# KEYWORD COMPETITION APPROACH IN RANKED DOCUMENT RETRIEVAL

POLTAK SIHOMBING

UNIVERSITI SAINS MALAYSIA

2010

# KEYWORD COMPETITION APPROACH IN RANKED DOCUMENT RETRIEVAL

by

POLTAK SIHOMBING

Thesis submitted in fulfillment for the
requirements for the degree of
Doctor of Philosophy

June 2010

## ACKNOWLEDGEMENT

During the time I have been studying at Universiti Sains Malaysia, I have been truly fortunate to have had the guidance, support, and friendship from a number of people who help me develop academically and personally so that I can finish writing this thesis. I acknowledge that this thesis would not finish if they didn't help me so much.

I would like to thank Assoc. Prof. Dr. Putra Sumari as my supervisor and Prof. Dr. Abdullah Embong as my field supervisor, for their continuous support and guidance from the very beginning until the completion of my study at Universiti Sains Malaysia. Their supporting critiques and thoughtful views will never be forgotten. To the Dean of the School of Computer Sciences and all the staff, I would also like to express my thankfulness for all the facilities and services that I have received during my study. To the Dean of the IPS (Institut Pengajian Siswazah) Universiti Sains Malaysia, I would also like to express my gratitude to them for the excellent services during my study. My greatest thanks would also be addressed to the Rector of Universitas Sumatera Utara, the Dean of the Faculty Mathematics and Natural Sciences for their widest support and permission given to continue my study.

To my wife dr. Lindawati, my beloved children Johan, Kiko, Edwin, Erwin, and Jessica, I should also express my gratitude for sacrificing their wonderful life in Indonesia. May Allah bestows His blessing for them and always provide us with happiness. My deep appreciation would also be given to my Mother who always pray for me. To My brother, sisters, and mother in-law, brother and sisters' in-law that should be very long to mention individually, I would also like to thank for their support. May Allah also give His prosperity and change to pay them all moral and financial debts I owe them. To my nephews and nieces, I would also like to say my thanks for their support to my study. For all in the family, may Allah give His blessing and opulence for them.

Lastly, to Prof.Dr. Muhammad Zarlis, the head of computer science department, who has given widest support, Drs. Ronsen Purba, MSc, Darwis Manalu, S.Kom, MM, Drs. Lungguk Sitorus, M.Ed, who have given English correction and all of my colleagues at Computer Sciences Department at Universitas Sumatera Utara, who have given their support during my study, I thank them all and hopefully successful life and career will always accompany us. Thank you very much.

## TABLE OF CONTENTS

**CHAPTER 3: LITERATURE SURVEY**

**CHAPTER 4: KEYWORD COMPETITION (KC) SCHEME**



**CHAPTER 5: KEYWORD BASED RANKING SCHEME**

## LIST OF TABLES

**Table** **Page**

## LIST OF FIGURES

## LIST OF PROCEEDINGS & PUBLICATIONS

1. Poltak Sihombing, Putra Sumari, Abdullah Embong. Effective Information Retrieval using Genetic Algorithm based Queries Competition. In Proceeding of the National Conference Software Engineering System (NaCSES'07). Kuantan, Malaysia. 2007.
2. Poltak Sihombing. Optimizing of Retrieval: A query competition approach to information retrieval. In Proceeding of the 5th International Conference on Numerical Analysis in Engineering (NAE2007). Padang, West Sumatera, Indonesia. 2007.
3. Poltak Sihombing. Search Engine to Determine Similarity Ranking of Document Retrieval Automatically by using Genetic Algorithm (GA). In Proceeding of the National Conference Software Engineering System (NaCSES'07). Kuantan, Malaysia. 2007.
4. Poltak Sihombing. Automatic Generation of Matching Function by Genetic Algorithm for Information Retrieval. In Proceeding of the 'Seminar Nasional Riset Teknologi Informasi. SRITI 2007'. Jogjakarta, Indonesia. 2007.
5. Poltak Sihombing. Optimizing Information Retrieval System by Keyword Selection in Genetic Algorithm. In Proceeding of the 'Seminar Nasional Riset Teknologi Informasi. SRITI 2007'. Jogjakarta, Indonesia. 2007.
6. Poltak Sihombing, Putra Sumari, Abdullah Embong. Application of Genetic Algorithm to Determine a Document Similarity by Different Formulation**.** In Proceeding of the Regional Computer Science Postgraduate Conference (ReCSPC'06). Malaysia. 2006.
7. Poltak Sihombing, Putra Sumari, A.Embong. A Comparison of Document Similarity in Information Retrieval System by different formulation. In Proceeding of the IMT-GT (Indonesia Malaysia Thailand) Conference. Penang, Malaysia. 2006.
8. Poltak Sihombing. Using Genetic Algorithm in Document Similarity Access. In Proceeding of the 'Seminar Nasional Riset Teknologi Informasi. SRITI 2006'. Jogjakarta, Indonesia. 2006.

9. Poltak Sihombing. Application of Genetic Algorithm to Determine a Document Similarity by Cosine Formulation. In Proceeding of the 2nd IMTGT_Conference. Penang, Malaysia. 2006.
10. Poltak Sihombing, Putra Sumari, A.Embong. Application of Genetic Algorithm to Determine a Document Similarity Level in IRS. In Proceeding of the 1st Malaysian Software Engineering Conference 2005 (MySec'05). Penang, Malaysia. 2005.
11. Poltak Sihombing, Putra Sumari, Abdullah Embong, A Technique of Probability in Document Similarity Comparison in Information Retrieval System. In Proceeding of the IMT-GT Conference. Parapat, Indonesia. 2005.
12. Poltak Sihombing, Abdullah Embong, Putra Sumari. Application of GA to Determine a Document Similarity Level in IRS. In Proceeding of the Malaysian Software Engineering Conference '05 (MySec05). Penang, Malaysia. 2005.
13. Poltak Sihombing. A Technique of Probability in Ranked Document Retrieval Similarity. Jurnal SIFO Mikroskil. ISSN.1412-0100. Vol.04.No.02. Medan, Indonesia. 2004.
14. Poltak Sihombing. Implementation of Genetic Algorithm in Information Access by Different Formulation. Journal of Information Science and Technology-Universitas HKBP Nommensen (UHN). Medan, Indonesia. 2007.

**PENDEKATAN PERSAINGAN KATA KUNCI BAGI DAPATAN KEMBALI DOKUMEN SECARA TERSUSUN**

## ABSTRAK

Sistem Dapatan Kembali Maklumat pada masa kini berdepan dengan pangkalan data yang sangat besar kerana adanya ruang storan yang besar, peranti storan berbilang dan media storan yang berbeza. Pertumbuhan data yang cepat dalam pangkalan data akan mengakibatkan dalam masa yang singkat data yang disimpan tidak dapat diuruskan dengan baik dan menyebabkan timbulnya masalah dalam dapatan kembali maklumat, iaitu pengguna tidak mendapat kembali dengan tepat maklumat yang dikehendaki daripada pangkalan data. Ini merupakan salah satu masalah penting di dalam sistem dapatan kembali maklumat. Penggunaan kata kunci merupakan salah satu kaedah dalam sistem dapatan kembali maklumat yang boleh menyelesaikan masalah ini. Di dalam tesis ini, kami mencadangkan satu kaedah dalam Algoritma Genetik [*Genetic Algorithm* (GA)] yang dikenali sebagai pendekatan persaingan kata kunci [*keyword competition* (KC)]. KC merupakan skim bagi mencari kata kunci yang terbaik, yang dikenali sebagai ’kata kunci penyelesaian’ [*keyword solution* (KS)], antara kata kunci yang ada. Kemudian KS berkenan akan dipadankan dengan koleksi dokumen dalam pangkalan data bagi mendapatkan kembali dokumen yang paling relevan. Dalam kajian ini koleksi prosiding daripada BADAN TENAGA ATOM NASIONAL (BATAN) Indonesia, yang dibangun oleh Universitas Indonesia (UI), Jakarta digunakan sebagai set data piawai. Kami juga mencadangkan skim penyusunan berdasarkan kata kunci bagi dapatan kembali dokumen secara tersusun yang memberikan dokumen yang paling relevan kepada pengguna. Skim penyusunan berdasarkan kata kunci terdiri daripada dua (2) fasa utama; dinamai pemadanan kata kunci dan perumusan peratusan keserupaan. Dalam proses pemadanan kata kunci, sistem akan memadankan KS dengan mencari perkataan yang sama di dalam *title, abstract, dan keyword* dari koleksi dokumen dalam pangkalan data. Perumusan peratusan keserupaan digunakan untuk menyusun dokumen yang didapatkan kembali berasaskan nilai keserupaan. Skim ini telah diuji dengan dua rumus kesesuaian yang berbeza iaitu fungsi Jaccard dan fungsi Cosine. Kami kemudian membandingkan hasil KC dengan tingkat keserupaan dalam kaedah Hopfield. Sebuah prototaip yang dinamai ***J**ournal **B**rowser **S**ystem (**JBS**)* telah dibangunkan berdasarkan skim ini. Hasil yang dikumpul daripada JBS membuktikan bahawa kaedah persaingan kata kunci dan skim penyusunan berasaskan kata kunci memberi prestasi yang lebih baik berbanding kaedah Hopfield.

**KEYWORD COMPETITION APPROACH IN RANKED DOCUMENT RETRIEVAL**

## ABSTRACT

**I**nformation **R**etrieval **S**ystem **(IRS)** currently deals with tremendously large database due to the availability of huge storage spaces, multiple storage devices and different storage media. The rapid growth of data in the database will eventually render the data unmanageable and cause problems in retrieval, where the users are unable to retrieve the right document. This is one of the most important problems in IRS. The use of keywords is one of the methods in IRS which can solve this problem. In this thesis, we propose a methodology in GA (Genetic Algorithm) which is known as Keyword Competition (KC) approach. KC is a competition scheme in finding the best keyword, known as 'keyword solution' (KS), among the available keywords. The keyword solution is then matched to the document collection in the database in order to retrieve the most relevant document. In this research, the collection of proceedings of BADAN TENAGA ATOM NASIONAL (BATAN) Indonesia, presented by University of Indonesia (UI), Jakarta was used as a standard dataset. We also propose a keyword based ranking scheme aimed to better rank the retrieved document in the spirit of presenting the most relevant document to the users. Keyword based ranking scheme consists of two (2) main phases; namely keyword solution matching and similarity percentage formulation. In the keyword matching process, the system will match those KS by finding the same words in the *title, abstract & keyword* of each document collection in the database. The similarity percentage formulation is used to rank the retrieved document based on the similarity value. The scheme was tested with two different fitness formulations, i.e. Jaccard's function and Cosine's function. We then compare the result of KC to the similarity level in Hopfield method. A prototype called **J**ournal **B**rowser **S**ystem (**JBS**) based on this scheme was developed. The results collected from JBS provide the evidence that KC approach and keyword based ranking scheme give better performance compared to Hopfield method.

# ABSTRACT OF PROCEEDINGS & PUBLICATIONS

# EFFECTIVE INFORMATION RETRIEVAL USING GENETIC ALGORITHM BASED QUERIES COMPETITION*

Poltak Sihombing, Putra Sumari, Abdullah Embong.

## Abstract

We are addressing in this research is how to develop a retrieval system that can generate documents which are very relevant to user's expectations This paper describes a technique to select some of keywords in Genetic Algorithm (GA) by queries competition. An evaluation function for the *fitness* of each chromosome was selected based on Jaccard, Horng-Yeh, Cosine and Dice's formulation. These formulations are common measure of association in information retrieval. To initialize a population, we need first to decide the number of genes for each individual and the total number of chromosomes in the initial population. GA is basically based on natural biological evolution. The parent solution (chromosome) with the higher level of fitness has a bigger similarity percentage of documents, while those with lower level of fitness have less similarity percentage of documents. The result of this system, the user can retrieve the most relevant document from a database.

Keywords: Effective, information retrieval, similarity, queries competition, genetic algorithm

* *In* Proceeding of the *National Conference Software Engineering System (NaCSES'07). Kuantan, Malaysia. 2007.*

# OPTIMIZING OF RETRIEVAL: A QUERY COMPETITION APPROACH TO INFORMATION RETRIEVAL*

Poltak Sihombing.

**Abstract**

This paper describes a technique to optimize a similarity level of some documents retrieved by using query competition in Genetic Algorithm (GA). An evaluation function for the fitness of each chromosome was selected based on Jaccard, Horng-Yeh, Cosine, and Dice's formulation. By these formulations we can improve the performance of our system. To initialize a population, we need first to decide the number of genes for each individual and the total number of chromosomes (pop size) in the initial population. GA is basically based on natural biological evolution. We have implemented those techniques in a prototype of Journal Browser Search (JBS). By the similarity level of documents, the user can retrieve the most similar document from a database.

Keywords**:** optimize, retrieval, similarity, query competition, jaccard, horng-yeh, cosine, dice, genetic algorithm

* *The 5th International Conference on Numerical Analysis in Engineering (NAE2007, Padang, West Sumatera Indonesia. 2007.*

# SEARCH ENGINE TO DETERMINE SIMILARITY RANKING OF DOCUMENT RETRIEVAL AUTOMATICALLY BY USING GENETIC ALGORITHM (GA)*

Poltak Sihombing

**Abstract**

The most important problem of any Information Retrieval System (IRS) is to locate the most similar documents that have potential to satisfy the user information needs. In the IRS literature study, researchers have implemented several methods. This paper describes a technique of ranking function to determine a similarity level of document retrieval from database. Ranking function play a substantial role in the performance of IRS and search engine. We propose a ranking function to determine a similarity level of document automatically by using Genetic Algorithm (GA). GA is basically based on natural biological evolution. In our research design, a keyword represents a gene (a bit pattern), a document's list of keywords represents individuals (a bit string), and a collection of documents initially judged relevant by a user represents the initial population. The similarity between keyword (as a query) and document is calculated by Jaccard's formulation, Horng-Yeh's formulation, Cosine's formulation and Dice's formulation. These formulations are common measure of association in information retrieval. By the similarity ranking of document retrieval, the user can choose the most relevant document from the database.

Keywords**:** Ranking document, Information retrieval, similarity, genetic algorithm

* *In* Proceeding of the *National Conference Software Engineering System (NaCSES'07). Kuantan, Malaysia. 2007.*

# AUTOMATIC GENERATION OF MATCHING FUNCTION BY GENETIC ALGORITHM FOR INFORMATION RETRIEVAL*

Poltak Sihombing

**Abstract**

This paper describes a technique to determine automatic generation of matching function in order to get a similarity level of document by using Genetic Algorithm (**GA**). An evaluation function for the *fitness* of each chromosome was selected based on Jaccard, Cosine and Horng-Yeh's formulation. These formulations are common measure of association in information retrieval. To initialize a population, we need first to decide the number of genes for each individual and the total number of chromosomes in the initial population. GA is basically based on natural biological evolution. The parent solution (chromosome) with the higher level of fitness has a bigger similarity percentage of documents, while those with lower level of fitness have less similarity percentage of documents. The result shows that if the sum of generation is increased, then the sum of crossover and mutation is increased. The bigger percentage of similarity is in Cosine's formulation. By the matching function, the user can obtain and retrieve the most relevant document according to percentage of similarity.

Keywords: Matching function, information retrieval, document similarity, genetic algorithm.

**In* Proceeding of the *'Seminar Nasional Riset Teknologi Informasi. SRITI 2007'. Jogjakarta, Indonesia. 2007.*

# OPTIMIZING INFORMATION RETRIEVAL SYSTEM BY KEYWORD SELECTION IN GENETIC ALGORITHM*

Poltak Sihombing

## Abstract

Optimization of some keywords is essential in Information Retrieval System (IRS) in order to find the most potential keyword. This paper describes a technique to optimize some keywords of by using keyword competition in Genetic Algorithm (GA). In fitness function evaluation, we used Dice's formulation, and compare it with Jaccard's result, and Cosine's result in our research before. By these formulations we want to know the performance of each formulation in IRS. We used GA processing to optimize keyword in all of population. To initialize a population, we need first to decide the number of genes for each individual and the total number of chromosomes (pop size) in the initial population. GA is basically based on natural biological evolution. We have implemented those techniques in a prototype of Journal Browser Search (JBS). The JBS result provides percentage of similarity of document retrieval. In all test cases, we found that if the sum of generation in GA is increased, then the percentage of similarity is not increased. By all of test case, we found that the bigger similarity percentage is in Cosine.

Keywords**:** information retrieval, optimize, similarity, keyword selection, dice, genetic algorithm

**In* Proceeding of the *'Seminar Nasional Riset Teknologi Informasi. SRITI 2007'. Jogjakarta, Indonesia. 2007.*

# APPLICATION OF GENETIC ALGORITHM TO DETERMINE A DOCUMENT SIMILARITY BY DIFFERENT FORMULATION*

Poltak Sihombing, Putra Sumari, Abdullah Embong.

## Abstract

In recent years, we have witnessed an immense growth in the availability of document storage in Information Retrieval Systems (IRS). Storing documents in an IRS is no longer an issue due to the availability of huge storage space, multiple storage devices and different storage media, and the occurrence of various methods of document storage. The challenge is more on the retrieval of the right documents since documents stored in a database grow very fast and soon become unmanageable. The most important problem in IRS is to get the most relevant document from a database. In this study we have implemented the Genetic Algorithm (GA) by different formulation in an IRS prototype called the Journal Browser. The GA was to find a set keywords of documents which best matched the searcher's needs. An evaluation function for the fitness of each chromosome was selected based on Jaccard's formulation, Horng-Yeh's formulation, and Cosine's formulation. These formulations are used as common measure of association of keywords with some documents in a database. To initialize a population of the keywords, we need first to decide the number of genes for each individual and the total number of chromosomes in the initial population. GA is basically based on natural biological evolution theory. By the similarity percentage of documents, the user can choose the most relevant document from a database. The result shows that the similarity value of document is consistent even though the percentage of similarity may change.

Keywords**:** Information, retrieval, genetic algorithm, document, similarity

**In* Proceeding of the *Regional Computer Science Postgraduate Conference (ReCSPC'06). 2006*

# A COMPARISON OF DOCUMENT SIMILARITY IN INFORMATION RETRIEVAL SYSTEM BY DIFFERENT FORMULATION*

Poltak Sihombing, Putra Sumari, Abdullah Embong.

## Abstract

In this paper we are going to implement Horng-Yeh's formulation in ***I****nformation* ***R****etrieval* ***S****ystem, (IRS)* and to compare it with the Jaccard's formulation and Dice's formulation. In the previous research, we have developed the Jaccard and Dice's formulation in a prototype called the Journal Browser. Each technique has been implemented in IRS using Genetic Algorithm (GA). The objective of GA was to find a set of documents which best fit the searcher's needs. In this study, an evaluation function for the *fitness* of each chromosome was selected based on Horng-Yeh's score. This score is formulated to measure the relationship of the query with some documents in a database. To initialize a population of the queries, we need first to decide the number of genes for each individual and the total number of chromosomes in the initial population. GA is basically based on natural biological evolution theory. The parent solution (chromosome) with the higher level of fitness has a bigger similarity percentage of documents, while those with lower level of fitness have less similarity percentage of documents. By the similarity percentage of documents, the user can choose the most relevant document from the database.

Keywords: Database, information, retrieval, document, similarity, genetic algorithm.

* *In* Proceeding of the IMT-GT (Indonesia Malaysia Thailand) Conference. Penang, Malaysia. 2006

# USING GENETIC ALGORITHM IN DOCUMENT SIMILARITY ACCESS*

Poltak Sihombing

Abstract

In recent years, storing documents in an Information Retrieval System (IRS) is no longer an issue due to the availability of huge storage space. The challenge is more on the retrieval of the right documents since documents stored in a database grow very fast and soon become unmanageable. This situation often resulted in difficulty to retrieve a document from a database which is expected to be very relevant to a query, and this has become the most important problem in IRS In this paper we are going to implement Genetic Algorithm (GA) in document similarity access. The GA was to find a set keywords of documents which best matched with the searcher's needs. An evaluation function for the *fitness* of each chromosome was selected based on different formulation. The formulation of score is used as common measure of association of keywords. In the previous research, we have developed the Jaccard's formulation in a prototype called the **JBS** (**J**ournal **B**rowser **S**ystem). In this work, we implemented the GA by Cosine's fitness formulation in document similarity access. We found that the Cosine have the bigger similarity value than Jaccard.

Keywords**:** Teks retrieval, genetic algorithm, document similarity, Cosine

**In* Proceeding of the *'Seminar Nasional Riset Teknologi Informasi. SRITI 2006'. Jogjakarta, Indonesia. 2006*

# APPLICATION OF GENETIC ALGORITHM TO DETERMINE A DOCUMENT SIMILARITY BY COSINE FORMULATION*

Poltak Sihombing

## Abstract

There are many techniques to determine a similarity of document in **IRS** (***I****nformation* ***R****etrieval* ***S****ystem).* In this paper we propose a GA technique in IRS. The goal of GA (Genetic Algorithm) is to select set of keywords in order to find the best keywords based on their fitness. The *fitness* function of each chromosome was selected based on Cosine's method. In GA processing, we need first to decide the number of genes for each individual and the total number of chromosomes in the initial population. The parent solution (chromosome) with the higher level of fitness has a bigger similarity percentage of documents, while those with lower level of fitness have less similarity percentage of documents. We found that even though the sum of crossover and mutation is increased, then the percentage of document similarity is not increased.

Keywords: Information retrieval, similarity, cosine, genetic algorithm

**In* Proceeding of *the 2nd IMTGT_Conference.* Penang, Malaysia. 2006.

# APPLICATION OF GENETIC ALGORITHM TO DETERMINE A DOCUMENT SIMILARITY LEVEL IN IRS*

Poltak Sihombing, Putra Sumari, Abdullah Embong.

## Abstract

The most important problem in **IRS** (***I**nformation **R**etrieval **S**ystem)* is to get the most relevant document from a database. In this paper we propose a technique to determine a document similarity level in IRS using Dice's formulation in Genetic Algorithm (**GA**). The goal of GA is to find a set of documents which best matched the searcher's needs. An evaluation function for the *fitness* of each chromosome was selected based on Dice's score. The Dice's score is a common measure of association in information retrieval. The parent solution (chromosome) with the higher level of fitness has a bigger similarity percentage of documents, while those with lower level of fitness have less similarity percentage of documents. We observed that if the sum of generation is increased, then the sum of crossover and mutation is increased. If the sum of generation is increased, then the percentage of similarity is not increased. This system provides a document ranking of document retrieval according to the similarity level of document retrieval.

Keyword**:** Information retrieval, similarity level, document, Dice, genetic algorithm.

**In* Proceeding of the *1st Malaysian Software Engineering Conference 2005 (MySec'05). Penang, Malaysia. 2005.*

# A TECHNIQUE OF PROBABILITY IN DOCUMENT SIMILARITY COMPARISON IN INFORMATION RETRIEVAL SYSTEM*

Poltak Sihombing, Putra Sumari, Abdullah Embong

## Abstract

Nowadays, storing of documents in an information retrieval system is no longer an issue due to the availability of huge storage space, multiple storage devices and different storage media, and the occurrence of various methods of document storage. The challenge is more on the retrieval of the documents since documents stored in a database grow very fast and soon become unmanageable. In this paper we propose a technique of probability to retrieve a document from one or more databases based on a similarity measure. The similarity measure is calculated by using Jaccard formulation. Jaccard's formulation is used to represent a general measurement of document similarity. We have implemented a prototype of an information retrieval system based in Genetic Algorithm (GA) processing. This algorithm is basically based on natural biological evolution. The parent solution (chromosome) with the higher level of fitness has a bigger probability to reproduce, while those with lower level of fitness have less probability to reproduce. Documents with a higher Jaccard's score reflect a higher probability of similarity. Application of this technique will facilitate searching and retrieval of required document from one or more databases based on the representation of the similarity level.

Keywords: probability technique, similarity level representation, document retrieval, Genetic Algorithm.

* *In Proceeding of the IMT-GT Conference. Parapat, Indonesia. 2005.*

# APPLICATION OF GENETIC ALGORITHM TO DETERMINE A DOCUMENT SIMILARITY LEVEL IN IRS*

Poltak Sihombing, Abdullah Embong, Putra Sumari.

## Abstract

The most important problem in **IRS** (***I**nformation **R**etrieval **S**ystem)* is to get the most relevant document from a database. In this paper we propose a technique to determine a document similarity level in IRS using Genetic Algorithm (**GA**). The goal of GA was to find a set of documents which best matched the searcher's needs. An evaluation function for the *fitness* of each chromosome was selected based on Dice's score. The Dice's score is a common measure of association in information retrieval. To initialize a population, we need first to decide the number of genes for each individual and the total number of chromosomes in the initial population. GA is basically based on natural biological evolution. The parent solution (chromosome) with the higher level of fitness has a bigger similarity percentage of documents, while those with lower level of fitness have less similarity percentage of documents. By the similarity percentage of documents, the user can choose the most relevant document from a database.

Keywords: Information retrieval, similarity, Dice, genetic algorithm

* *In Proceeding of the Malaysian Software Engineering Conference '05 (MySec05). Penang, Malaysia. 2005.*

# IMPLEMENTATION OF GENETIC ALGORITHM IN INFORMATION ACCESS BY JACCARD AND DICE FORMULATION*

Poltak Sihombing

## Abstract

In this paper we are going to implement Jaccard formulation in ***I****nformation* ***R****etrieval* ***S****ystem, (IRS)* and to compare it with the Dice's formulation. In the previous research, we have developed the Cosine and Horng-Yeh's formulation in a prototype called the Journal Browser. Each technique has been implemented in IRS using Genetic Algorithm (GA). The objective of GA was to find a set of documents which best fit the searcher's needs. In this study, we used the formulation of Jaccard and Dice as the *fitness* of each chromosome. This formulation used to measure the relationship of the query with some documents in a database. To initialize a population of the queries, we need first to decide the number of genes for each individual and the total number of chromosomes in the initial population. GA is basically based on natural biological evolution theory. The parent solution (chromosome) with the higher level of fitness has a bigger similarity percentage of documents, while those with lower level of fitness have less similarity percentage of documents. By the similarity percentage of documents, the user can choose the most relevant document from the database. The average percentage of similarity by using 2 queries was found to be 25.00.% in Jaccard, 27.27% in Dice. By using 3 queries, the percentage of similarity was found to be 20.00% in Jaccard, 27.39% in Dice's formulation. .

Keywords: Information, retrieval, document, similarity, genetic algorithm

* *Journal of Information Science and Technology- UHN*. 2007

# A TECHNIQUE OF PROBABILITY IN RANKED DOCUMENT RETRIEVAL SIMILARITY*

Poltak Sihombing

## Abstract

This paper describes a technique to extract a correct keyword by keyword competition in order to find a keyword solution automatically in Genetic Algorithm (GA) processing. An evaluation *fitness* function of each chromosome was selected based on Jaccard, Cosine and Dice's formulation. To initialize a population, we need first to decide the number of genes for each individual and the total number of chromosomes in the initial population. All of keywords represented by chromosomes, and compete in generation population based on their *fitness* value. We use GA in their competition process because of its ability to optimize all of keywords to eliminate unnecessary keywords and left those only legitimate (most important) keywords. Therefore, the keywords solution as the result of keyword competition matches to the document collection in database that hoped potential relevant to user's expectation.. We have implemented this technique in Journal Browser System, called JBS. The system result ranked document retrieval according to the similarity level and user can retrieve them according to this similarity value.

Keywords: Keyword competition, matching function, information retrieval, document similarity, genetic algorithm.

**Jurnal SIFO Mikroskil. ISSN.1412-0100. Vol.04.No.02.* 2004.

# Chapter 1

# INTRODUCTION

In recent years, we have witnessed an immense growth in the availability of document volume in Information Retrieval Systems (IRS). Storing documents in an IRS is no longer an issue due to the availability of huge storage space, multiple storage devices, different storage media, and the occurrence of various methods of document storage. The challenge is more on the retrieval of the right documents since documents stored in a database grow very fast and soon become unmanageable. This situation often results in difficulty to retrieve a document from a database which is expected to be very relevant to users, and this has become one of the most important problems in IRS.

There has been a strong focus placed on digital libraries such as e-journals, e-books, etc as a means of making such on-line information readily and easily available. Given that much of this information is textual in nature, the question arises is how to access so much information can be facilitated that satisfy users needs. Indeed, it becomes necessary to build tool aimed at helping users find those documents that satisfy their needs.

In digital library, users still often find difficulties in getting a relevant document from database (such as research result, journal, paper, and other important documents). Most often documents obtained from digital library were not relevant to their expectation. This matter has resulted not only disappointment for them but also mainly due to the inefficient IRS.

Currently, methods for accessing large text collections (such as journal database) deal with retrieving documents from a database is via a user's keyword. In this paradigm, users are required to specify their information needed in the form of keywords as a query which then matches with a keyword to documents in a database. Unfortunately, some of the methods for accessing information are quickly becoming inadequate. The amount of on-line information is very huge and grows at an unprecedented rate. Methods that respond to a users query with a simple list of documents quickly become unwieldy as the list of matching documents becomes incomprehensibly large and not relevant. Besides, even now, users often find themselves having to wade through several hundred documents returned in response to their queries. This situation will only get worse in the future.

### 1.1 Motivation

In a ranked process in a document retrieval system, a computer system usually gives weight to a document that has a very high similarity to a query. The high weight is considered as high similarity to the query and will be presented to users as a most relevant to them. The collections are typically quite large, on the order of megabytes or gigabytes of text and tens of thousands of documents or more and yet give hurdle in ranking them. One of the most problem issues is: The percentage of similarity value in document retrieval does not accurate. The problem statement in this thesis is how to provide documents to users that better ranked to their information needs. The system does this by estimating the degree of relevance

of each document to the user's keyword. These relevant estimates are used to rank the documents for the user from the most relevant to the least relevant.

The general motivation that we are addressing in this research is how to develop a retrieval system that can generate documents, very relevant to user's expectations. This is a very difficult problem. The main issues that motivates us is presented below.

### 1.1.1 Keyword issues

The first that motivates us is the keyword issue. Keyword is one of the most important elements in IRS, because keyword will determine a potential document to retrieve from database. Given a query in the form of a list of paper or journal document (*n*) in which each contains a list of keyword (*k*). The problems arise in two aspects:

(i) There are *nk* keywords in the query (query means words which are used as a question form) and *nk* can be a large number. These keywords are not unique to each other as they can occur in redundancy. Many documents have similar keywords to each other. Picking the best keywords from *n* document is indeed a crucial task. These picked keywords represent the best to all *n* papers or journals (documents) collection and will be used later in matching process to those keywords in database. The wrong picked keyword results in the un-relevant/wrong document being retrieved from database.

The best keywords that we choose will power our searching optimization and performance of our document retrieval similarity.

Keyword should be the first step in any process that involves optimizing our document retrieval result. We would say that keyword is one of the most important in the IRS process, yet often one that glosses over as either largely unimportant to spend enough time doing effectively.

(ii) Given *k* picked keywords, to match these keywords to those keywords in database indeed consumes a lot of time. Comparing these *k* keywords to every word in the document in database indeed requires very much time. So it is not the best approach.

### 1.1.2 Ranking issues

The second aspect that motivates us is the ranking issue of retrieved documents. Most often there are too many documents being retrieved and yet not properly ranked. The first document in the rank list is supposed to be the most relevant to users. The relevant become less as the document is located at the lower rank of the list.

The approach of ranking documents retrieved based on the most occurring of keyword in document itself is not feasible. This is simply due to the very huge document collections in the database, and to go through every document in database to count the similar words to the keyword requires too much of time.

The rank process issue in general involves two criteria; they are; similarity value and ranking similarity.

**(i) Similarity value**

Similarity value is the term used to give weight of text indicating the most occurring keyword. Sometimes too many documents retrieval is

resulted by the IRS, but there is no similarity value that shows the connection how similar the document and query are. This situation often results in difficulty to choose which document is expected to be relevant to a query.

**(ii) Ranking similarity**

Many information retrieval systems do not provide the ranking similarity of document retrieved. This becomes a serious problem to users, because the users cannot determine which is the most potential of document retrieved.

### 1.1.3 Genetic Algorithm

The third aspect that motivates us is machine learning approach in which Artificial Intelligence (AI) tool is used to solve a problem. AI tool specifically Genetic Algorithm (GA), used in various application has increased in the last decade. GA is an optimization method that has ability to solve a difficult problem. And this seems reliable to be adopted in IRS.

## 1.2 Research objectives

Based upon the motivation previously discussed, the main objectives of this research are as follows:

(i) To develop a method of finding the best final keywords from a given user's queries and keyword of document in database. The final keywords are crucial since they affect the matching process of keywords

to those document collections in database and yet will produce the most relevant document retrieved.

(ii) To develop a keyword based ranking scheme aimed to better rank retrieved document and yet presents most relevant document to users. Two aspects will be focused, namely final keyword matching process and Similarity percentage calculation formulation.

(iii) To asses the performance of the proposed work. Assessing will be done on the following aspects: The similarity percentage of the final keyword and later the performance of ranking scheme to the other method.

## 1.3 Research Scope

The focus of this thesis is the use of the Keyword Competition (KC) approach in ranked document retrieval. The fitness value in KC process is measured by Jaccard's function and Cosine's function. For the data testing, we use the database of "Journal collection proceeding in ***BATAN (Badan Tenaga Atom Nasional)"*** as the benchmark data-set. The proceeding is presented by Computer Science Faculty, University of Indonesia (UI) Jakarta. As a comparison in performance we use the existing Hopfield scheme in Jacard's function and Cosine's function.

## 1.4 Contributions

The research in this thesis gives a number of contributions to the fields of information retrieval. The contributions are:

(i) **Keyword Competition (KC) scheme**

We propose enhancement method of finding the best final keywords called Keyword Competition (KC) scheme to pick the best keywords and eliminates the un-important keywords. *Keywords* are competed to each other through KC process, such as; initial population, fitness evaluation, parent selection, crossover, and mutation. By KC process remains the best potential keyword as the Keyword Solution (KS); KS is the potential keyword in order to find the most potential relevant document retrieval from database. We contributed that the KC approach can be used as one of the new methods in IRS to retrieve some relevant documents in database according to the user's need.

(ii) **Keyword-based ranking scheme**

We also propose enhancement method of ranking process called keyword-based ranking scheme called **POSI** (***P****ercentage* ***O****f* ***SI****milarity*) formulation through keyword matching process. The Keyword Solution (KS) will match to documents collection in database by computing the number of appearance of all keyword solution, in the titles, abstracts and keyword of all document collection. The system will count how many times the appearance of keyword solution and rank document retrieved according

to each keyword solution based on **POSI** (**P**ercentage **O**f **SI**milarity) formulation. We contributed that the **POSI** (**P**ercentage **O**f **SI**milarity) formulation can be used as one of the new methods in ranking document retrieved.

### 1.5 Research methodology

We have observed number of issue in IRS such as keyword issue and ranking issue. The focus of this thesis is enhancement of optimization of Keyword Competition (KC) in ranked document. KC is a competition scheme in finding the best keyword, known as keyword solution (KS), among the available keywords. The keyword solution is then matched to the document collection in the database in order to retrieve the most relevant document. We used the proceeding collection of BADAN TENAGA ATOM NASIONAL (BATAN) Indonesia, presented by University of Indonesia (UI), Jakarta as a standard data-set. The fitness value of KC process was based on Jaccard's function and Cosine's function.

We also develop a keyword based ranking scheme aimed to better rank the retrieved document in the spirit of presenting the most relevant document to the users. Keyword based ranking scheme consists of two (2) main phases; namely Keyword solution matching and percentage of similarity formulation.

Later, we are going to evaluate the performance of our proposed work as presented in figure 7.1. We will compare the KC scheme performance to Hopfield method. To do that, we will develop a prototype of Journal Browser System (JBS) to represent our proposed work.

## 1.6 Organization of the thesis

This thesis is composed of eight chapters. Chapter 1 presents introduction, Chapter 2 presents background, Chapter 3 presents literature survey, Chapter 4 presents Keyword Competition (KC) scheme, Chapter 5 presents keyword based ranking scheme, Chapter 6 presents Journal Browser System, Chapter 7 presents performance evaluation and Chapter 8 presents conclusion and future work. The main chapters are Chapters 4, 5, and 6 as shown in figure 1.1.

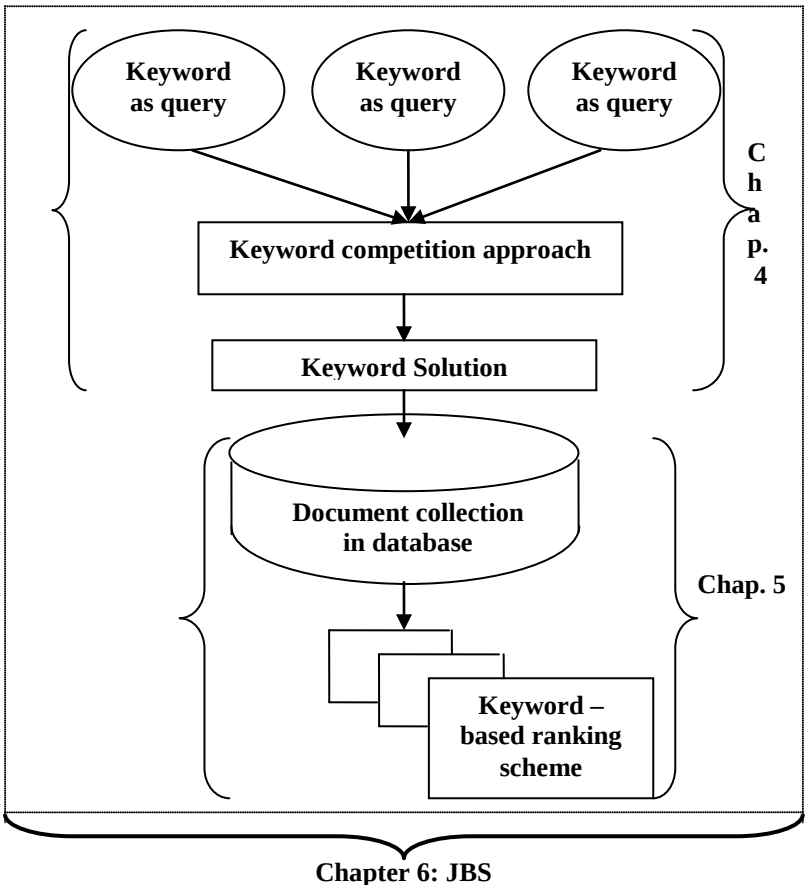


Figure 1.1: Overview of the technical component in JBS

The description of each chapter is described as follows:

**Chapter 2: Background**

In this chapter, we present the background and concept related to the proposal of this thesis. Primary emphasis is on three major fields: IRS, GA and Hopfield method. In IRS section, we outline the basic fundamental of IRS, basic fundamental of GA, and Hopfield method.

**Chapter 3: Literature survey**

In this chapter, we present a summary of literature review pertinent to our work. We report some of the works that use GA and IRS, such as; automatic indexing, document clustering, and query processing. We then outline SONIA, feature selection, NUITS, Hopfield method, and summary of this chapter.

**Chapter 4: Keyword Competition (KC) scheme**

In this chapter, we present the KC scheme as our contribution in this thesis. We present an overview of KC scheme. We then present keyword as the chromosome generation, representation of keyword, evaluation of keyword's fitness, the keyword selection process, crossover of keyword's chromosome, mutation of keyword's chromosome, and recombination of keyword's chromosome. Finally, this chapter ends with the summary of this chapter.

**Chapter 5: Keyword-base ranking scheme.**

In this chapter, we present a ranking process called keyword-based ranking scheme or ***POSI formulation***. We then present document ranking process. Simply, this is a procedure to sort out all documents retrieved. In the next section, we present a document format in database. Finally, this chapter ends with the formulation of document ranking, and summary of this chapter.

**Chapter 6: Journal Browser System (JBS).**

In this chapter, we present the description of **J**ournal **B**rowser **S**ystem (**JBS**). Primary emphasis is on three major fields, namely; a component View of JBS, JBS at work, and JBS in the future. The most significant extension of JBS beyond existing systems, however, is the ability in select the keyword solution as the best keyword. It can be used to get the most relevant document to user's need. This ranking allows users to not only navigate a given document retrieval percentage more easily but also enable to choose them quickly according to the similarity value.

**Chapter 7: Performance Evaluation**

In this chapter, we present a performance evaluation of KC scheme and compare to Hpfield method. We present the processing time, the behavior of number of queries against similarity, and the behavior of GA process on similarity. Finally, this chapter ends with discussion and summary of this chapter.

**Chapter 8: Conclusions and Future Work**

In this chapter, we outline the conclusion and future work of this thesis. This thesis has presented a KC approach and keyword base ranking scheme. We have demonstrated the KC approach by applying it in two different fitness formulations in ranked document retrieved. In all of evaluation, the KC approach is successful. We then present the future work, improvement of information access, and optimization function.

# Chapter 2

# BACKGROUND

The goal of this chapter is to provide a background and definition necessary for the remainder of the work. Two topics are the main focus in this chapter; they are IRS and GA methods. From section 2.1 to 2.5 introduce Information Retrieval and preliminaries for the basis of IRS. Then, section 2.6 presents the fundamental GA method. Finally, section 2.7 presents Hopfield method.

## 2.1 Information retrieval

Information retrieval is a field devoted to the retrieval of useful information from large collections of textual information to fulfill user needs. One of the early definitions of IRS is given by Salton [1]. He defines IRS in a very general way:

> *Information retrieval is a field concerned with the structure, analysis, organization, storage, searching, and retrieval of information.*

A compact definition of the basic function of an IRS has been given by Lancaster[2]:

> "*An information retrieval system does not inform (i.e. change the knowledge of) the user the subject of his enquiry. It merely informs on the existence (or non-existence) and whereabouts of documents relating to his request.*"

Within the few lines of the above definitions, a document based IRS typically consists of three main subsystems: document representation,

representation of query, and the matching process that uses algorithm to match user's query with document representation as shown in figure 2.1. The task of IRS is, through its response, to help user locate those documents that have the potential to satisfy their need. Documents repository stores the document collection in the database. A document collection consists of many documents containing information about various subjects or topics of interests. Document collections are transformed into a document representation based their matching to these queries. Another consideration in document representation is that such a representation should correctly reflect the author's intention. The primary concern in representation is how to select proper index terms. Typical representation proceeds by extracting keywords that are considered as content identifiers and organizing them into this given format.

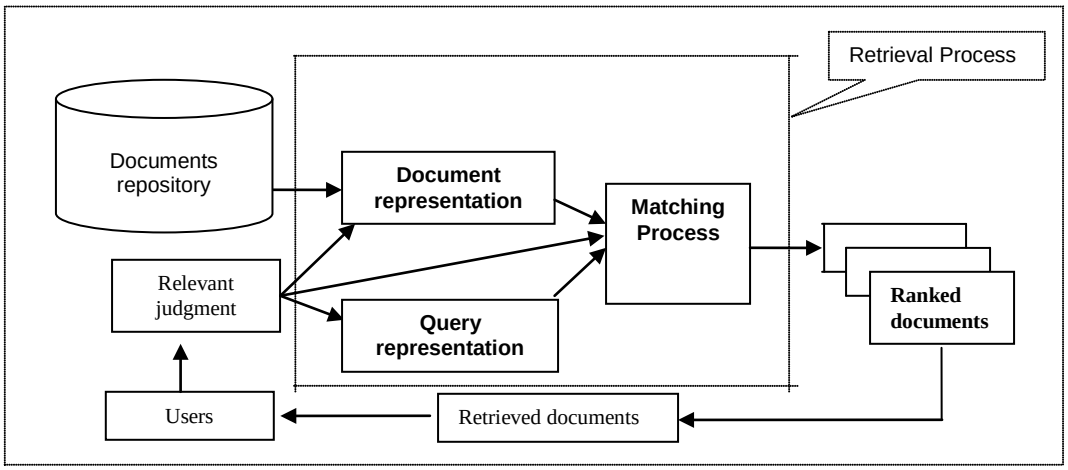


Figure 2.1: A basic architecture of IRS

Query representation formulates their queries (keywords) into the form understood by system. Query formatting depends on the underlying model of

retrieval used (Boolean models [38], vector space models [3, 25], probabilistic models [32.33.106], fuzzy retrieval models [183], and models based on artificial intelligence techniques [10, 68, 24, 56]).

Matching process evaluates the degree of similarity to which the documents in the database satisfy the requirements expressed in the query. A matching algorithm matches a user's request (in term or queries) with the document collection and retrieves documents that are most relevant to the user's need.

Ranked documents indicate the most relevance document presented to users. The rank process uses the similarity levels to rank documents retrieval. The system will sort those documents retrieval according to the biggest percentage value of similarity, where the biggest one is placed on the topmost position.

A set of documents retrieval are processed by IRS in such a way that an internal representation of these documents is derived. This internal representation can be further processed by the IRS. A user who wishes to search this document collection expresses information needed in the form of a query that is posed to the system. The IRS represents this query in an internal form that is suitable for further processing. The retrieval processing matches the user's query against each document in the collection, and produces a list of documents (that is usually ranked in some way) that is presented to the user. The user can interact with this ranked list by indicating documents that relate to his information need. Each of these steps is covered in more details in the remainder of this chapter.

## 2.2 Document representation

This section describes a document representation in IRS. We choose to apply a vector space representation of documents, described in detail in the next section. In this representation, documents are cast as vectors in a very high dimensional space.

### 2.2.1 Initial document processing

Document representation is characterized by a numerical vector [3, 25]. These vectors are embedded in a space in which each dimension corresponds to a term in the corpus of documents being characterized. Figure 2.2 shows the initial document processing and representation phases.

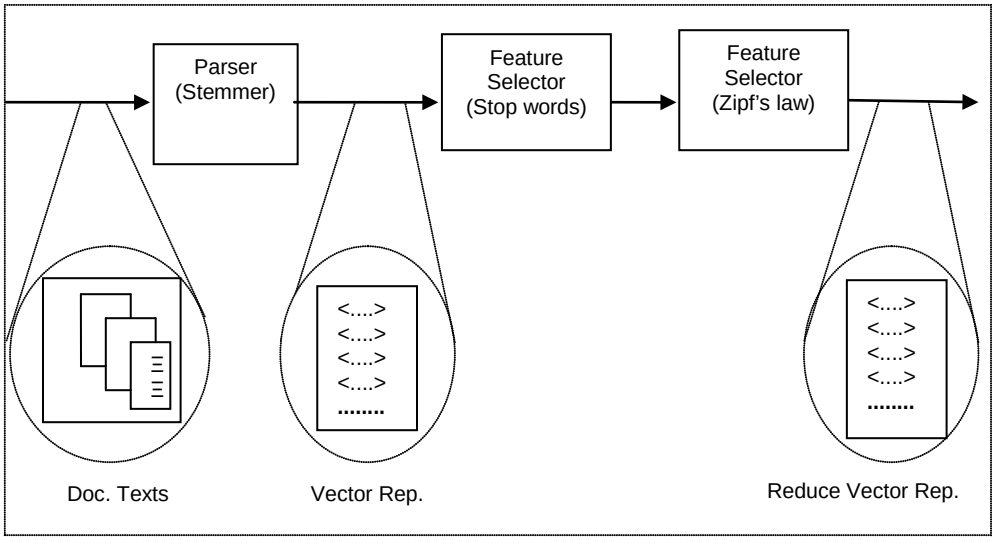


Figure 2.2: Initial document processing and representation phases

A given document vector has in each component a numerical value denoting some function *f* of how often the term corresponding to that dimension appears in

the document collection. By varying the function *f*, we can produce alternative term "*weightings*" [33, 38, 52]. We explore some of standard weighting functions below, and give examples of vector representations of documents using these schemes.

### 2.2.2 Defining "Terms"

The term is a sequence of alpha numeric characters which is delimited by white space (spaces, tabs or newline characters) or punctuation marks (such as a period or a comma). Moreover, all uppercase letters in a document are converted to lowercase. So effectively, capitalization is ignored. To be clearly in defining "terms", let's see the two short sample documents below.

| Computing is not about computers any more. It is about living. |
|---|

Sample document 1

| To live is to compute! |
|---|

Sample document 2

Table 2.1 shows the results of parsing these two documents into single word terms, and then representing them as vectors with simple term frequencies (i.e., term counts) in each component. Such a representation is sometimes also referred to *a bag of words* [42, 45, 52], since the relative position of terms in the document, and hence the language structure, is not captured in the resulting vectors.

Table 2.1: A simple vector representation of the sample documents

| Term | Vector for document 1 | Vector for document 2 |
|---|---|---|
| about | 2 | 0 |
| any | 1 | 0 |
| compute | 0 | 1 |
| computers | 1 | 0 |
| computing | 1 | 0 |
| is | 2 | 1 |
| it | 1 | 0 |
| live | 0 | 1 |
| living | 1 | 0 |
| more | 1 | 0 |
| not | 1 | 0 |
| to | 0 | 2 |

### 2.2.3 Word stemming

In some cases, rather than defining terms to be the distinct words in the corpus, word stemming is used to reduce words to some root form. Thus, the terms that define the dimensions of the vector space are not actual words, but word stems. For example, the words “computer”, “computers”, and “computing”, would all be reduced to the word stem “comput”. Porter [45] has developed a commonly used algorithm for word stemming. By the two short sample documents above, we can see the result of word stemming below:

| Comput i not about comput any more. It i about live. |
|---|

Stemmed version of sample document 1

To live i to compute!.

Stemmed version of sample document 2

From the example above, we can see that the two sample documents presented earlier were stemmed using Porters stemming algorithm. Table 2.2 shows the vector representation of the stemmed version of the documents. While it is clear that, in some cases, stemming may be useful to help conflate similar terms (such as the stem “comput”), in other cases, the results of stemming are counter intuitive (such as stemming “is” to “i”). Frakes [52] provides an overview of studies comparing various stemming methods to unstemmed representations for the retrieval task and shows that in many cases, both representations perform roughly equally.

Table 2.2: A vector representation of the stemmed version

| Stem | Vector for document 1 | Vector for document 2 |
|---|---|---|
| about | 2 | 0 |
| any | 1 | 0 |
| i | 2 | 1 |
| it | 1 | 0 |
| live | 1 | 1 |
| more | 1 | 0 |
| not | 1 | 0 |
| to | 0 | 2 |

### 2.2.4 Stop words

These stop words such as prepositions, conjunctions and pronouns that are used to provide structure in language rather than content. Such words are commonly

used in documents regardless of topic, and thus have no topical specificity. As a result, we can eliminate such words (and the dimensions corresponding to them) from our document vectors as they will be little use when clustering or classifying documents. Such words are commonly referred to a *stop words*, and their elimination from documents is common in IR.

### 2.3 Query representation

Such a formulation of an information need is usually called a query or keyword. Query representation is one of the most important processes, because in this step the system proceeds by extracting keywords to determine a potential keyword (or query). The expectation is that the potential keyword will be matched to a potential document in database. The keyword processing is one of the most important, because if used the wrong keyword, hence the retrieval will result in an irrelevant document. Figure 2.3 shows the query representation phase.

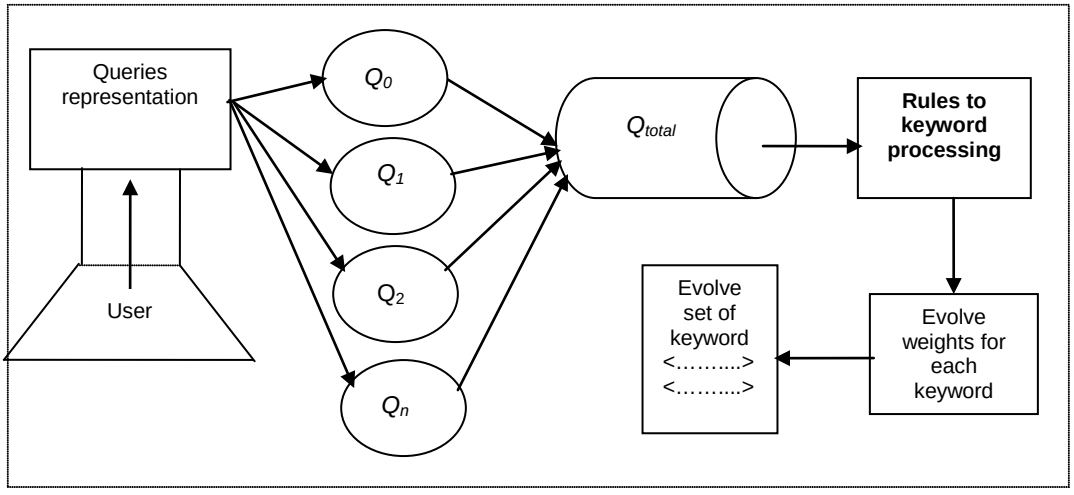


Figure 2.3: Query representation phase

Figure 2.3 shows query representation which is represented by keywords $Q_0$, $Q_1$, $Q_2$,..., $Q_n$, ($n$ is an integer) and set of those keywords as $Q_{total} = \{Q0 \cup Q_1 \cup Q_2 \cup .... Q_n \}$. Query processing then generates a rule to optimize those keywords. This process will evolve weight of each keyword, then put it just one set of maximizing weight keyword. This set of weight keyword is important keywords because these are the most influencing keywords in accessing the most relevant documents in database. The calculation of weight keywords can be used as the base in ranking function similarity of document retrieval [198, 199].

The evolving set of keyword in figure 2.3 can be used to retrieve new documents by combining score of weight keyword based on frequency. Document collections are then ranked by comparing the weight keyword with their summary in the document collection in database.

### 2.3.1 Z-score method

Z-score is the weight formula of keyword $q$ for document $d$ which is formulated by Andrade and Valencia [200] based on the keyword frequencies in document. The weight of keyword $q$ ($q$ = keyword) for document $d$ ($d$ = document) is represented in equation 2.1.

$$Z_{d}^{q} = \frac{F_{d}^{q} - \bar{F}^{q}}{\sigma^{q}} \qquad (2.1)$$

where $F_d^q$ equals the document frequency of keyword *q* in the document *d* and, as defined by Andrade and Valencia [200],

$\bar{F}^q$ equals the average frequency, and $\sigma^q$ is the standard deviation, of keyword *q* in the background set.

### 2.3.2 Term Frequency – Inverse Document Frequency (TF-IDF) scheme

The TFIDF scheme combines term frequency (TF), which measures the number of times a term (or keyword) occurs in the document collection, and inverse document frequency (IDF) [125, 123, 121]. Later, Sparck Jones [33] continues this concept in more accurate quantification and says that keyword (or term) importance can be achieved if one also makes use of information about keyword usage within the entire document collection. The *inverse document frequency (idf)* captures this belief: for a document collection comprising N documents, if keyword i occur in $n_i$ documents, then the keyword's *idf* weight is given by equation 2.2.

$$\text{Log}\left(\frac{N}{n_i}\right) \tag{2.2}$$

A frequently used keyword weighting function is a combination of the *tf* and *idf* weights, typically referred to as a *tf-idf* weight:

$$Wij = \frac{\text{Log}\,(freq_{ij}+1)}{\text{Log}(length_j)} . \text{Log}\left(\frac{N}{n_i}\right) \tag{2.3}$$

$w_{ij}$ = *tf-idf* weight of term *i* in document *j*

$freq_{ij}$ = frequency of term *i* in document *j*
$length_j$ = length (in words) of document *j*
N = number of documents in the collection
$n_i$ = number of documents that term *i* is assigned to

It should be noted that keyword weighting methods have been extensively researched. Salton and Buckley [123], provide a comprehensive overview of various keyword weighting schemes and their comparative effect on retrieval effectiveness.

### 2.3.3 Keyword occurrences scheme

In [38], Luhn postulates that the most discriminating keywords (or terms) are those that occur with medium frequency. High frequency keywords are the most potential for carrying information, and low frequency keywords are rejected.

The keyword representation in documents, various functions may be applied by frequency of keyword occurrences in documents in order to produce "*weighted*" of keyword to document. More formally, let $\xi(t_i, d)$ denote the number of occurrences of keyword (or term) $t_i$ in document *d*. We may then apply some function *f* to $\xi(t_i, d)$ to produce the value for the $i^{th}$ component of the vector for document *d*. For the vectors in Table 2.1, for example, we simply use the identity function $f(\alpha) = \alpha$ applied to the term counts, which was defined by Robertson and Spark Jones [50]. The function $f(\alpha)$ applied to term frequencies is described in equation 2.4.

$$f(\alpha) = \log(\alpha + 1) \qquad (2.4)$$

Some of the formulation of weighting keyword presented above can be used as the base of similarity function process or similarity searching, where a measurement of keyword-document similarity is calculated for each document and for each keyword. Once a keyword has been posed to an IRS, a similar processing to that for documents may take place. A keyword may also be expanded before retrieval is performed, where the expansion process is controlled by the user.

### 2.4 Matching process

The outcome of the matching process is to quantify the likelihood of a specific document to be relevant to the query. The user is subsequently presented with a ranked list of documents, sorted in decreasing order of their relevant scores. It should be noted that in this way, any number of documents may be presented to the user by simply selecting that number of documents from the top of the retrieved list.

#### 2.4.1 Boolean matching model

A Boolean matching process strategy retrieves those documents which are 'true' for the query. This formulation only makes sense if the queries are expressed in terms of index terms (or keywords) and combined by the usual logical connectives AND, OR, and NOT. For example, if the query $Q = (K_1 \text{ AND } K_2) \text{ OR } (K_3 \text{ AND } (\text{NOT } K_4))$. Then the Boolean search will retrieve all documents indexed by $K_1$ and $K_2$, as well as all documents indexed by $K_3$ which are *not* indexed by $K_4$.

Some systems which operate in Boolean matching allow the user's query by the given keywords. An obvious way to implement the Boolean matching is through the inverted file. We store a list for each keyword in the vocabulary, and in each list put the addresses (or numbers) of the documents containing that particular word. To satisfy a query we now perform the set operations, corresponding to the logical connectives, on the *Ki*-lists. For example, if

$K1$ -list : $D1$, $D2$, $D3$, $D4$

$K2$ -list : $D1$, $D2$

$K3$ -list : $D1$, $D2$, $D3$

$K4$ -list : $D1$

and $Q = (K1 \text{ AND } K2) \text{ OR } (K3 \text{ AND } (\text{NOT } K4))$

Then to satisfy the ($K1$ AND $K2$) part we intersect the $K1$ and $K2$ lists, to satisfy the ($K3$ AND (NOT $K4$)) part we subtract the $K4$ list from the $K3$ list. The OR is satisfied by now taking the union of the two sets of documents obtained for the parts. The result is the set {$D1$, $D2$, $D3$} which satisfies the query and each document in it is 'true' for the query.

A slight modification of the full Boolean search is one which only allows AND logic but takes account of the actual *number* of terms the query has in common with a document. This number has become known as the *co-ordination level*. The search strategy is often called *simple matching*. Because at any level we can have more than one document, the documents are said to be *partially* ranked by the co-ordination levels.