, B., Kazemian, M., Piroozmand, M., and Lucas, C., " Aggregation of Web search engines based on users' preferences in WebFusion", Knowledge-Based Systems, vol. 20, issue 4, pp. 321–328, 2007.

[9] Weiyi Meng, Clement Yu and King-lup Liu, "Building Efficient and Effective Metasearch Engines", ACM Computing Surveys, vol. 34, issue no. 1, pp. 48-89, 2002.

[10] Manoj M., Elizabeth Jacob, "Information Retrieval on Internet using meta-search engines: A review", Journal of Scientific and Industrial Research, vol. 67, pp. 739-746, 2008.

[11] David A. Grossman and Ophir Frieder, Information Retrieval – Algorithms and Heuristics, 2nd edition.

[12] Ronald R. Yager, "On Ordered Weighted Averaging Aggregation Operators in multicriteria Decisionmaking", IEEE Transactions on Systems, Man, and Cybernetics, vol. 18, pp. 183-190, 1988.

[13] Christer Carlsson, Robert Fullér, and Szvetlana Fullér, "OWA Operators for doctoral student selection problem", R.R.Yager and J.Kacprzyk eds., The ordered weighted averaging operators: Theory, Methodology, and Applications, Kluwer Academic Publishers, Boston, pp. 167-178, 1997.

[14] José M. Merigó and Anna M. Gil-Lafuente, "Using the OWA Operator in the Minkowski Distance", International Journal of Social and Human Sciences, vol. 2, 2008.

[15] Torra, V., "OWA operators in data modeling and reidentification", Fuzzy Systems, IEEE Transactions, vol. 12, Issue 5, pp. 652 – 660, 2004.

[16] Arijit De, Elizabeth E. Diaz and Vijay V. Raghavan, "On Fuzzy Result Merging for Metasearch",Fuzzy System Conference, IEEE, pp. 1-6, 2007.

[17] José M. Merigó, Anna M. Gil-Lafuente, "Decision Making with the OWA operator in sport management", Expert Systems with Applications, Volume 38, Issue 8, pp. 10408–10413, 2011.

[18] Ali Emrouznejad, "MP-OWA: The most preferred OWA operator", Knowledge-Based Systems, vol. 21, issue 8, pp. 847-851, 2008.

[19] Devendra Tayal, Neha Dimri, and Shuchi Gupta, "Evolution Of Ordered Weighted Averaging Operators And Their Role In Solving MCDM And GDM Problems", International Conference on Arificial Intelligence and Soft Computing, IIT-BHU, pp. 147-153, 2012.

[20] Francisco Chiclana, Francisco Herrera, and Enrique Herrera-Viedma, "The Ordered Weighted Geometric Operator: Properties and Application in MCDM problems", in Proc. 8th Conference on Information Processing and Management of Uncertainty in Knowledge based Systems (IPMU), pp. 985-991, 2000.

[21] Azcel, J., and Alsina, C., "Procedures for synthesizing ratio judgments", Journal of Mathematical Psychology, vol. 27, pp. 93-102, 1983.

[22] Azcel, J., and Alsina, C., "Synthesizing judgements: A functional equations approach", Mathematical Modelling, vol. 9, pp. 311-320, 1987.

[23] F. Chiclana a, E. Herrera-Viedma b, F. Herrera b, and S. Alonso, "Some induced ordered weighted averaging operators and their use for solving group decision-making problems based on fuzzy preference relations", European Journal of Operational Research, vol. 182, pp. 383–399, 2007.

[24] Zarghami, M., Ardakanian, R., Memariani, and A., Szidarovszky, F., "Extended OWA operator for group decision making on water resources projects", Journal of Water Resources Planning and Management, vol. 134, issue 3, pp. 266–275, 2008.

[25] M.Q. Suo, Y.P.Li, G.H.Huang, "Multicriteria decision making under uncertainty: An advanced ordered weighted averaging operator for planning electric power systems", Engineering Applications of Artificial Intelligence, vol. 25, issue 1, pp. 72-81, 2012.

[26] E. D. Diaz, A. De, V.V. Raghavan, "A Comprehensive OWA based Framework for Result Merging in Metasearch", 10th International Conference on Rough Sets, Fuzzy Sets, Data Mining, and Granular-Soft Computing, Canada, Springer, pp. 193-201, 2005.

[27] E. D. Diaz, "Selective Merging of Retrieval Results for Metasearch Environments", Ph.D. Dissertation, University of Louisiana, Lafayette, LA, 2004.

[28] T. L. Saaty, The Analytic Hierarchy Process, McGraw-Hill, New York, 1980.

[29] T.L. Saaty, "Relative Measurement and its Generalization in Decision Making: Why Pairwise Comparisons are Central in Mathematics for the Measurement of Intangible Factors - The Analytic Hierarchy/Network Process". Review of the Royal Spanish Academy of Sciences, Series A, Mathematics, vol. 102, 2, pp 251-318, December 2007.

[30] Dinesh. M.S, K.ChidanandaGowda and P.Nagabhushan, "Fuzzy Hierarchical Analysis for Remotely Sensed data", Geoscience and Remote Sensing Symposium Proceedings, IEEE, vol. 2, pp. 782-784, 1998.

[31] M.H. Vahidnia, A. Alesheikh, A. Alimohammadi, and A. Bassiri, "Fuzzy Analytical Hierarchy Process In GIS Application", The International Archives of the Photogrammetry, Remote Sensing and patial Information Sciences, Vol. 37, Part B2, Beijing, 2008.

[32] Yumei Chen, "Fuzzy AHP-based Method for Project Risk Assessment", Seventh International Conference on Fuzzy Systems and Knowledge Discovery, 2010.

[33] Hing Kai Chan, "An Extended Fuzzy-AHP Approach for the Evaluation of Green Product Designs", IEEE Transactions On Engineering Management, Issue 99, pp. 1-13, 2012.

[34] Feng Kong, Hongyan Liu, "Applying Fuzzy Analytic Hierarchy process To Evaluate Success Factors of E-Commerce", International Journal Of Information and System Sciences, vol. 1, pp. 406-412, 2005.

[35] Yu-Cheng Tang and Malcolm J. Beynon, "Application and Development of a Fuzzy Analytic Hierarchy Process within a Capital Investment Study", Journal of Economics and Management, vol. 1, pp. 207-230, 2005.

[36] Gwo-hshiung Tzeng, Min-Jiu Hwang, Jia-Horng Shieh, and Hsin-Chi Wu, "Applying Fuzzy AHP and Nonadditive Fuzzy Integral Methods for Evaluation and Selection of Construction Project Contractor", 6th ISAhP, 2001.

[37] Arijit De, Elizabeth Diaz, "Hybrid Fuzzy Result Merging for Metasearch Using Analytical Hierarchy Process", 28th North American Fuzzy Information Processing Society Annual Conference (NAFIPS), USA, IEEE, 2009.

[38] Arijit De and Elizabeth E. Diaz, "On the Role of t-norms on Hybrid Fuzzy Result Merging for Metasearch", 15th IEEE International Conference On Fuzzy Systems, Barcelona, Spain, IEEE Press, pp. 1-6, 2010.

[39] Arijit De and Elizabeth E. Diaz, "Fuzzy Search Result Aggregation using Analytical Hierarchy Process", Fuzzy Information Processing Society (NAFIPS), Annual Meeting of The North American, IEEE, pp. 1-6, 2011.

[40] Erik Selberg, Oren Etzioni, "Multi-Service Search and Comparison Using the MetaCrawler", Proceedings of the 4th International World Wide Web Conference, pp. 195-208, 1995.

[41] Erik Selberg, Oren Etzioni, "The MetaCrawler Architecture for Resource Aggregation on the Web", IEEE Expert, 1997.

[42] Available Online at: http://www.alexa.com/

[43] J. Aslam and M. Montague, "Models for Metasearch", Proceedings of the 24th annual international ACM SIGIR Conference on Research and Development in Information Retrieval, New Orleans, LA, USA, pp. 276-284, 2001.

[44] C. C. Vogt., "How much more is better? Characterizing the effects of adding more IR systems to a combination", Content-Based Multimedia Information Access (RIAO), Paris, France, pp. 457–475, 2000.

[45] W. Hersh, Chris Buckley, T. J. Leone, and David Hickam, "OHSUMED: An interactive retrieval evaluation and new large test collection for research", Proceedings of the 17th annual international ACM SIGIR Conference on Research and Development in Information Retrieval, ACM/Springer, New York, NY, USA, pp. 192-201, 1994.

[46] Available Online at: https://code.google.com/p/matlabcontrol/

[47] Available Online at: https://developers.google.com/custom-search/

[48] Available Online at: https://datamarket.azure.com/dataset/bing/search

[49] Available Online at: http://www.java2s.com/Code/Jar/h/Downloadhttpclient41jar.htm

[50] Available Online at: http://www.java2s.com/Code/Jar/h/Downloadhttpmime41jar.htm

[51] Available Online at: http://www.java2s.com/Code/Jar/h/Downloadhttpcore41jar.htm

[52] Available Online at: https://commons.apache.org/proper/commons-codec

[53] Available Online at: http://commons.apache.org/proper/commons-logging/

[54] Available Online at: https://commons.apache.org/proper/commons-net/download_net.cgi

[55] Available Online at: http://www.sciencedirect.com/

[56] Available Online at: http://www.dogpile.com/

[57] Available Online at: http://www.excite.com/

[58] Available Online at: http://webcrawler.com/