

**APPLICATION OF SOFT COMPUTING TECHNIQUES  
FOR EFFICIENT INFORMATION RETRIEVAL**

*A Dissertation submitted to the  
School of Computer & Systems Sciences,  
Jawaharlal Nehru University, New Delhi  
in partial fulfillment of the requirements for the award of the degree of*

**MASTER OF TECHNOLOGY  
IN  
COMPUTER SCIENCE AND TECHNOLOGY**

**BY  
AMIT KUMAR SINGH**

**UNDER SUPERVISION OF  
Dr. Aditi Sharan**



**SCHOOL OF COMPUTER AND SYSTEMS SCIENCES  
JAWAHARLAL NEHRU UNIVERSITY  
NEW DELHI-110067, INDIA  
JULY 2009**



# जवाहरलाल नॅहरू विश्वविद्यालय

JAWAHARLAL NEHRU UNIVERSITY

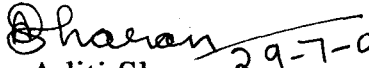
School of Computer & Systems Sciences

NEW DELHI- 110067, INDIA

## CERTIFICATE

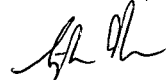
This is to certify that the dissertation entitled “**APPLICATION OF SOFT COMPUTING TECHNIQUES FOR EFFICIENT INFORMATION RETRIEVAL**” being submitted by Mr. **Amit Kumar Singh** to the School of Computer and Systems Sciences, **Jawaharlal Nehru University**, New Delhi, in partial fulfillment of the requirements for the award of the degree of **Master of Technology in Computer Science and Technology**, is a record of bonafide work carried out by him under the supervision of **Dr. Aditi Sharan**.

This work has not been submitted in part or full to any university or institution for the award of any degree or diploma.

  
Dr. Aditi Sharan. 29-7-09

(Supervisor)

SC&SS, JNU, New Delhi

  
(Dean, SC&SS,)

JNU, New Delhi



# जवाहरलाल नॅहरू विश्वविद्यालय

**JAWAHARLAL NEHRU UNIVERSITY**

**School of Computer & Systems Sciences**


**NEW DELHI- 110067, INDIA**

## **DECLARATION**

This is to certify that the dissertation entitled “**APPLICATION OF SOFT COMPUTING TECHNIQUES FOR EFFICIENT INFORMATION RETRIEVAL**” is being submitted to the School of Computer and Systems Sciences, Jawaharlal Nehru University, New Delhi, in partial fulfillment of the requirements for the award of the degree of **Master of Technology in Computer Science & Technology**, is a record of bonafide work carried out by me.

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

July 2009  
JNU, New Delhi

  
Amit Kumar Singh  
M.Tech., Final Semester,  
SC&SS, JNU, New Delhi.

## ACKNOWLEDGEMENTS

---

I am very glad to express my sincere gratitude and thanks to my supervisor **Dr. Aditi Sharan** for her expert guidance in completing this work. Without her constructive suggestions, critical comments, encouragement, help and kind cooperation, this dissertation would not have taken a shape. I feel it is a great privilege to have had the opportunity to work under her prestigious supervision. Her constant encouragement and support helped me a lot. She led me into the exciting field of Information Retrieval and taught me how to approach the problem in rigorous way. The methodology and philosophy that I have learned in my research will definitely benefit my career for life.

I wish to thank my lab mates, Mrs. Hazra Imran, Sonia, and Sonu Lal Gupta for their valuable help, comments and suggestions for this dissertation.

I am thankful to my seniors Gaurav Rajput, Ashu Singh, Nitin Jain and Sudheer Sharma for their guidance and assistance.

I am grateful to all the faculty members for providing motivation and time to time guidance.

Thanks are due to administration of JNU for providing a congenial environment for making our work a success. Their academic support has been a real asset in completing this work.

I am also thankful to Nikhil, Asif, Ajay, Suresh, Dilip, Surendra, Tanveer and all others for their discussions and help at all time during the course of my dissertation work.

Finally, I am most thankful to my parents and brothers for their unlimited love, care and encouragement. Over the years, they cheer for even a tiny progress I made and always have faith in me no matter how difficult life is.

Amit Kumar Singh

## **Abstract**

---

This dissertation work is an attempt to explore role of soft computing techniques in the field of information retrieval. Since the process of retrieving relevant information is full of uncertainty, imprecision and cannot be handled by hard computing, soft computing is drastically used in every field attached with IR. Moreover with increased use of WWW for searching information, soft computing tools have a larger role to play. Since information on web is full of uncertainty, imprecision and subjectiveness. We have focused on genetic algorithm, fuzzy set theory and neural network components of soft computing.

The dissertation is divided in five chapters. First chapter introduces information retrieval, soft computing and focuses on application of soft computing for information retrieval. Next three chapters are devoted for application of fuzzy set theory, neural network and genetic algorithm respectively in IR. Finally, chapter five concludes the work.

# Contents

	<b>Page No.</b>
Certificate	i
Declaration	ii
Acknowledgements	iii
Abstract	iv
<b>Chapter 1 Information Retrieval and Soft Computing</b>	<b>1-10</b>
1.1 Information Retrieval(IR): problems, need of research and tools used	1
1.2 Information Retrieval System	2
1.3 Information Retrieval Models	4
1.4 Information Retrieval Tasks	7
1.5 Soft Computing	8
1.6 Application of Soft Computing in IR	9
<b>Chapter 2 Application of Fuzzy Set Theory in IR</b>	<b>11-21</b>
2.1 Introduction to Fuzzy Set Theory	11
2.2 Application of Fuzzy Set Theory in IR	12
2.3 Fuzzy Similarity Measures	14
2.4 Concept Based IR using fuzzy set theory	16
2.4.1 Limitation of VSM in Identifying the Concepts	16
2.4.2 Methods of Finding Semantically Related Terms	17
2.4.3 Concept Based Model for IR Using Fuzzy Set Theory	17
2.5 Algorithm and Experiment for Concept Based IR	19
2.5.1 Algorithm for Identifying Concept Clusters	19
2.5.2 Experiment	20

<b>Chapter 3 Application of Neural Network in IR</b>	<b>22-31</b>
3.1 Introduction to Neural Network	22
3.1.1 Definition of Neural Network	22
3.1.2 Mathematical Model of Neural Network	23
3.2 Learning Process of Neural Network	24
3.2.1 Supervised Learning	24
3.2.2 Unsupervised Learning	26
3.2.3 Reinforcement Learning	26
3.3 Application of Neural Network Models in IR	26
3.3.1 Vector Space Model with Neural Networks	28
3.3.2 Relevance Feedback with Neural Nets	30
3.3.3 Learning Modification to the Neural Network	31
<b>Chapter 4 Application of Genetic Algorithm in IR</b>	<b>32-38</b>
4.1 Introduction to Genetic Algorithm (GA)	32
4.2 Application of Genetic Algorithm in IR	34
4.2.1 GA for Document Description	34
4.2.2 GA for Query Weight	35
4.2.3 GA for Matching Function	35
4.2.3.1 Proposed Algorithm and Experiments on Matching Function	37
<b>Chapter 5 Conclusion</b>	<b>39</b>
<b>References</b>	<b>40-45</b>
<b>Appendix 1</b>	<b>45-46</b>

# Chapter 1

## Information Retrieval and Soft Computing

---

### 1.1 Information Retrieval (IR): Problems, need of research and tools used

For last few decades, the problem of information storage and retrieval has become a big and attractive but challenging issue. With developments in hardware and software technologies vast amounts of information can be stored, accurate and speedy access of information is also possible but retrieving relevant information, especially in textual databases is still very difficult. These databases are widely used in traditional library science environments, in business applications (e.g. manuals, news letters and electronic exchanges) and in scientific applications (e.g. electronic community systems and scientific databases). In principle information storage and retrieval is simple. Suppose there is a store of documents and a person (user of the store) formulates a question (request or query) to which the answer is a set of documents satisfying the information need expressed by his question. He can obtain the set by reading all the documents in the store, retaining the relevant documents and discarding all the others. In a sense, this constitutes 'perfect' retrieval. This solution is obviously impracticable. A user either does not have the time or does not wish to spend the time reading the entire document collection, apart from the fact that it may be physically impossible for him to do so.

The problem of finding relevant information is not new. However the advent of World Wide Web has changed the focus and increased the importance of information retrieval. Instead of going to local library to find something, people search the web. Now a user enters a query that describes a request for information, and expects that an information retrieval system responds by identifying documents that are relevant to query. The relative number of manual versus computer assisted searches for information has shifted dramatically in past few years. This has increased the need for automated information retrieval for extremely large document collections. In fact a search engine is also a form of an information retrieval system. However in spite of efficient search technologies being developed, the basic problem of getting relevant documents and that too in the top ranked documents is still unsolved. In the coming



subsection we discuss the problems associated in effective information retrieval, need of research in this field and tools that can be used for increasing efficiency of information retrieval.

IR can be defined, in general, as the problem of the selection of “relevant” documentary information from storage in response to search questions provided by a user. Yet, often when IR systems are evaluated, they are found to miss numerous relevant documents. Moreover users have become complacent in their expectation of accuracy of IR system. To increase the retrieval efficiency early systems tried to classify knowledge into a set of known fixed categories (Yahoo open directory). The problem with this approach is that categories commonly do not place documents into the categories where searches expected to find them. No matter what categories a user thinks of- they will not match what someone who is searching will find. Another approach is to try to understand the content of documents. Ideally, documents would be loaded into computer, the computer would read and understand them, users would simply ask questions and be given direct answers. The field of Natural Languages Processing works on this approach- but suffice to say that this approach is extremely difficult, and we currently lack systems that can build good knowledge structures for even simplest of texts [18].

If we rule out hand categorization and language processing that leaves us with Information Retrieval. The key problem is that simply matching on query words is not sufficient to satisfy user requests. With this approach, no real attempt is made to have the computer understand the documents- instead techniques that use pattern matching, statistical sampling, machine learning, soft computing and probability theory are used to guess which documents are relevant. These systems are not perfect, but they offer users the opportunity to sift through large volumes of text.

## **1.2 Information Retrieval System**

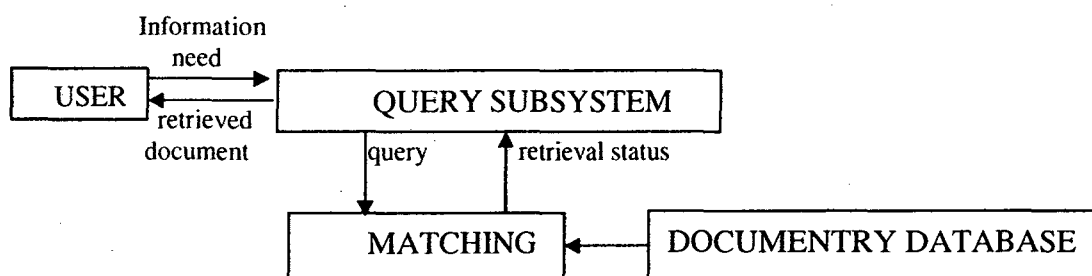
Information retrieval is devoted to finding relevant documents from storage in response of a query provided by a user [18]. Information Retrieval System (IRS) retrieves information in response to user queries. It stores a collection of information (generally called documents), accepts a user query, searches the document collection and returns ranked set of relevant documents. An IRS consists of three basic components: documentary database, query subsystem and matching mechanism [18].

**The documentary database:** This component stores documents, along with the representation of their information contents. It is associated with the indexer module, which automatically generates a representation of each document by extracting the document contents. Textual document representation is typically based on index terms (single terms or phrases), which are content identifier of documents.

**The Query Subsystem:** It allows user to specify their information needs and presents the relevant documents retrieved by the system to them. To do this it requires a query language that allows user to formulate query and a procedure to select the relevant documents.

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

Most of the IRS's use keywords to retrieve documents. The system first extracts keywords from documents and then assigns weights to the keywords by using different approaches. Efficiency of such systems is based on how efficiently they solve two major problems: one is how to extract keywords precisely and other is how to decide weight of each keyword.



**Fig 1.1:** Graphical representation of an information retrieval system

To measure effectiveness of IRS, Precision and recall are used. Precision is the ratio of the number of relevant documents retrieved to the total number retrieved.

$$\text{precision} = \frac{\text{no. of relevant documents retrieved}}{\text{no. of documents retrieved}} \quad (1.1)$$

Precision provides an indication of the quality of the retrieved set. However, this does not consider the total number of relevant documents.

Recall considers the total number of relevant documents. It is the ratio of relevant documents retrieved to the total number of documents in the collection that are believed to be relevant.

$$\text{recall} = \frac{\text{no. of relevant documents retrieved}}{\text{total no. of relevant documents}} \quad (1.2)$$

## 1.3 Information Retrieval Models

For building an IRS the documents and queries have to be represented in some model. Based on the model, a retrieval strategy is defined that assigns a measure of similarity between a query and the document. A ranked list of relevant documents is then returned based on the retrieval strategy. Several retrieval models have been studied and developed in IR areas [18]. Some of the popularly used models are:

### **Boolean Model**

In a Boolean Model a document is represented on the basis of index terms using a binary indexing technique. That is index term is either 1 or 0 depending on whether it is present in the document or not. User queries can also be represented in a similar way. User query may also allow combination of query terms with logical operators AND, OR and NOT. In this case the document is either relevant or not relevant to a query. Therefore conventional Boolean model does not lend itself well to relevance ranking. However there are extended Boolean models that allow for relevance ranking of the retrieved relevant documents.

### **Probabilistic Model**

The Probabilistic model computes the similarity coefficient between a query and a document as the probability that the document will be relevant to the query. There are two fundamentally approach- the first relies on usage patterns to predict relevance [21] and the second uses each term in the query as clues as to whether or not a document is relevant [21].

### **Vector Space Model (VSM)**

VSM is based on phenomenon of vector space. Each document in VSM is represented as a vector of terms in n- dimensional vector space, where n is the number of distinct term in document collection after preprocessing.

A document is represented as

$$D = (w_1, w_2, \dots, w_n) \quad (1.3)$$

Where  $w_i$ 's are term weights in document D.

In order to use VSM, the document is first preprocessed, indexed based on term weights and then appropriate similarity measure is then used to matching query and document.

### **Preprocessing**

Preprocessing of a document deals with removing words or symbols that do not convey meaningful information. Most of the Natural languages have some words like prepositions, connectives, punctuations prefix and suffix of a word that can be preprocessed [18].

Document preprocessing can be divided mainly into five text operations (or transformations):

- (1) Lexical analysis of the text with the objective of treating digits, hyphens, punctuation marks and the case of letters, and identifying tokens.
- (2) Elimination of stop-words- common non-relevant words like preposition and connectives
- (3) Stemming of the remaining words with the objective of removing affixes (i.e., prefixes and suffixes) and allowing the retrieval of documents containing syntactic variations of query terms (e.g., connect, connecting, connected, etc).
- (4) Selection of index terms to determine which words/stems (or groups of words) will be used as indexing elements. In fact, noun words frequently carry more semantics than adjectives, adverbs, and verbs.
- (5) Construction of term categorization structures such as a thesaurus, or extraction of structure directly represented in the text, for allowing the expansion of the original query with related terms (a usually useful procedure).

### **Indexing**

There are many techniques to compute term weight [54]. Some are them based on term frequency. Out of them (tf-idf) technique is most prevalent. To understand this

(tf-idf) technique to weight each preprocessed term of document, consider the following definitions.

$n$  = number of distinct terms.

$tf_{ij}$  = number of occurrences of term  $t_j$  in document  $D_i$  [term frequency].

$df_j$  = number of documents that contain term  $t_j$ .

$$idf_j = \log\left(\frac{d}{df_j}\right)$$

where  $d$  is total number of documents (inverse document frequency).

The weight  $w_{ij}$  for each term  $t_j$  in document  $D_i$  is defined as

$$w_{ij} = tf_{ij} * idf_j. \quad (1.4)$$

The query is also represented as document is.

### Similarity Measures

Once documents and query are represented as vector of terms, a vector based similarity measure can be used for matching query and documents. The similarity measure that is accurate and provides good retrieval performance is obviously used more by the IR system. There are many similarity measures [23], some of popularly used similarity measures are given below.

**Dot product measure:** dot product of two vectors  $Q$  and  $D_i$  is defined as [23]

$$SC(Q, D_i) = \sum_{j=1}^n w_{Qj} w_{ij} \quad (1.5)$$

It measures how many terms are matched but it has a measure flaw. A longer measure document (not necessarily relevant to given query) may result in a higher score simply because it is longer and thus, has a higher chance of containing terms that match the query.

**Cosine measure:** Cosine measure gives the cosine of the angle between the query and document vector. It can be seen as a dot product divided by product of length of the documents. It is the most commonly used similarity measure [23].

$$SC(Q, D_i) = \frac{\sum_{j=1}^n w_{Qj} w_{ij}}{\sqrt{\sum_{j=1}^n (w_{ij})^2 \sum_{j=1}^n (w_{Qj})^2}} \quad (1.6)$$

Cosine measure does not depend on the length of the document and hence does not rank to those longer documents that are not relevant.

**Dice coefficient:** dice coefficient is defined as twice the number of terms common to compared entities divided by the total number of terms in both tested entities.

$$SC(Q, D_i) = \frac{2 \sum_{j=1}^n w_{Qj} w_{ij}}{\sum_{j=1}^n (w_{ij})^2 + \sum_{j=1}^n (w_{Qj})^2} \quad (1.7)$$

The coefficient result of 1 indicates identical vector where as a 0 equals orthogonal vectors

**Jaccard measure:**

$$SC(Q, D_i) = \frac{2 \sum_{j=1}^n w_{Qj} w_{ij}}{\sum_{j=1}^n (w_{ij})^2 + \sum_{j=1}^n (w_{Qj})^2 - \sum_{j=1}^n w_{Qj} w_{ij}} \quad (1.8)$$

It measures the degree of overlap between two sets. The Jaccard similarity penalizes a small number of shared entities more than the coefficient.

Where  $SC(Q, D_i)$  denotes the similarity coefficient that is used to calculate the score of document  $D_i$  for given query  $Q$ .  $w_{Qj}$  denotes the weight for  $j$ th term of query  $Q$  and  $w_{ij}$  is the weight of  $j$ th term of document  $D_i$ .

## 1.4 Information Retrieval Tasks

IR models provide a method to represent documents and query. Further similarity measures are used to compare query and document. However, designing better model and better similarity measure are only two aspect of improving efficiency of IR systems. There are many other tasks that can help in improving efficiency of IR systems. Following are some important subtasks of IRS aimed towards this objective

- Document Ranking
- Automatic document indexing

- Document classification
- Document clustering
- Query learning
- Query expansion

These parts are becoming essential parts of good search engines.

## **1.5 Soft Computing**

Soft computing is an expression used to indicate a synergy of methodologies useful for solving problems requiring some form of intelligence that diverts from traditional computing. Soft Computing is a collection of techniques which use the human mind as a model and aims at formalizing our cognitive processes. These methods are meant to operate in an environment that is subject to uncertainty, imprecision partial truth and approximation [57]. The objective is to study, model and analyze complex phenomena for which more conventional methods have not yielded low cost, analytic, and complete solutions. According to Zadeh [63] the guiding principle of soft computing is to exploit the tolerance for imprecision and uncertainty to achieve tractability, robustness, and low solution cost. Soft computing replaces the traditional time-consuming and complex techniques of hard computing with more intelligent processing techniques.

The key aspect for moving from hard to soft computing is the observation that the computational effort required by conventional approaches which makes in many cases the problem almost infeasible, is a cost paid to gain a precision that in many applications is not really needed or, at least, can be relaxed without a significant effect on the solution. A basic difference between perceptions and measurements is that, in general, measurements are crisp whereas perceptions are fuzzy.

Soft computing differs from conventional (hard) computing in that, unlike hard computing, it is tolerant to imprecision, vagueness, partial truth and approximation. The principal constituents of soft computing are: fuzzy logic, genetic algorithm, neural network, evolutionary computing, chaotic computing, rough sets and parts of machine learning theory. Real world problems are pervasively imprecise and uncertain and hard computing is based on precision and certainty. So soft computing plays a role in solving problems of uncertainty and vagueness.

## 1.6 Application of Soft Computing in IR

Different components of soft computing are being used in information retrieval. These techniques have changed the criteria of hard computing used for problems of information retrieval. Since the process of retrieving relevant information are full of uncertainty and imprecision. Soft computing is drastically used in every field attached with IR. The constituents of soft computing share common features and they are considered complementary instead of competitive [11, 17].

Fuzzy set theory provides algorithms for dealing with imprecision and uncertainty [33]. It is used in many areas of information retrieval. Due to the large corpus, systems are not capable to manage imperfect documents and retrieve relevant information. Due to imperfection in documents use of fuzzy set theory is prevalent in IRS's. Document representation is quite tuff due to the subjectivity of documents, but fuzzy set theory makes easier the task of getting representation of document. Ranking function for finding relevant documents against a query can be found using fuzzy set.

Artificial neural network is the machinery for learning and adaption [26]. It is used to modify stored information in response to new inputs from the user. Neural network provides capability to associatively recall information despite noise or missing pieces in the input. Information can be categorized by neural networks by their associative patterns. If there is no exact data against the query, neural network can be used to retrieve "nearest neighbor" data.

The genetic algorithm is an optimizing technique which is based on principle of evolution and heredity [20, 21,27]. It is targeted to the learning of description of documents in the documentary database. GA has been applied for term and document clustering [18]. Effective matching function for retrieving relevant documents from corpus can be achieved using GA.

The mentioned technologies can be combined in models which exploit their best characteristics. As an important consequence, some subjective problems can be solved most effectively by using hybrid systems what is increasing the interest on them. The first and probably the most successful hybrid approach till now are the so-called



neurofuzzy systems [49], although some other hybridations are being developed with great success as, for instance, the genetic fuzzy systems [16]. In the partnership of the mentioned collection of computational techniques that compose soft computing, and representation of aspects that are only qualitatively known; Probabilistic Reasoning such as Bayesian Belief Networks, with uncertainty and belief propagation; the main characteristic is its ability to update previous outcome estimates by conditioning them with newly available evidence [48]; last but not least, Genetic provide approaches to computing based on analogues of natural selection, such as, the optimization methods based on particle swarms [32].

## Application of Fuzzy Set Theory in IR

---

### 2.1 Introduction to Fuzzy Sets

Real world is full of uncertainty. Uncertainty could arise due to generality, vagueness, ambiguity, chance or incomplete knowledge. Although probability theory has been an effective tool to handle uncertainty, it can be applied only to situations whose characteristics are based on random processes. However, there turn out to be problems, a large class of them whose uncertainty is characterized by a non-random process. Here the uncertainty can be due to partial information which is not fully reliable or due to inherent imprecision in the language. In such situations, fuzzy set theory exhibits immense potential for effective solving of the uncertainty [33,51].

#### Definition

Let  $X$  be the universe of discourse,  $X = \{x_1, x_2, \dots, x_n\}$ . And let  $A$  be a fuzzy set in  $X$ , then the fuzzy set  $A$  can be represented as:

$$A = \{ (x_1, \mu_A(x_1)), (x_2, \mu_A(x_2)), \dots, (x_n, \mu_A(x_n)) \} \quad (2.1)$$

Where  $\mu_A, \mu_A : X \rightarrow [0,1]$  is the membership function of the fuzzy set  $A$ .  $\mu_A(x_i)$  indicates the degree of membership of  $x_i$  in  $A$  [33,51].

Eqn (2.1) can also be represented as

$$A = \sum_{i=1}^n \frac{\mu_A(x_i)}{x_i} = \frac{\mu_A(x_1)}{x_1} + \frac{\mu_A(x_2)}{x_2} \dots + \frac{\mu_A(x_n)}{x_n} \quad (2.2)$$

The membership function need not always be discrete function, it can be continuous also. The fuzzy membership function for the fuzzy linguistic term “moderate” relating to temperature may turn out to be as illustrated in fig. 2.1.

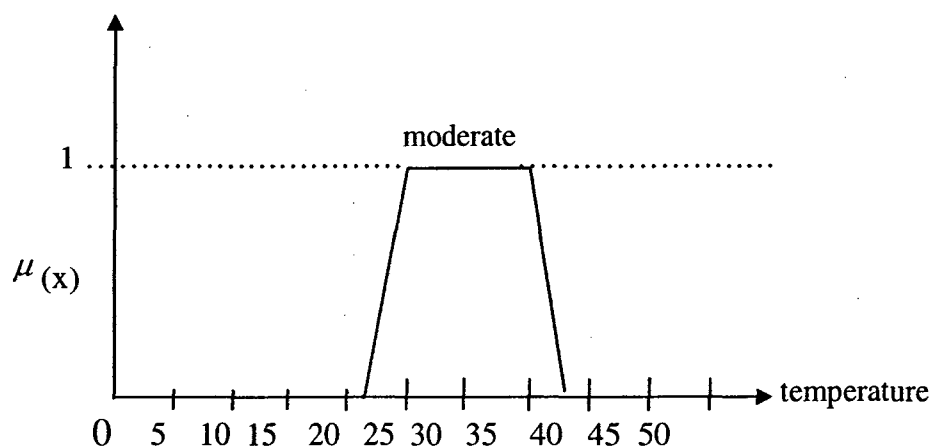


Fig 2.1: continuous membership function for “moderate”

## 2.2 Application of Fuzzy Set Theory in IR

Given a document collection, information retrieval systems are designed to provide a set of document that are likely to contain the information desired by the user. With the advent of the internet the amount of information has increased to a point that there are great demands for effective systems that allow easy and flexible access to information [8]. Here flexibility means that capability of the system to manage imperfect information expressed by imprecise uncertain vague and inconsistent information. Uncertainty refers to the truth of the information stated in a proposition, relating to the credibility of the information content. Uncertain information is generally expressed by terms such as likely, probable possible. A well-established formal framework for dealing with uncertainty is fuzzy theory. Imprecision is related to the content of the information stated in a propositional form with respect to the granularity of the values of a variable. Imprecise information is characterized by subset values of reference domain. For example, the statement “Mona is young” contains vagueness when the reference domain is the interval  $[0,100]$ , while it is precise when the reference domain is the set of levels {“very young”, ‘young’, ‘old’,}. Fuzzy set theory can handle vagueness and imprecision [33, 37 ].

The imperfect information comes from three major aspects in an information retrieval system environment including the representation of user queries, the representation of document and the similarity measures between user’s queries and documents.

A representation of what a document is difficult to build. TF-IDF technique is not sufficient to represent a document [30]. In addition of terms of documents, term

relationships play an important role in document representation. Term relationship includes synonyms, hypernyms, broader terms etc. These term relationships are also imperfect. Also user may not use the specific terms as given by experts [31]. So, in order to get rid of imperfect information, a document can be represented as a fuzzy set of terms where weights to the terms and term relationships are given by membership values [1,31]. Moreover concept based models are coming into existence for better representation of documents concepts describe the topical content and structure of documents.

When expressing needs for information, users often formulate queries in a Natural Language (NL). NL is vague and imprecise [36]. In order to respond properly a retrieval system should interpret query meaning and then expand the query. Fuzzy set theory is applied to improve the vagueness of query and expand the query.

In addition to the problem of characterization of documents and queries, a crucial point is to establish a similarity measure between the user request and documents in the collection to determine whether a document is relevant or not. A document should be considered relevant when the document meets the actual user information need. Only the user can determine the true relevance of a document. As user's need are imprecise and uncertain, fuzzy sets can play an important role in formalizing user needs.

Documents can also be clustered by Fuzzy set theory. Document clustering is putting similar objects together and dissimilar separately. With Fuzzy clustering techniques, a fuzzy partition of the document space is created in which each Fuzzy cluster is defined by a Fuzzy set of documents [43, 25].

T. Radecki extended Boolean model in terms of representation of documents as fuzzy set of terms that is more accurate than the traditional representation [50]. Bordogna and Pasi have proposed a fuzzy representation of structured documents [6,17]. For HTML documents, Molinari and Pasi have proposed an approach to index documents [45]. Radecki, Bookstein have interpreted query weights as indicators of the relative importance among terms in a query [5,50]. Bordogna, Carrara and Pasi [9] have interpreted query weights as specifications of ideal significance degrees. Kraft, Bordogna, and Pasi [36] have proposed a closeness measure that allows for asymmetry. Bordogna and Pasi [7] have defined a fuzzy retrieval model in which the linguistic descriptors are formalised within the framework of fuzzy set theory through

linguistic variables. Miyamoto and Nakayama [43] have introduced the concept of fuzzy pseudo-thesauri and fuzzy associations based on a citation index. Kohout et al. [35] proposed fuzzy relational products for thesaurus construction. Bezdek, Biswas, and Huang [4] generate a thesaurus based on the max-star transitive closure for linguistic completion of a thesaurus generated initially by an expert linking terms. Miyamoto has introduced the definition of a fuzzy thesaurus [44], as a fuzzy relation. Miyamoto [43], and De Mantaras et al. [41] have worked on fuzzy clustering for retrieval.

## 2.3 Fuzzy Similarity Measures

We often encounter situations where we have to find out the similarity between two fuzzy sets. For this we require fuzzy similarity measure. As we have discussed, query and documents can be represented as fuzzy sets. Some of the fuzzy similarity measures to documents and query are discussed below [31,60].

Let  $D$  and  $Q$  be two fuzzy sets on a universe of discourse  $X$ ,  $S(D,Q)$  denotes similarity between  $D$  and  $Q$ , and  $\mu_D(x)$  denotes the membership of  $x$  in  $D$ . Then

$$1. \quad S(D, Q) = \frac{\left\{ \sum_{i=1}^n \left[ \frac{m(\mu_D(x_i), \mu_Q(x_i))}{M(\mu_D(x_i), \mu_Q(x_i))} \right] \right\}}{n} \quad (2.3)$$

Where  $m(a,b)$  and  $M(a,b)$  denote the minimum and maximum values of  $a$  and  $b$  respectively. In order to avoid the denominator being zero, we set  $0/0 = 1$ .

$$2. \quad S(D, Q) = \frac{\sum_{i=1}^n [1 - |\mu_D(x_i) - \mu_Q(x_i)|]}{n} \quad (2.4)$$

$$3. \quad S(D, Q) = \max_{x \in X} \min(\mu_D(x), \mu_Q(x)) \quad (2.5)$$

Most of these similarity measures have no mechanism to reflect the user preference in retrieving relevant. Thus a new similarity measure as discussed below has been proposed to incorporate the user preference or intention.

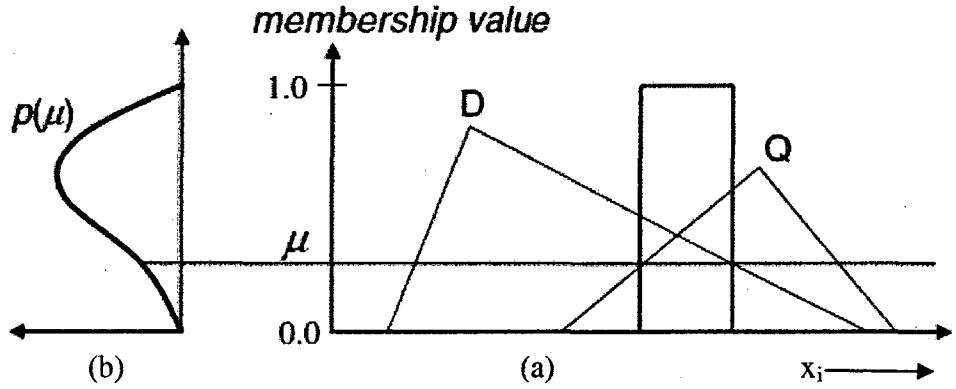
4. Before computing similarity between document and query, a function, called overlap function [31] is defined as

$$f(\mu : D, Q) = \sum_{i=1}^n \delta(x_i, \mu : D, Q) \quad (2.6)$$

Where

$$\delta(x_i, \mu : D, Q) = \begin{cases} 1.0 & \text{if } \mu_D(x_i) \geq \mu \text{ and } \mu_Q(x_i) \geq \mu \\ 0.0 & \text{otherwise} \end{cases} \quad (2.7)$$

$\delta(x_i, \mu : D, Q)$  determines whether two sets are overlapped at the membership degree  $\mu$  for index term  $x_i$ . It returns an overlap value of 1.0 when the membership degrees of the two sets are both greater than  $\mu$ ; otherwise, it returns 0.0.  $f(\mu : D, Q)$  be the sum for all terms  $x_i$ .



**Figure 2.2:** Preference-based similarity computation: (a) overlap degree at between a document D and query Q; (b) a membership preference function [31].

Figure 2.2 (a) shows an overlap value  $f(\mu)$  at membership degree  $\mu$  between two fuzzy sets. The index terms  $x_i$ , satisfying both  $\mu_D(x_i) \geq \mu$  and  $\mu_Q(x_i) \geq \mu$ , are given a value 1.0 by Eqn. 2.7.

Now the similarity measure between D and Q is defined as

$$SC(D, Q) = \sum_{\mu} f(\mu : D, Q) p(\mu) \quad (2.8)$$

Where  $p(\mu)$  is a preference function, which has fuzzy membership and it is determined by user. It is possible when two documents that have different fuzzy sets get the same degree of similarity. In this case the preference function can recognize their relevancy by focusing on the higher range of membership degrees. Users searching the Web for information tend to focus on the document with the terms having higher matching values. Thus the relevance of the highest-matched document plays an

important role in user satisfaction. In such cases,  $p(\mu)$  is given a value in the range  $[0.8,1.0]$  when  $\mu_D(x_i)$  is considered significant, i.e.,  $\mu_D(x_i) \geq 0.8$ . Under this case, the index terms with higher weights are given greater preferences in the calculation of the similarity between D and Q. Conversely,  $p(\mu)$  is given a value in the range  $[0.0,0.4]$  when  $\mu_D(x_i)$  is considered insignificant, i.e.,  $\mu_D(x_i) \leq 0.4$ . Given the preference function  $p(\mu)$  in Fig. 2.2(b), the similarity  $S(D,Q)$  is obtained by summing the product of  $f(\mu : D,Q)$  and  $p(\mu)$  for all membership degrees [31].

## 2.4 Concept Based IR Using Fuzzy Set Theory

### 2.4.1 Limitation of VSM in IR

As we have discussed in chapter 1, in VSM - a document is represented by the terms present in the document and similarity between query and document depends on terms, which are common to both query and document. Instead of using all the terms, some representative terms called keywords may be used for representing a document. Considering subjectivity in NLP (Natural language processing), it is well known fact that keywords (important terms present in the document) are not sufficient to provide information about content. Therefore, **VSM does not take into account the content of a document. The content of a document can be extracted by finding semantically important terms [30, 31].**

This can be explained by an example. Let us consider a document  $D_1$  with following content.

$D_1 =$  "Now automobile companies are focusing on manufacturing of cars because four wheelers are booming in market. The reason behind flourishing of cars is safety"

Reading this document  $D_1$ , one can make out that automobile, car and four-wheeler are more important than other terms such as market, companies and safety. The TF weight of the term 'automobile' is 1, which is the same as that of terms 'market' and 'manufacturing'. Therefore, in VSM 'market' and 'manufacturing' will have equal role in representation of document.

## 2.4.2 Methods of Finding Semantically Related Terms

Various attempts have been made to enhance the indexing performance by exploiting linguistic phenomena [30, 55, 56]. One such linguistic phenomenon is the lexical chain; two words are in a lexical chain if they are related by a relation. There are many relations among words such as - identity, synonymy, hypernymy and hyponymy. These relations link related terms in a document to represent the lexical cohesion structure of the document. Thus presence of a lexical chain identifies semantically important terms. If we again consider the earlier example (document  $D_1$ ) we observe that 'automobile', 'car' and 'four-wheelers' form a lexical chain. Therefore, 'automobile', 'car', and 'four-wheelers' can be identified as semantically important terms. These semantic aspects can be captured using some ontology. WorldNet is one of examples of ontology [28].

A concept based model can be used to capture content based similarity between document and query. In this model a new indexing technique is used that regards a document as a cluster of concepts. These concepts are extracted by lexical chains for index terms.

## 2.4.3 Concept Based Model for Information Retrieval Based on Fuzzy Logic

IR using concept based model can be done using following two steps: (1) Form an index based on concept identification and (2) Use the index find similarity between documents and the given query.

A concept based model can be used to capture content based similarity between document and query. In this model a new indexing technique is used that regards a document as a cluster of concepts. These concepts are extracted by lexical chains for index terms.

To construct index of a document, concept clusters are used. Concept cluster is a weighted lexical chain in which one term is related to another term with some relationship having membership degree. This can be explained as- the degree of membership (relationship) from term  $t_1$  to term  $t_2$  is given by a **fuzzy function** (relationship function)  $f: t_1 * t_2 \longrightarrow [0, 1]$ . There can be many relationships between two terms as given above.



**Definition 1.** Let  $T = \{ t_1, t_2, \dots, t_k \}$  be set of nouns in a document, and  $R = \{ r_1 = \text{identity}, r_2 = \text{synonym}, r_3 = \text{hypernym}, r_4 = \text{hyponym} \}$  be set of lexical relations. Let  $C = \{ C_1, C_2, \dots, C_m \}$  be the set of concept clusters in a document. A concept  $C_j \in C$  is composed of a set of  $t_g \in T$  and  $r_k \in R$ .

Now again consider the example given in section 2.4.1. Fig.2.3 shows the relationship between nouns (shown as nodes) extracted from document  $D_1$  and lines (shown as edges of graphs) denoting relations between terms:

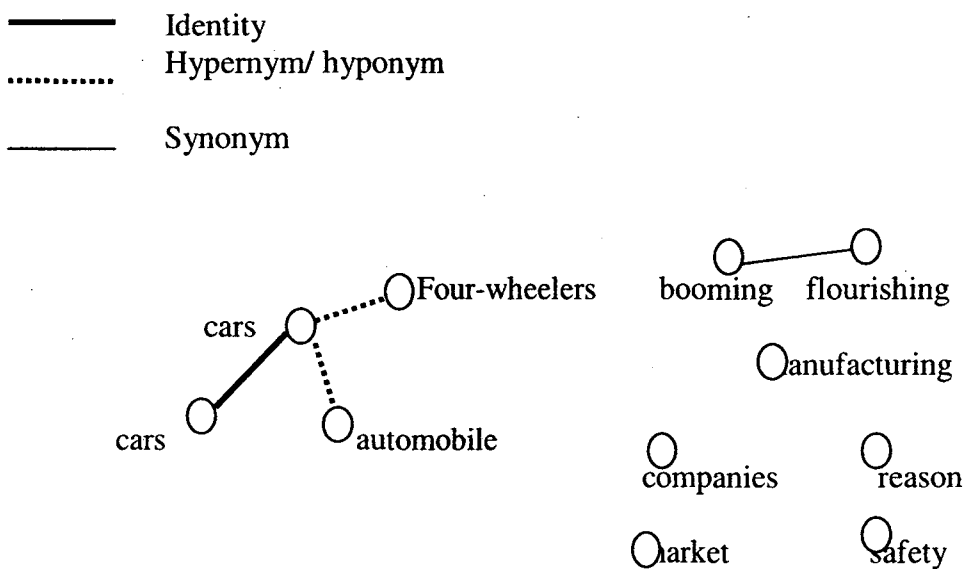


Fig. 2.3 Concept clusters from the text “ $D_1$ ”.

Cluster 1 cars, automobile, four-wheelers.

Cluster 2 companies

Cluster 3 market

Cluster 4 manufacturing

Cluster 5 booming, flourishing

Cluster 6 reason

Cluster 7 safety

**Definition 2.** Let  $T = \{ t_1, t_2, \dots, t_m \}$  be the set of terms in a concept cluster. Let  $R = \{ \text{identity}, \text{synonyms}, \text{hypernym}, \text{hyponym} \}$  be the set of lexical relations. Let  $N(r_k, t_g)$  be the number of relations  $r_k \in R$  that term  $t_g \in T$  has with the other terms, and let

$W(r_k)$  be the weight of relation  $r_k$ . Then the score  $S_{Term}(t_g)$  of term  $t_g$  in a concept cluster is defined as

$$S_{Term}(t_g) = \sum_{k=1}^4 N(r_k, t_g) \times W(r_k) \quad 1 \leq g \leq m. \quad (2.9)$$

$S_{Term}(t_g)$  is determined by the relations that  $t_g$  has with the other terms and their weights. A large value of  $S_{Term}(t_g)$  indicates that  $t_g$  is a semantically important term in a document.

**Definition 3.** Let  $C = \{C_1, C_2, \dots, C_n\}$  be the set of concept clusters for a document. Let  $T = \{t_{i1}, t_{i2}, \dots, t_{ik}\}$  be the set of terms in concept cluster  $C_j \in C$ . Then the index terms and their weights for the document are defined as

$$I = \{(t_{ij}, S_{Term}(t_{ij})) \mid t_{ij} \in C_i, 1 \leq i \leq n, 1 \leq j \leq k\} \quad (2.10)$$

Now, in order to calculate the similarity between document and a query a vector based similarity measure such as cosine measure can be used.

## 2.5 Algorithm and Experiment for Concept Based IR

Based on definitions given above, we have developed an algorithm and performed an experiment to implement the algorithm.

### 2.5.1 Algorithm for Identifying Concept Clusters

Before giving algorithm for identifying concept clusters we incorporate the concept of graph in our algorithm to alleviate it. A concept cluster is a connected component of a graph, nodes represent terms, and labeled edges represent relation between two terms (refer to Fig.2.3). And the graph having these connected components represents the documents.

Now we present simplified algorithm for concept based IR.

1. Initialize weights of semantic relations.
2. For each document  $D_i$  find  $T_i$ , where  $T_i$  represents the set of all nouns in document  $D_i$ . (use POS tagger)
3. Identify all possible concept clusters

- 3.1 Start with a single node term  $x$ .
- 3.2 Find all nodes  $y_i$  related to  $x$  and use them to extend the graph by putting proper edges.
- 3.3 Go on extending the graph for all the nodes obtained until all possibilities of expansion are exhausted.
4. Find weight of each term using concept cluster
  - 4.1 For each term  $x$  find its concept cluster  $k$ .
  - 4.2 repeat following steps within concept cluster  $k$ .
    - 4.2.1 For node  $x$ , identify the relation using the edges connected to  $x$ . and calculate weight of  $x$  using formula given in eqn. (2.9).
5. Represent document by these remaining terms.

## 2.5.2 Experiment

We have done a simple experiment to explain our algorithm and efficiency of our model in comparison of VSM. The data set used in our experiment is shown in appendix 1. Assigning similarity manually, we can say that for query “cricket”, first four ranked relevant documents are: doc. 3, doc. 4, doc. 2, doc. 1 and for query “yoga”, first four ranked relevant documents are: doc 5, doc 7, doc 8, doc 6.

We are explaining working of our algorithm for doc 4. Let relation weights  $W(r_k)$  are : (identity, 1), (synonym, 0.7), (hypernym, 0.5), (hyponym, 0.5) . We have used POS tagger(cognitive computation group) to extract nouns. The set  $T_4$  of extracted nouns from doc. 4 are: {cricket, sport, India, development, country, caste, religion, nationality, nation, indubitably}. Now consider a node term ‘cricket’ and then find those node terms, which are related to ‘cricket’ having labeled edges as relations with ‘cricket’. These related terms are: {(cricket, identity), (sport, hypernym)}. This connected graph cannot be extended further.

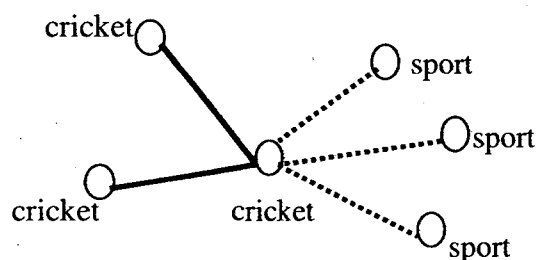


Fig. 2.4: Concept cluster of ‘term cricket’ in doc. 4 in appendix 1.

In the same way all concept cluster can be identified. After constructing connected components of graph we calculate weight of each term of  $T_4$  by eqn. (2.9). Here weight of term 'cricket' is equal to 3.5. Finally, we present document with the remaining term of  $T_i$ . The same procedure has been done for all documents.

Table 1 and table 2 show the weights assigned to the terms using standard TF-IDF and fuzzy weighting technique.

(We have shown weights only for those terms present in queries.)

Doc/term	Cricket	yoga	Doc/term	cricket	Yoga
doc. 1	1.510	0	doc. 1	3	0
doc. 2	0.9060	0	doc. 2	2	0
doc. 3	0.9060	0	doc. 3	4	0
doc. 4	0.6040	0	doc. 4	3.5	0
doc. 5	0	0.6040	doc. 5	0	4.2
doc. 6	0	0.6040	doc. 6	0	1
doc. 7	0	0.6040	doc. 7	0	2
Doc. 8	0	0.9060	Doc. 8	0	1.5

**Table1: weights by TF-IDF technique    Table 2: weights by concept based technique**

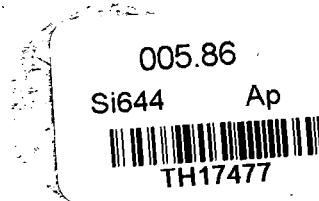
Table 3 shows the result of ranked relevance documents retrieved by VSM and concept based technique.

techniques queries	VSM	Concept based	Expected result
Cricket	Doc 1	Doc 3	Doc 3
	Doc 2	Doc 4	Doc 4
	Doc 3	Doc 2	Doc 2
	Doc 4	Doc 1	Doc 1
Yoga	Doc 8	Doc 5	Doc 5
	Doc 5	Doc 7	Doc 7
	Doc 6	Doc 8	Doc 8
	Doc 7	Doc 6	Doc 6

**Table 3: relevant documents retrieved by VSM, Concept based and expected result.**

The result shows that returned by fuzzy model is more accurate in terms of precision and recall. This is quite obvious, as fuzzy model is able to semantic similarity between the terms.

005.86 Si644 AP TH-17477



TH

# Application of Neural Network in IR

---

## 3.1 Introduction to Neural Network

Neural network (NN) is a simplified model of the biological neuron system. It is a massively parallel distributed processing system consisting of highly interconnected neural computing elements which are intended to interact with the object of the real world in the same way as biological systems do [51].

### 3.1.1 Definition of Neural Network

Neural networks are one of the important components in artificial intelligence. Neural network is also called connectionist model, neural net, collective model, parallel distributed processing model, and artificial neural network. Various definitions of neural network are given by researchers. Two of them are given below.

A definition of neural network given by Kevin Gurney [22] is as follows:

A Neural Network is an interconnected assembly of simple processing elements, units or nodes, whose functionality is loosely based on the animal neuron. The processing ability of the network is stored in the inter-unit connection strengths, or weights, obtained by a process of adaptation to, or learning from, a set of training patterns.

Another definition given by Haykin [26] is as follows:

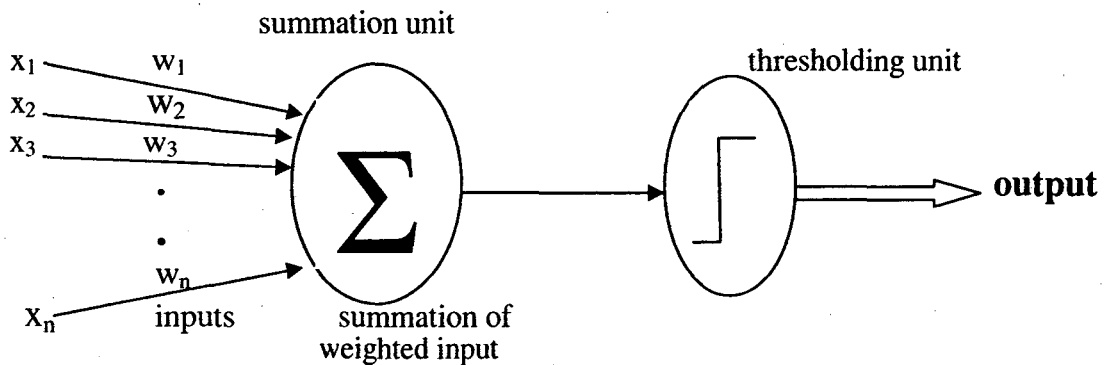
A neural network is a parallel distributed processor made up of simple processing units, which has a natural propensity for storing experiential knowledge and making it available for use. It resembles the brain in two respects:

- Knowledge is acquired by the network from its environment through a learning process;
- Interneuron connection strengths, known as synaptic weights, are used to store the acquired knowledge.

### 3.1.2 Mathematical Model of Neural Network

A biological neuron receives all inputs through the dendrites, sums them and produces an output if the sum is greater than a threshold value. The inputs are passed on to the cell body through the synapse which may accelerate or retard an arriving input [51]. Based on this theory a mathematical model is constructed as below:

A typical artificial neuron and the modeling of a multilayered neural network are illustrated in Fig. 3.1.



**Fig. 3.1:** simple model of an artificial neuron

Referring to Figure 3.1, Let  $x_1, x_2, x_3, \dots, x_n$  are the inputs to the artificial neuron.  $w_1, w_2, w_3, \dots, w_n$  are weights attached to the inputs links. Here acceleration or retardation of the inputs is modeled by the weights. An effective synapse which transmits a stronger signal will have a correspondingly larger weight while a weak synapse will have smaller weights. Thus weights here are multiplicative factors of the inputs to account for the strength of the synapse. Hence total input received by the artificial neuron is

$$\begin{aligned}
 I &= w_1x_1 + w_2x_2 + \dots + w_nx_n \\
 &= \sum_{i=1}^n w_i x_i
 \end{aligned}
 \tag{3.1}$$

The sum is then passed on to a non linear filter  $\phi$  called activation function or transfer function, which release the output  $y$ .

$$\text{i.e.} \quad y = \phi(I)
 \tag{3.2}$$

Out of many activation functions, the thresholding function is commonly used. In this function the sum is compared with a threshold value  $\theta$ . If the value of  $I$  is greater than  $\theta$ , then the output is 1 else it is 0.

$$y = \varphi \left( \sum_{i=1}^n w_i x_i - \theta \right) \quad (3.3)$$

where  $\varphi$  is called step function known as Heaviside function and is such that

$$\varphi(I) = \begin{cases} 1 & \text{if } I > 0 \\ 0 & \text{if } I \leq 0 \end{cases} \quad (3.4)$$

There are other activation functions like Signum function, Sigmoidal function etc.

## 3.2 Learning Process of Neural Network

A neural network should be configured such that the network would get knowledge by training inputs and make it available for user. Various methods have been defined to make able network to learn. One way is to adjust the weights under the supervision of a teacher. Another way is to train the neural network without supervision of a teacher. The learning methods in neural networks may be classified in to three basic types. These are supervised learning, unsupervised learning, and reinforcement learning.

### 3.2.1 Supervised Learning

In a supervised learning process, the network learns under the supervision of a teacher. For each training input teacher tells the desired output, the weights are adjusted after finding error between the desired output and given by teacher and network's computed output. Multi-layer feed forward networks are always trained in supervised manner with a highly popular algorithm known as the error back propagation algorithm [51].

### Back Propagation Algorithm

Back propagation is one of the popular algorithms of supervised learning. The network using back propagation algorithm learns by iteratively processing a dataset of training tuples, comparing the network's prediction with known desired value. The

desired value may be the known class value of the training tuple or a continuous value. For each training tuple, the weights are modified so as to minimize the mean squared error between network prediction and the actual value. These modifications are done in the 'backwards' direction. i.e. from the output layer, through each hidden layer down to the first layer. In general, the weights will converge and the learning process stops [51]. The algorithm is given below.

Step 1: Initialize each weights and biases in network.

Step 2: Find output  $O_j$ , for each node in input layer

$$O_j = I_j$$

where  $I_j$  is the input of node  $j$  in input layer.

Step 3: Find input  $I_j$  for each hidden or output layer node  $j$

$$I_j = \sum_i w_{ij} O_i + \Theta_j$$

Where  $w_{ij}$  is the weight of the connection from node  $i$  in previous layer to node  $j$  and  $\Theta_j$  is bias to node  $j$ .

Step 4: The output  $O_j$ , for each hidden or output layer node  $j$

$$O_j = \frac{1}{1 + e^{-I_j}}$$

Step 5: For each node  $j$  in the output layer, find error  $Err_j$

$$Err_j = O_j (1 - O_j) (T_j - O_j)$$

where  $T_j$  is known desired value of giving training tuple.

Step 6: Find error  $Err_j$ , for each node  $j$  in the hidden layers, from the last to first hidden layer.

$$Err_j = O_j (1 - O_j) \sum_k Err_k w_{jk}$$

where  $k$  is the next higher layer.

Step 7: Update weight  $w_{ij}$

$$\Delta w_{ij} = (l) Err_j O_j, \text{ where } \Delta w_{ij} \text{ is change in the weight } w_{ij}.$$

$$w_{ij} = w_{ij} + \Delta w_{ij}$$

where  $l$  is a learning rate, a constant having a value between 0.0 and 1.0.

Step 8: Update each bias  $\Theta_j$

$$\Delta \Theta_j = (l) Err_j$$

$$\Theta_j = \Theta_j + \Delta \Theta_j$$

Step 9: Continue this process until terminating condition (as  $w_{ij}$  is very small below some specified value) is satisfied.



It is important to make the initial weights 'small'. Choosing initial weights too large make the network untrainable .

### **3.2.2 Unsupervised Learning**

In unsupervised learning procedures, the neural network does not receive any teaching or learning feedback, but it is left to learn by itself. This procedure is also often referred to as "self-organization" because the process relies only upon local information and internal control to learn by capturing regularities in the stream of input patterns.

### **3.2.3 Reinforcement Learning**

In this learning, a teacher though available, does not present the expected answer but indicates if the computed output is correct or not. The information provided helps the network in its learning. A reward is given for a correct answer and a penalty for wrong answer. Reinforcement learning is also usually involved in exploring a new environment when some knowledge (or subjective feeling) about the right response to environmental inputs is available. The system receives an input from the environment and produces an output as response. Subsequently, it receives a reward or a penalty from the environment and learns from the environment.

## **3.3 Application of Neural Network Models in IR**

The application of connectionist models to Information Retrieval is not a recent phenomenon. A number of researches have been done about the application of neural network in IR and are still is in progress. Neural networks computing, in particular, seem to fit well with conventional retrieval models such as the vector space model [53] and the probabilistic model. Doszkocs et al. [19] provided an excellent overview of the use of connectionist models in information retrieval. These models include several related information processing approaches, such as artificial neural networks, spreading activation models, associative networks, and parallel distributed processing. In contrast to more conventional information processing models, connectionist models are "self-processing" in that no external program operates on the network: the network literally processes itself, with "intelligent behavior" emerging from the local

interactions that occur concurrently between the numerous network nodes through their synaptic connections. Belew proposed probably the earliest connectionist model adopted in IR [3]. In adaptive information retrieval (AIR), Belew developed a three-layer neural network of authors, Index terms, and documents. The system used relevance feedback from its users to change its representation of authors, index terms, and documents over time. The result was a representation of the consensual meaning of keywords and documents shared by some group of Users. Kohonen has shown that his self-organizing feature map "is able to represent rather complicated hierarchical relations of high-dimensional space in a two-dimensional display [34]. He had concluded that "the self-organized mappings might be used to visualize topologies and hierarchical structures of high-dimensional pattern spaces. Rose and Belew extended AIR to a hybrid connectionist and symbolic system called SCALIR which used analogical reasoning to find relevant documents for legal research [52]. Kwok also represented an IR system into a three-layer network of queries, index terms, and documents [38]. A modified Hebbian learning rule was used to reformulate probabilistic information retrieval. Wilkinson and Hingston [61] incorporated the vector space model in a neural network for document retrieval. Their network also consisted of three layers: queries, terms, and documents. They have shown that spreading activation through related terms can help improve retrieval performance. Lin, Soergel, & Marchionini adopted a Kohonen network for information retrieval [39]. Kohonen's feature map, which produced a two-dimensional grid representation for N- dimensional features, was applied to construct a self-organizing (unsupervised learning), visual representation of the semantic relationships between input documents. In MacLeod and Robertson used neural algorithm for document clustering [40]. The algorithm compared favorably with conventional hierarchical clustering algorithms.

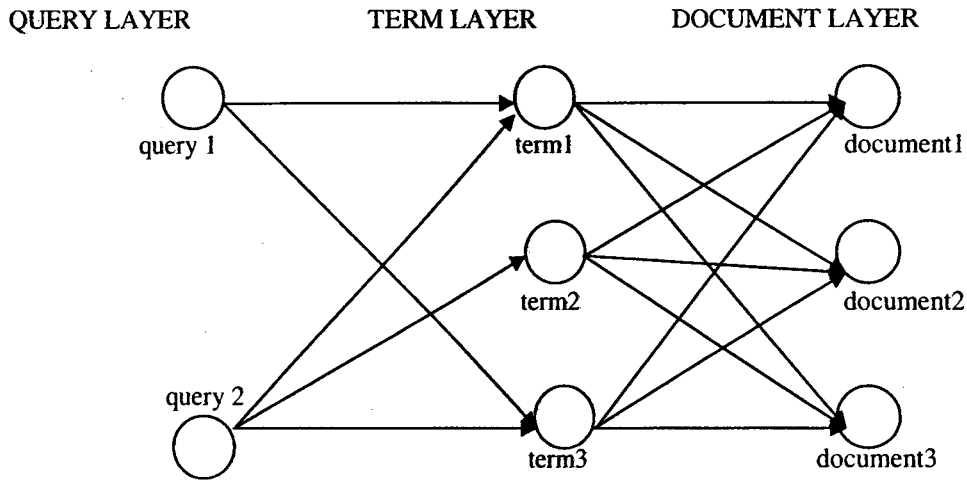
A variant of Hopfield network is developed by Chen and his colleagues [13] to create a network of related keywords. It uses an asymmetric similarity function to produce thesauri (or knowledge bases) for different domain-specific databases. These automatic thesauri are then integrated with some existing manually created thesauri for assisting concept exploration and query refinement. In addition, a variant of the Hopfield parallel relaxation procedure for network search and concept clustering is also implemented [12].

Hatano and his colleagues have proposed an information organizer for effective clustering and similarity-based retrieval of text and video data using Kohonen's self-organizing map [24]. In their model, a vector space model and DCT (Discrete Cosine Transform) image coding is used to extract characteristics of data and SOFM is used to cluster data. SOFM is also used by the DLI (digital library initiative) project at the University of Illinois. On one hand, it is used to classify and map the category of text documents. On the other hand, it is used to map the category of the texture of images. The Hopfield network has been adapted for concept analysis by chung [15]. The network is an asymmetric, continuous network in which the neurons are updated synchronously.

The performance of a branch-and-bound serial search algorithm was compared with that of the parallel Hopfield network activation in a hybrid neural-semantic network (one neural network and two semantic networks) [14]. Both methods achieved similar performance, but the Hopfield activation method appeared to activate concepts from different networks more evenly.

### **3.3.1 Vector Space Model with Neural Networks**

Neural network has been used to implement the vector space model [18]. The layered representation of NN using VSM model is shown in fig.3.2. The input layer represents QUERY layer and its nodes are set as queries. Hidden layer represents TERM layer and its nodes denote terms, and output layer represents DOCUMENT layer and its nodes denote documents. The links between the nodes are defined as query-term links and document-term links. A link between a query and a term indicates the term appears in the query. The weight of the link is computed by tf-idf technique for the term. Document-term links appear for each term that occurs in a given document. Again a tf-idf technique can be used for weighting the edges.



**Fig. 3.2:** neural network with document-term link

A feed-forward network works by activating a given node. A node is active when its output exceeds a given threshold. To begin, a query node is activated by setting its output value to one. All of its links are activated and subsequently new input weights for the TERM nodes are obtained. The input received by a term is

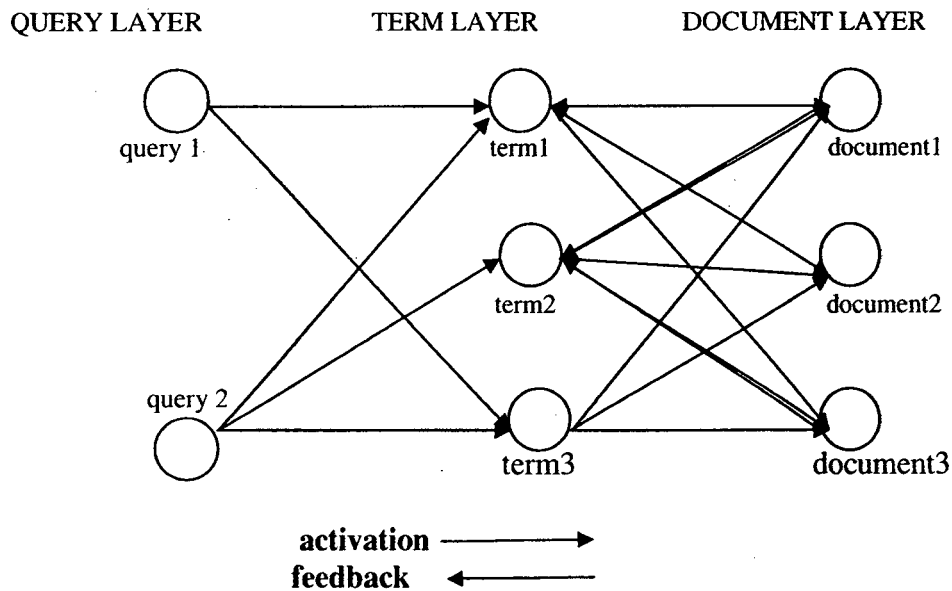
$$\sum_{i=1}^n w_i x_i$$

Where,  $w_i$  is weight computed by tf-idf for the link, and  $x_i$  is the output which is taken 1 for respective query node. Since there is only one node activated in QUERY layer. So total input received by a term in TERM layer is equal to (tf-idf)(1) which becomes input for the next phase. i.e.  $x = \text{tf-idf}(1)$ .

In the next phase, the TERM nodes and all of the links connected with  $n$  terms are activated. The DOCUMENT node receives the sum of all the weights associated with each term in the document multiplied by respective output of term node. The input received by a document node is  $\sum_{i=1}^n w_i x_i$ . Here,  $w_i$  is  $(\text{tf}_i)(\text{idf}_i)$ , the weight of document-term weight and  $x_i$  is output of respective term node (i.e. the weight of query term weight). Therefore, input received by a document is product of query-term weight and document-term weight. This input is equivalent to a simple dot product.

### 3.3.2 Relevance Feedback with Neutral Nets

Relevance feedback can be used to get a relevant set of documents [18]. Relevance feedback can be implemented by adding a new set of links to the network. These new links connect DOCUMENT nodes back to TERM nodes as shown in figure 3.3.



**Fig. 3.3:** neural network with feedback.

The working of neural network for IR can be divided into four phases.

Phase1: Network is processed from QUERY layer to TERM layer. Information is sent from from the QUERY nodes to the TERM nodes.

Phase2: Information is processed from the TERM nodes to the DOCUMENT nodes.

Phase3: The third phase sends information from the DOCUMENT nodes to the TERM nodes for the documents that are deemed to the relevant. The relevant documents are identified manually or the top n documents may be deemed relevant.

Phase4: The TERM nodes are activated, if they exceed a threshold parameter. The TERM-DOCUMENT links are used to send the newly defined weights obtained during the relevance feedback phase to the DOCUMENT nodes.

After completion phase 4 the DOCUMENT nodes get score with a value that indicates the effect of a single iteration of relevance feedback.

### 3.3.3 Learning Modification to the Neutral Network

In typical VSM, relevance feedback is used to adjust the score of an individual query [18]. This relevance is not used for subsequent query. That is, system does not gain any knowledge from any prior relevance assessments.

To exploit relevance feedback into subsequent queries, neural network is an efficient tool. In the neural network as described just above, the document nodes add a new signal called the learning signal. The learning signal is defined as

$$\text{Learning signal} = \begin{cases} 1 & \text{if the user judges the document as relevant.} \\ 0 & \text{if document is not judged.} \\ -1 & \text{if document is judged as non-relevant.} \end{cases}$$

Term-document links are then adjusted based on the difference between the user assessment and the existing document weight. Documents with high weights that are deemed relevant do not result in much change to the network. A document weighted 0.90 will have a  $\delta$  of  $1-0.90=0.10$ , so each of its term-document links will be increased by only ten percent. A document with a low weight that is deemed relevant will result in a much higher adjustment to the network.

# Application of Genetic Algorithm in IR

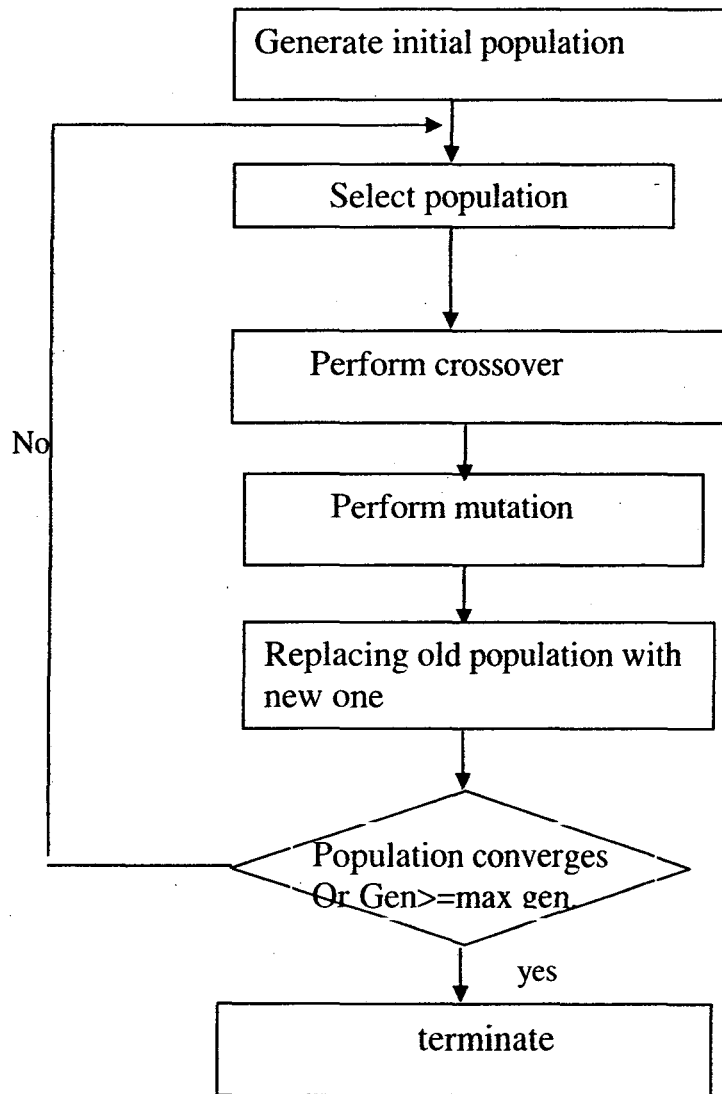
---

### 4.1 Introduction to Genetic Algorithm (GA)

A genetic algorithm (GA) is a search technique used in computing to find exact or approximate solutions from a domain space to optimize and search problems. Genetic algorithm is a particular class of evolutionary algorithm (EA) that uses techniques inspired by evolutionary biology such as inheritance, mutation, selection and crossover. Solutions in the domain space of the search, usually real numbers over some range, are encoded as bit strings, called chromosomes and each bit position in the string is called a gene [20].

An initial population is randomly selected from domain space. The members of this initial population are each evaluated for their fitness or goodness. A Fitness based selection called roulette wheel is generally used to select the members which can be possible candidates for next generation. From the selected population of chromosomes, a new population (new generation) is generated using genetic operators- crossover and mutation. When a pair of chromosomes in the new population is chosen at random to exchange genetic material, their bits, in mating operation, called crossover. This produces two new chromosomes that replace the parents. Randomly chosen bits in the offspring are flipped, called mutation. This process continues until no longer improvement in population or when a preset maximum number of generations is reached. Fig. 4.1 shows flow chart of GA.

GA is very different from most of the traditional optimization methods based on search methods- calculus-based, enumerative. Calculus-based methods require continuous and derivative existence domain but real world of search is fraught with discontinuities and noisy spaces. Enumerative methods work within a finite search space or a discretized infinite search space. This method is not efficient for different types of search space.



**Fig. 4.1:** Flow chart of GA



## 4.2 Application of GA in IR

GA has been applied to solve many important IR problems including: automatic document description, query definition, matching function learning, image retrieval, document clustering.

Gordon developed first approach to document indexing by GA [21]. Salton [54] uses genetic algorithms to select good indexes. Yang *et al.* proposed a technique using GA to choose weights for search terms in a query on the basis of user feedback [62]. Michalewicz [42] emphasized on models of genetic operators as observed in nature, such as crossover and mutation. Morgan and Kilgour suggested an intermediary between the user and IR system employing GAs to choose search terms from a thesaurus and dictionary [46]. Boughanem et al. [10], Horng and Yeh [29], and Vrajitoru [58] proposed new crossover and mutation operators, applicable for information retrieval. Vrajitoru suggested that a large population size is necessary for a better learning [59]. Despite the successes, little use has been made of genetic algorithms for Ad-Hoc queries. Harman observed different IR systems returning substantially different results, yet maintaining approximately equal performance [23]. Bartell *et al.* combined the output of different ranking functions to improve performance [2]. Fan *et al.* proposed an algorithm for indexing function learning, whose aim is to obtain an indexing function for the key term weighting of a documentary collection to improve IR process [47].

### 4.2.1GA for Document Description

A key problem in IR is finding a good representation for a document. A single representation can not reflect the content of a document. Multiple representations are required in order to match the subject content of document and requirements of users. But, from these multiple representations, some which are more appropriate should be chosen. This is a some kind of optimization problem. GA has been used to evaluate better representations of a document. A procedure of GA for this problem is given below.

Initial population consists of multiple representations for each document. A fixed set of queries for the document is then identified. A vector similarity measure is used

to measure the fitness of a given representation. The fitness of solution is based how well query is able to retrieve a particular document representation. This initial population then evolves using genetic operators: crossover and mutation. The output will be the set of the best representations of document [2].

#### 4.2.2 GA for Query Weight

A query is made of one or more terms. Each term in a query has its own importance and hence should have different weight. To assign relevant weight to each term of query, weight should be optimized and it can be done in a better way by using GA. Initially, the original query is taken without any weights. Then initial population is simply composed of randomly generating a specific no. of sets of weights for terms in the original query. This initial population evolves by using genetic operators: selection crossover and mutation. The output will be a set of best representations of the query [18].

#### 4.2.3 GA for Matching Function

As discussed in chapter 1, there exists many matching functions for finding similarity between query and document [18]. Each of these similarities has their properties. In order to increase the relevance of retrieved documents, researchers have tried to combine different similarity measures. Weights can be assigned to these similarity measures, in order to determine relevant importance of a similarity measure. GA can be used to find optimum weights for these similarity measures.

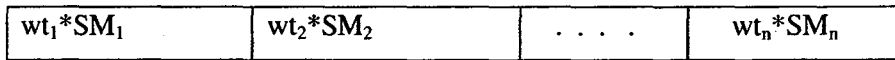
#### Problem Formulation

In this problem a combined similarity measure consisting of different similarity measures is used. Appropriate weights are assigned to these measures so as to achieve maximum retrieval efficiency. Combined similarity measure (CSM) is given by

$$\text{Combined\_similarity\_measure} = \sum_{i=1}^M (wt_i * SM(D, Q)) \quad (4.1)$$

Where  $SM_i(D, Q)$  denotes the similarity measure that is used to calculate the score of document D for given query Q,  $wt_i$  denotes the weights for  $i^{\text{th}}$  similarity measure and M is total number of similarity measures.

An individual member of population is represented in the following way:



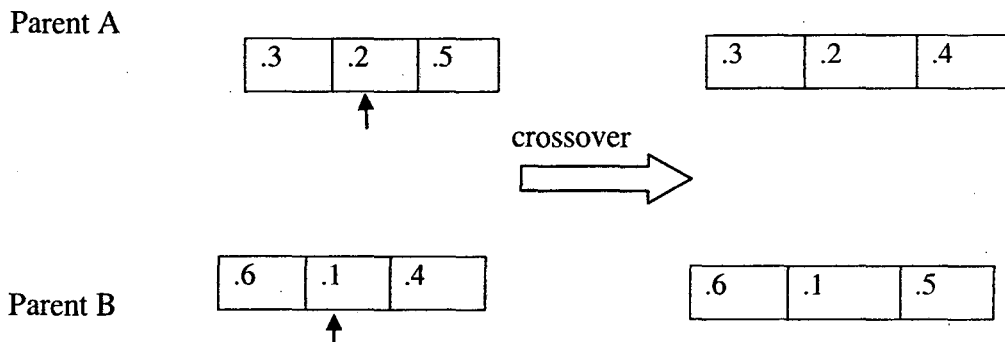
Here  $wt_1, wt_2, \dots, wt_n$  are weights which are initially generated randomly.

Fitness functions can be defined based on recall and precision. One of the fitness functions is:

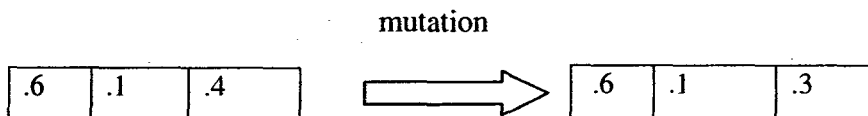
$$\text{Fitness} = \frac{(2 * \text{recall} * \text{precision})}{\text{recall} + \text{precision}} \quad (4.2)$$

This initial population evolves by using genetic operators: crossover and mutation

Following is an example for our crossover operation.



Following is an example for our mutation operation.



### 4.2.3.1 Proposed Algorithm and Experiments on Matching Function

Based on the role of GA for document matching as discussed above, we have proposed an algorithm to use GA for improving the document matching function.

GA has been used here to find optimum weights of combined similarity measure (as proposed in eqn 4.1.) used for matching document and query.

The proposed algorithm is as follows:

**Algorithm**

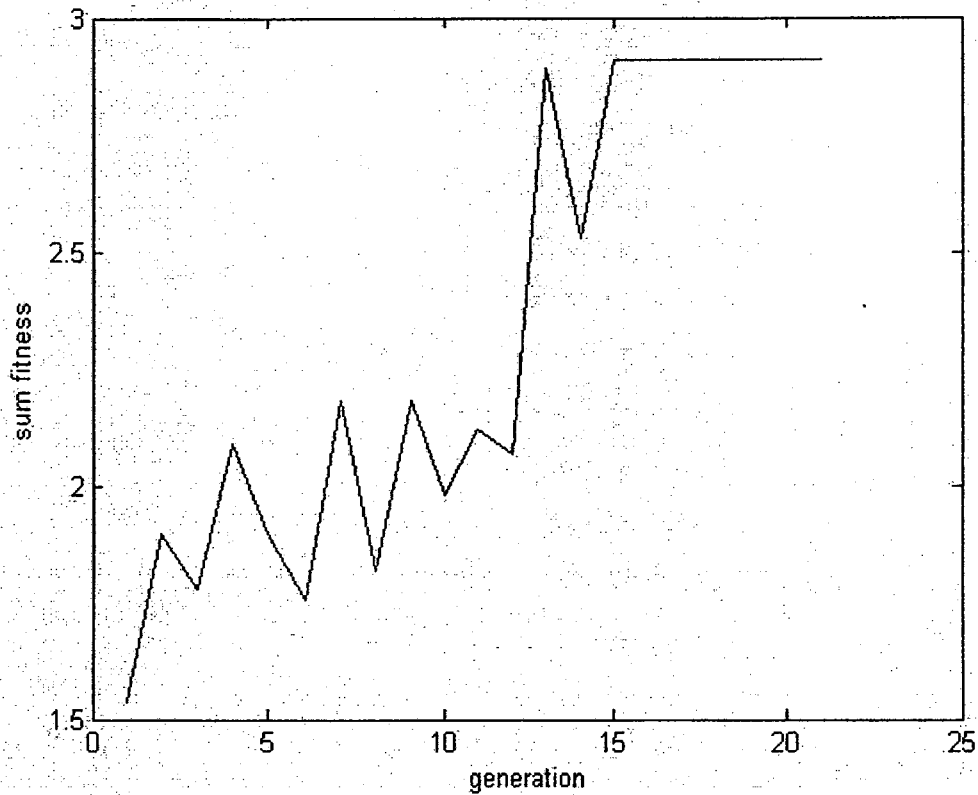
1. Preprocess the documents.
2. Find unique words from all documents.
3. Assign weights to words.
4. Generate initial population

$wt_1 * SM_1$	$Wt_2 * SM_2$	----- $wt_n * SM_n$
---------------	---------------	---------------------

5. Repeat step 6 until population converges
6. For each member of population
  - a) Find combined similarity measure for each document.
  - b) Sort documents based on combined similarity measure.
  - c) Select top N documents as relevant.
  - d) Calculate fitness based on recall/precision (using eqn. 4.2).
  - e) Select members for next generation.
  - f) Perform crossover.
  - g) Perform mutation.
  - h) Generate new population.

**Experiment and results**

We have implemented the algorithm by taking a combined similarity measure considering cosine and jaccard measure as individual similarity measures. We have implemented the algorithm on Adi data set using MATLAB software. Adi has 82 documents and 35 queries. We have experimented on 10 queries. Experiments were run for 50 generations. Crossover rate was 0.7 and mutation rate was set to 0.01. Fig 4.2 shows variation of a sum of fitness with generation number for a specific query 27 of adi data set . Optimized weight of w1 and w2 obtained after the convergence of GA are 0.8481 and 0.6416 respectively. As it is clear from the graph (fig. 4.2) that the average fitness has increased and becomes constant after 15<sup>th</sup> generation.



**Fig. 4.2:** curve showing variation of sum of fitness with generation number for a specific query of adi data set.

Measures	Recall	Precision
Cosine measure	0.40	0.51
Jaccard measure	0.67	0.45
CSM (in first gen.)	0.39	0.55
CSM(in 20 gen.)	0.73	0.83

**Table 4.1:** table showing recall and precision values for different similarity measures.

Table 4.1 shows that combined similarity measures (CSM) improves the retrieval efficiency. Moreover CSM using GA gets the better result than CSM in first generation. The results of experiments are encouraging indicating that genetic algorithm can be explored for designing good similarity measures by combining different similarity measures.

### Conclusion

---

The motivation of this work was to explore the role of soft computing to increase the efficiency of IR. We observe that fuzzy sets have been used for designing better documents and query representation. Fuzzy similarity measures have been developed to match document and query properly. Further fuzzy sets can provide a framework representing user needs, which are often imprecise and uncertain. Another emerging area is concept based information retrieval. We have explored this area and suggested an algorithm for concept based retrieval, which has been implemented on a small data set. Neural networks have been mainly used for learning of query weights using user relevance feedback, information visualization by converting high order information to low order information. Further they have been used for document classification and clustering. GA has been used for finding good representation of documents so that they can improve retrieval efficiency. GA's have also been used for finding optimal weights for query terms. Recently GA has been used for finding optimum weights for different similarity measures used for matching query and document. We have provided an algorithm suggesting use of GA for optimizing combined similarity measure. We have done experiment on adi dataset to improve algorithm. The results are encouraging.

We observe that www is emerging scope of applying soft computing techniques for efficient IR (especially from web data). A lot more can be explored, further hybrid methods can deal with many of existing problems in IR.

## *References*

1. Azzopardi, L., Girolami M. L., and Van Rijsbergen, C.J., "Topic Based Language Models for ad hoc Information Retrieval", In Proceedings of the International Joint Conference on Neural Networks, Budapest, Hungary, 2004.
2. Bartell , B.T., Cottrell, G.W. and Belew, R.K., "Automatic combination of multiple ranked retrieval systems", In proceedings of the 17<sup>th</sup> ACM SIGIR conference on information retrieval, pp. 173-181, 1994.
3. Belew, R. K., "Adaptive information retrieval", In Proceedings of the Twelfth Annual International ACM/SIGIR Conference on Research and Development in Information Retrieval NY, NY: ACM Press, pp. 11-20,1989.
4. Bezdek, J. C., Biswas, G., and Huang, L. Y. "Transitive closures of fuzzy thesauri for information-retrieval systems", International Journal of Man-Machine Studies, 25(3), 1986.
5. Bookstein, A., "Fuzzy requests: an approach to weighted boolean searches", Journal of the American Society for Information Science, 31(4), pp. 240-247, 1980.
6. Bordogna G. and Pasi, G. Controlling , "Retrieval trough a User-Adaptive Representation of documents", International Journal of Approximate Reasoning, 12:317-339, 1995.
7. Bordogna G., and Pasi G., "Linguistic aggregation operators of selection criteria in fuzzy information retrieval", International Journal of Intelligent Systems, 10, pp. 233-248, 1995.
8. Bordogna, G. and Pasi, G., "Soft fusion of Information Accesses", Fuzzy Sets and Systems, 148, pp.205-218, 2004.
9. Bordogna, G., Carrara, P., and Pasi, G., "Query term weights as constraints in fuzzy information retrieval", Information Processing and Management, 27(1), pp.15-26, 1991.
10. Boughanem, Chrisment, and Tamine. "On using genetic algorithms for multimodal relevance optimization in information retrieval", Journal of the American Society for Information Science and Technology, 53(11), pp. 934-942,2002.

11. Cabrera, I.P., Cordero, P., and Ojeda-Aciego, M., "Fuzzy logic, soft computing, and applications". Lecture Notes in Computer Science, 5517:236-244, 2009.
12. Chen, H., "Machine learning for information retrieval: neural networks, symbolic learning, and genetic algorithm", Journal of the American Society for Information Science, 46(3): 194-216, 1995.
13. Chen, H., and Kim, J., "GANNET: Information retrieval using genetics algorithms and neural networks", Center for Management of Information, College of Business and Public Administration, University of Arizona, 1993.
14. Chen, H., Basu, K., and Ng, T., "An algorithmic approach to concept exploration in a large knowledge network (automatic thesaurus consultation): Symbolic branch-and-bound vs. connectionist Hopfield net activation", Journal of the American Society for Information Science.
15. Chung, Y.M., Potternger, W.M., and Schatz, B. R., "Automatic subject indexing using an associative neural network", The 3rd ACM conference on digital libraries, 59-68, 1993.
16. Cordon, O., Alcalá, R., Alcalá-Fernández, J., and Rojas, I., "Genetic fuzzy systems: What's next?" an introduction to the special section. IEEE Trans. Fuzzy Systems, 15(4):533-535, 2007.
17. Crestani, F., Pasi, G.: Soft information retrieval: Applications of fuzzy Set Theory and Neural Networks. In "Neuro Fuzzy Techniques For Intelligent Information Systems ". N Kasabov and Robert Kozmz Editors, Physica – Verlag, Springer-Verlag Group, (1999) 287-313.
18. David, Grossman and Frieder, "Information Retrieval: algorithm and heuristic," kulwar academic prees, 1998.
19. Doszkocs, T. E., Reggia, J., and Lin, X., "Connectionist models and information retrieval", Annual Review of Information Science and Technology (ARIST), 25, 209-260, 1990.
20. Goldberg, D. E. "Genetic Algorithms in Search, Optimization, and Machine Learning", Addison-Wesley, Reading, Mass, 1989.
21. Gordan, M., "Probabilistic and genetic algorithms for document retrieval", Communications of ACM 31 (120) 1208-1218, 1998.



22. Gurney, K., Neural Nets by Kevin Gurney, 1999. <http://www.shef.ac.uk/psychology/gurney/notes/index.html>, visited on Jan. 3rd.
23. Harman, D.K., "Over view of the first text retrieval conference (TREC-1)," in proceedings of the First Text Retrieval Conference, D.K. Harman, Ed. NIST Special Publication 500-207, pp. 1-20, 1993.
24. Hatano, K., Qian, Q., and Tanaka, K., "A SOM-based information organizer for text and video data", Proceedings of the Fifth International Conference on Database Systems for Advanced Applications, 205-214, 1997.
25. Hathaway, R.J., Bezdek, J.C., and Hu, Y., "Generalized Fuzzy C-Means Clustering Strategies Using Lp Norm Distances", IEEE Transactions on Fuzzy Systems, 8(5), pp. 576-582, 2000.
26. Haykin, "Neural Networks: A Comprehensive Foundation", second Edition, Prentice-Hall, 1999.
27. Herrera, and Verdegay, J. L., "Genetic Algorithms and Soft Computing." Physica-Verlag, 1996.
28. Hirst, G., and St-Onge, D., "Lexical chains as representations of context for the detection and correction of malapropisms", In: Christiane Fellbaum (editor), WordNet: An electronic lexical database, Cambridge, MA: The MIT Press, 1998.
29. Horng, J. T., and Yeh. "Applying genetic algorithms to query optimization in document retrieval", Information Processing & Management, 36(5), pp. 737-759, 2000.
30. Kang, B., and Kim, V., Lee, S., "Exploiting concept clusters for content-based information retrieval", Information Sciences, Vol. 170, Issues 2-4, pp. 443-462,
31. Kang, B., Kim, D., and Kim, H., "Fuzzy information retrieval indexed by concept identification", LNAI 3658, pp. 179-186, 2005.
32. Kennedy, J., and Eberhart, R. C., "Particle swarm optimization". In IEEE International Conference on Neural Networks, pg. 1942-1948, 1995.
33. Klir, G.J. and Folger, T.A., "Fuzzy sets, uncertainty and information", Englewood Cliffs, NJ: Prentice Hall, 1988,
34. Kohonen, T., "Self-Organization and Associative Memory". Series in Information Sciences, 2nd Edition. Berlin: Springer-Verlag, 1988.
35. Kohout, L. J., Keravanou, E., and Bandler, W., "Information retrieval system using fuzzy relational products for thesaurus construction", In Proceedings IFAC Fuzzy Information, Marseille, France, pp.7-13, 1983.

36. Kraft, D. H., Bordogna, G. and Pasi, G., "An extended fuzzy linguistic approach to generalize Boolean information retrieval", *Journal of Information Sciences -Applications*, 2(3), 1995.
37. Kraft, D.H., Pasi, G., Bordogna, G., "Vagueness and uncertainty in information retrieval: how can fuzzy sets help?" In: *Proceedings of IWRIDL 2006*, pp. 1–10, 2006.
38. Kwok, K. L., "A neural network for probabilistic information retrieval". In *Proceedings of the Twelfth Annual International ACM/SIGIR Conference on Research and Development in Information Retrieval*. NY, NY: ACM Press, pp. 2 I-30,1989.
39. Lin, X., Soergel. D., and Marchionini. G., "A self-organizing semantic map for information retrieval", In *Proceedings of the Fourteenth Annual International ACM/SIGIR Conference on Research and Development in Information Retrieval*, Chicago, pp. 262-269, 1991
40. MacLeod, K. J., and Robertson, W., "A neural algorithm for document clustering", *Information Processing and Management*, 27, 337- 346, 1991.
41. Mantaras, Cortes, Manero, and Plaza, "Knowledge engineering for a document retrieval system", *Fuzzy Sets and Systems*, 38(2), 1990.
42. Michalewicz, "Genetic Algorithms +Data Structures=Evolution Programs", Springer-Verlag, 1996.
43. Miyamoto, S. "Fuzzy sets in Information Retrieval and Cluster Analysis", Kluwer Academic Publishers, 1990.
44. Miyamoto, S. and Nakayama, K., "Fuzzy information retrieval based on a fuzzy pseudo thesaurus.", *IEEE Transactions on Systems, Man and Cybernetics*, SMC- 16(2), 1986
45. Molinari A. and Pasi G., "A Fuzzy representation of HTML documents for Information Retrieval Systems", In *Proceedings of IEEE Int. Conf. on Fuzzy Systems*, New Orleans,1996.
46. Morgan and Kilgour. "Personalising on-line information retrieval support with a genetic algorithm". In A. Moscardini, & P. Smith (Eds.), *PolyModel 16: Applications of artificial intelligence*, 1996, pp. 142–149
47. Pathak, Gordon and Fan. "Effective information retrieval using genetic algorithms based matching functions adaption", in: *Proc. 33rd Hawaii International Conference on Science (HICS)*, Hawaii, USA, 2000
48. Pearl, J., "Probabilistic Reasoning in Intelligent Systems", Morgan Kaufmann, 1988.

49. Prasad, B., "Introduction to Neuro-Fuzzy Systems", Advances in Soft Computing Series, vol. 226, Springer-Verlag, 2000.
50. Radecki, T. "Fuzzy set theoretical approach to document retrieval". Information Processing and Management, 15(5), pp. 247-260, 1979.
51. Rajasekaran and Vijayalakshmi Pai Neural, "Networks, Fuzzy Logic and Genetic Algorithms: Synthesis and Applications", PHI, 2005.
52. Rose, D. E., and Belew, R. K., "A connectionist and symbolic hybrid for improving legal research", International Journal of Man-Machine studies, 35, pp.1-33, 1991.
53. Salton, G, and Buckley, C., "On the use of spreading activation methods in automatic information retrieval". In: Chiaramella, Y. (ed.). 11th International Conference on Research & Development in Information Retrieval, NY: Association for Computing Machinery, pp. 147-160, 1988.
54. Salton, G., and Buckley, C., "Term weighting approaches in automatic text retrieval," Information Processing and Management, Vol. 24, No. 5, pp. 513-523, 1988.
55. Stairmand, M.A., "Textual context analysis for information retrieval", Proceedings 20th ACM SIGIR Conference on Research and Development in Information Retrieval, ACM Press, 1997.
56. Stokes, N., and Carthy, J., "Combining semantic and syntactic document classifiers to improve first story detection", Proceedings of the 24th ACM SIGIR Conference, New Orleans, pp.424, 2001,
57. Sushmita Mitra, "Data Mining", Wiley Student Edition
58. Vrajitoru, D., "Crossover improvement for genetic algorithms in information retrieval", Information Processing and Management, 34(4) 405-415, 1998.
59. Vrajitoru, D., "Large Population or Many Generations for Genetic Algorithms? Implications in Information Retrieval", Soft Computing in Information Retrieval. Techniques and Applications, pp. 199-222. Physica-Verlag, 2000.
60. Wang, W.J., "New similarity measures on fuzzy sets and on elements", Fuzzy Sets and Systems, Volume 85, 1997, pp. 305-309.
61. Wilkinson, R., Hingston, P., and Osborn, T., "Incorporating the vector space model in a neural network used for document retrieval", Library Hi Tech, 10, pp. 69-75, 1992.
62. Yang, Korfhage, and Rasmussen, "Query improvement in information retrieval using genetic algorithms- a report on the experiments of the TREC project", in proceedings of the 1<sup>st</sup> text retrieval conference (TREC-1), 1992, pp. 31-58, 1992.

63. Zadeh, L. A., "Soft computing and fuzzy logic", Software, IEEE, 11(6):48-56, 1994.

## *Appendix 1.*

### **doc. 1**

Cricket is a bat-and-ball team sport that originated in southern England. The earliest definite reference is dated 1598, and it is now played in more than 100 countries. There are several forms of cricket; at its highest level is Test cricket, in which the current leading national team is Australia. Test cricket is followed in rank by One Day International cricket, whose last World Cup was also won by Australia; the tournament was televised in over 200 countries to a viewing audience estimated at more than two billion viewers. The current leading One Day International team is South Africa.

### **doc. 2**

Cricket is a high performance, extremely flexible system for monitoring trends in time-series data. Cricket was expressly developed to help network managers visualize and understand the traffic on their networks, but it can be used all kinds of other jobs, as well.

Cricket has two components, a collector and a grapher. The collector runs from cron every 5 minutes (or at a different rate, if you want), and stores data into a data structure managed by RRD Tool. Later, when you want to check on the data you have collected, you can use a web-based interface to view graphs of the data.

### **doc. 3**

Cricket was founded by England in the 1800s in an attempt for them to win a sport of some sorts because they are crap at everything else. Not a game of two halves, not in any way 'funny' or 'old', and played by only 13 and a half nations around the world (hey, that's more than baseball), cricket is a game that is played for ages, the conclusion of which is that England lose. Eventually he came up with the idea, that if every person in India joined in as a single team in one big game, not only would cricket become the longest, dullest, and most pointless sport in existence, but the side might also be able to beat win independence from England

### **doc. 4**

Cricket is the unofficial national sport of India, and its development has been closely tied up with the history of the country, mirroring many of the political and cultural developments around issues such as caste, religion and nationality. Though cricket is indubitably the most popular sport in India, it is not the nation's national sport.

## **doc. 5**

The term yoga comes from a Sanskrit word which means yoke. Traditionally, yoga is a method joining the individual self with the Divine, Universal Spirit, or Cosmic Consciousness. Physical and mental exercises are designed to help achieve this goal. On the physical level, asanas are designed to tone, strengthen, and align the body. On the mental level, breathing techniques (pranayama) and meditation (dyana) are used to quiet, clarify, and discipline the mind.

## **doc. 6**

Yoga was introduced to American society in the late 19th century by Swami Vivekananda, the founder of the Vedanta Society. He believed that India has an abundance of spiritual wealth and that yoga is a method that could help those who were bound by the materialism of capitalist societies to achieve spiritual well-being.

## **doc. 7**

A survey released in May 2004 by the National Center for Complementary and Alternative Medicine (CAM) focused on who used complementary and alternative medicine (CAM), what was used, and why it was used in the United States by adults age 18 years and over during 2002.[3] According to this survey, Yoga was the 5th most commonly used CAM therapy (2.8%) in the United States during 2002. [4] Yoga is considered a mind-body intervention that is used to reduce the health effects of generalized stress.

## **doc. 8**

Yoga can also provide the same benefits as any well-designed exercise program, increasing general health and stamina, reducing stress, and improving those conditions brought about by sedentary lifestyles. Yoga has the added advantage of being a low-impact activity that uses only gravity as resistance, which makes it an excellent physical therapy routine; certain yoga postures can be safely used to strengthen and balance all parts of the body.

