Al al-Bayt University

Prince Hussein bin Abdullah College of Information Technology

Computer Science Department

Improving the Effectivness of Information Retrieval System Under Vector Space Model Using Adaptive Genetic Algorithm

By

Wafaa Zaal Mohammad Maitah

2010

Improving the Effectiveness of Information Retrieval System Under Vector Space Model Using Adaptive Genetic Algorithm

By

Wafaa Zaal Mohammad Maitah

Supervisor: Dr. Mamoun Al-Rababaa

Co. Supervisor: Prof. Ghassan Kanaan

A Thesis Submitted to the
Scientific Research and Graduate Faculty in partial fulfillment of the
Requirements for the Degree of Master of Science
in Computer Science

Members of the Committee	Approved
Dr. Mamoun Al-Rababaa	…………..
Prof. Ghassan Kanaan	....
Prof. Adnan Al-Smadi	....
Dr. Jehad Al-Alnihoud	……………
Prof. Riyad Al- Shalabi	……………

Al al-Bayt University
Mafraq, Jordan
2010

Dedication

This thesis is dedicated to everyone who gave me love, friendship and support during my research.

Acknowledgements

All praise and glory to Allah who alone made this research to be accomplished .My deep appreciation goes to my thesis supervisor Dr.Mamoun Rababaa for his constant help, guidance, and the countless hours of attention he devoted throughout this research.

My special thanks and sincere gratitude are due to my co supervisor Prof.Ghassan Kanaan, his priceless suggestion made this research more learning for me.

I wish to express my heartfelt gratitude to my parents for their encouragement, constant prayers. I owe a lot of thanks to my sisters, brothers for their support.

Table of Contents

List of Tables

List of Figures

List of Appendices

List of Abbreviations

Abbreviation	Meaning
IR	Information Retrieval
GA	Genetic Algorithm
AGA	Adaptive Genetic Algorithm
VSM	Vector Space Model
LM	Language Model
EBM	Extended Boolean Model
BM	Boolean Model
P_m	Mutation probability
P_c	Crossover probability

Abstract

Traditional Genetic Algorithm which is used in previous studies depends on fixed control parameters especially crossover and mutation probabilities, but in this research we tried to use adaptive genetic algorithm.

Genetic algorithm started to be applied in information retrieval system in order to optimize the query by genetic algorithm, a good query is a set of terms that express accurately the information need while being usable within collection corpus, the last part of this specification is critical for the matching process to be efficient, that is why most research efforts are actually put toward the query improvement.

We investigated the use of adaptive genetic algorithm (AGA) under vector space model, Extended Boolean model, and Language model in information retrieval (IR), the algorithm used crossover and mutation operators with variable probability, where a traditional genetic algorithm (GA) uses fixed values of those, and remain unchanged during execution. GA is developed to support adaptive adjustment of mutation and crossover probability; this allows faster attainment of better solutions. The research has been tested using 242 Arabic abstracts collected from the proceedings of the Saudi Arabian National conference.

Keywords: Information Retrieval, Adaptive Genetic Algorithm, Vector Space Model, Language Model, Extended Boolean Model.

Chapter One

Introduction

Information retrieval (IR) deals with the representation, storage, organization, and access to information items [1]. One central problem of IR is the issue of determining which documents are relevant and which are not to the user information need. In practice, this problem is usually regarded as a ranking problem, whose goal is to define, according to the degree of relevance (or similarity) between each document and the user query, an ordering among documents so as to rank relevant documents in higher positions of the retrieved list than irrelevant ones[1], [2].

Information retrieval is concerned with collection and organization of texts, responding to the requests of internet users for the information seeking text, retrieving the most relevant documents from a collection of documents; and with retrieving some of non-relevant as possible. Information retrieval is involved in:

- Representation,
- Storage,
- Searching,
- Finding documents or texts or images those are relevant to some requirements for information desired by a user [1], [3].

Genetic algorithm started to be applied in information retrieval system in order to optimize the query by genetic algorithm, a good query is a set of terms that express accurately the information need while being usable within collection corpus, the last part of this specification is critical for the matching process to be efficient, that is why most research efforts are actually put toward the query improvement.

Traditional Genetic Algorithm which is used in previous studies [4], [5] depends on fixed control parameters especially crossover and mutation probabilities, but in this research we tried to use adaptive genetic algorithm in other words it depends on variable crossover and mutation probabilities so as to improve performance in an information retrieval.

1.1Scope of the Research

In this research we constrain our attention to three highly popular and successful families of models: Vector space, extended Boolean and language models, and in this research we constrain our attention to the use of Adaptive Genetic Algorithm (AGA) that use crossover and mutation operators with variable probability, the dataset used in this thesis is the 242 Arabic abstracts collected from the proceeding of the Saudi Arabian National Conference [6].

1.2. Aims and Objectives

In this research we:

1. Attempt to enhance the performance of information retrieval by using adaptive genetic algorithm which can improve the quality of query, and obtain more developed queries that fit the searcher's needs.
2. Investigate and evaluate different fitness such as the Cosine, and Horng and Yeh's under Vector Space Model, Extended Boolean Model, and Language Model.
3. Reduce the search space which leads to saving time and reduction the number of iterations needed to generate the most optimized query.
4. Obtain the best techniques to modify the query in an information retrieval system.

1.3 Thesis Outline

This chapter provides a brief introduction to our research and explains the objective of the research.

The out line of the remaining chapters of the thesis will be as follows:

Chapter 2: Presents an overview of information retrieval

Chapter 3: Presents an overview of Genetic Algorithm

Chapter 4: Describes previous related work.

Chapter5: Describes the methodology that has been used.

Chapter6: Describes the proposed algorithm.

Chapter7: Experimental results of the proposed algorithm.

Chapter8: Presents conclusions and future work.

Chapter Two

Information Retrieval

2.1 Information Retrieval Definition

The discipline of information retrieval (IR) is almost as old as the computer itself Information retrieval is the name of the process or method whereby a prospective user of information is able to convert his need for information into an actual list of citations to documents in storage containing information useful to user [1].

An information retrieval system is software programmed that stores and manages information on documents. The system assists users in finding the information they need. Unlike so-called question answering systems [3], the system does not explicitly return information or answer questions. Instead, it informs on the existence and location of documents that might contain the needed information. Some suggested documents will, hopefully, satisfy the user's information need. These documents are called relevant documents. A perfect retrieval system would retrieve only the relevant documents and no irrelevant document [1], [4].

2.1.1 Basic Processes of Information Retrieval

There are three basic processes an information retrieval system has to support: the representation of the content of the documents, the representation of the user's information need, and the comparison of the two representations. The processes are visualized in figure 2.1.

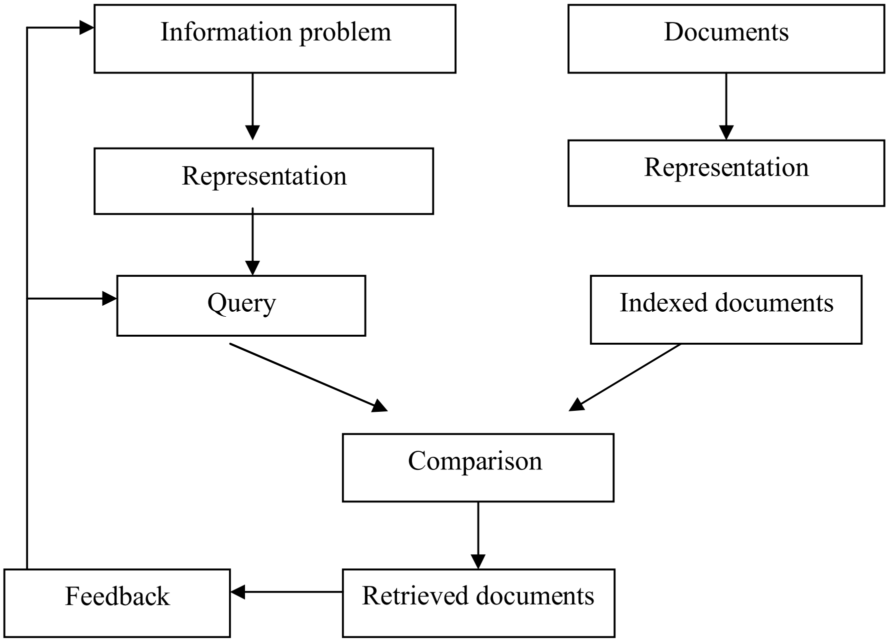

Figure 2.1: Information Retrieval Processes [1]

Representing the documents is usually called the indexing process. The process takes place off-line, that is, the end user of the information retrieval system is not directly involved. The indexing process results in a formal representation of the document: the index representation or document representation. Often, full text retrieval systems use a rather trivial algorithm to derive the index representations, for instance an algorithm that identifies words in an English text and puts them to lower case [1], [2]. The indexing process may include the actual storage of the document in the system, but often documents are only stored partly, for instance only title and abstract, plus information about the actual location of the document. The process of representing the information problem or need is often referred to as the query formulation process. The resulting formal representation is the query. In a broad sense, query formulation might denote the complete interactive dialogue between system and user, leading not only to a suitable query but possibly also to a better understanding by the user of his/her information need [3].

2.2 Text Information Retrieval Models

A range of different models have been proposed in the information retrieval literature, based upon different notions of what it means for a document to be relevant to a query. While some models, such as the Boolean model, have been important historically, the most common form of information retrieval today is ranked retrieval. A query is treated as an unordered set of keywords (also known as a "bag of words" query) [1]. Using statistics about how terms are distributed in documents and across the collection as a whole, the IR system calculates a similarity measure between the query and each document, and returns a list of documents ordered by decreasing similarity score to the user. Different models calculate the similarity between queries and documents in different ways. In this research we constrain our attention to three highly popular and successful families of models: Vector space, Extended Boolean and language models [2].

2.2.1Vector Space Model

In this model, a document is viewed as a vector in n-dimensional document space (where n is the number of distinguishing terms used to describe contents of the documents in a collection) and each term represents one dimension in the document space [1], [7]. A query is also treated in the same way and constructed from the terms and weights provided in the user request. Document retrieval is based on the measurement of the similarity between the query and the documents by using different similarity measures (as shown in Table 2.1). This means that documents with a higher similarity to the query are judged to be more relevant to it and should be retrieved by the IRS in a higher position in the list of retrieved documents. In This method, the retrieved documents can be orderly presented to the user with respect to their relevance to the query [8].

By letting ***N*** be the number of documents and ***t*** be the number of index terms, the documents and the query are represented as the following [8]:

$$\vec{q} = (w_{1,q}, w_{2,q}, \ldots, w_{t,q})$$

$$\vec{d}_j = (w_{1,j}, w_{2,j}, \ldots, w_{t,j})$$

Table 2.1: Different Similarity Measures

Similarity Measure	Evaluation for Binary Term Vector	Evaluation for Weighted Term Vector
Cosine	$sim(d,q) = 2\frac{\lvert d \cap q \rvert}{\lvert d \rvert^{1/2} \bullet \lvert q \rvert^{1/2}}$	$sim(d_j, q) = \frac{\sum_{i=1}^{t} w_{i,j} \times w_{i,q}}{\sqrt{\sum_{i=1}^{t} w_{i.j}^2} \times \sqrt{\sum_{j=1}^{t} w_{i,q}^2}}$
Dice	$sim(d,q) = 2\frac{\lvert d \cap q \rvert}{\lvert d \rvert + \lvert q \rvert}$	$sim(d_j, q) = \frac{2\sum_{i=1}^{t} w_{i,j} \times w_{i,q}}{\sum_{i=1}^{t} w_{i,j}^2 + \sum_{i=1}^{t} w_{i,q}^2}$
Jaccard	$sim(d,q) = \frac{\lvert d \cap q \rvert}{\lvert d \rvert + \lvert q \rvert - \lvert d \cap q \rvert}$	$sim(d_j, q) = \frac{\sum_{i=1}^{t} w_{i,j} \times w_{i,q}}{\sum_{i=1}^{t} w_{i,j}^2 + \sum_{i=1}^{t} w_{i,q}^2 - \sum_{i=1}^{t} w_{i,j} \times w_{i,q}}$

2.2.2 Boolean model

In the Boolean retrieval model, the indexer module performs a binary indexing in the sense that a term in a document representation is either significant (appears at least once in it) or not. User queries in this model are expressed using a query language that is based on these terms and allows combinations of simple user requirements with the logical operators AND, OR and NOT. The result obtained from the processing of a query is a set of documents that totally match with it, i.e., only two possibilities are considered for each document: to be or not to be relevant for the user’s needs, represented by the user query [9].

2.2.3 Extended Boolean model

The main problem of Boolean model is that no ranking is possible. Query for "term1 & term2 & term3" will not differentiate a document with no hits and a document with two out of three hits, discarding both. In the same way, "term1 OR term2 OR term3" will make no difference between documents with only one hit and one with all three hits, and retrieve both. This problem could be partially addressed by weighting the strength of the Boolean connectives and ranking the documents retrieved according to those weights. This is calling the Extended Boolean Model [8].

Extend the Boolean model with the functionality of partial matching and term weighting its Combination of Boolean and Vector models ,in comparison with Boolean model, adds "distance from query" that mean some documents satisfy the query better than others ,in comparison with Vector model, adds the distinction between AND and OR combinations[8], [9].

The idea of the extended Boolean model [1], [10] is based on a critique of a basic assumption in Boolean algebra.

Let

q = kx $\wedge$ ky, use weights associated with kx and ky

In boolean model:

wx = wy = 1; all other documents are irrelevant

In extended Boolean model:

If the query is q=kx $\vee$ ky (conjunctive query)

- The docs near the point (1, 1) are preferred
- The similarity measure is defined in equation1 [11]

$$\text{sim(qand,dj)} = 1 - \text{sqrt(} \frac{(1\text{-}x)^2 + (1\text{-}y)}{2} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(1)$$

If the query is q=kx ^ ky (disjunctive query)

- The docs far from the point (0, 0) are preferred
- The similarity measure is defined in equation2 [11]

$$\text{sim (qor,dj)} = \text{sqrt} \frac{\left(x^2 + y^2\right)}{2} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (2)$$

2.2.4 Language Model

Statistical language models estimate the distribution of words in an input language. In the context of information retrieval, a document is generally viewed as a sample from an underlying language model. That is, the document is only one possible version of the information that is being conveyed by the author; terms in the collection are generated with specific probabilities. Documents are ranked by the likelihood that each document language model could have generated the user's query terms. Many variations on the language modeling approach to information retrieval have been proposed, including multiple Bernoulli models [1], multinomial models [12], and relevance models [13]. Despite differences in the model implementations, the underlying process can be broadly viewed as consisting of three main steps: first, a language model is estimated for each document in the collection; second, the system calculates the probability that we would observe the sequence of query terms if we sampled terms at random from each document language model; and, finally, the documents are ranked in order of these probabilities.

Under the query-likelihood approach, language models for IR try to estimate for each document the probability that the query Q was generated by the underlying language model, MD. If it is assumed that terms occur independently, then the probability becomes the product of the individual query terms given the document model [11], [13].

In information retrieval, it is common to use unigram models, where terms do not depend on their context. (While more sophisticated models could be expected to improve performance, work using higher order models has not been able to demonstrate consistent gains for IR; while such models are much more complex to estimate [14].It therefore remains to estimate the probability of individual query terms. The document under consideration, D, is a sample from the language model.

We note here that if a query term does not occur in the document, then the maximum likelihood estimate for that term is zero, giving an overall similarity score of zero for the query and the document. However, it is not sensible to rule out a document just because a single query term is missing. Therefore language models make use of smoothing to balance probability mass between occurrences of terms in documents, and those terms not found in the documents.

Smoothing is an important feature of language models: it balances term probabilities by discounting the probability of terms seen in the document, and adjusting low or zero probabilities upwards for other terms. This circumvents the zero frequency problems, where query terms that do not occur in a document would otherwise lead to an overall query likelihood of zero. Typically, smoothing approaches combine document term frequencies with the frequency of the term in the collection as a whole [14], [15].

The idea of the language model can perform as the following steps:

a) A probabilistic mechanism for generating a query-“query likelihood” retrieval:

- Build a language model M for each document D in collection (view each doc as a LM); then
- Rank documents based on their likelihood of generating the query

$$SC\left(Q, D_i\right) = P\left(Q|M_{Di}\right) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (3)$$

$P\ (Q|M_D)$: Probability of producing a query

b) A common approach to calculate $P\ (Q|M_{Di})$:

- Consider query terms as independent terms, and
- Calculate $P\ (Q|M_{Di})$ as the product of probabilities, for both the terms present in the query and absent

$$SC\left(Q, D_i\right) = \prod_{t_j \in Q} (t_j |\ M_{Di}) \prod_{t_j \notin Q} \left(1\text{-}p\left(t_j |\ M_{Di}\right)\right) \ldots\ldots\ldots\ldots\ldots\ldots\ldots (4)$$

c) Maximum likelihood estimate of the term distribution (relative term frequency):

$$P\left(t_j|M_{Di}\right) = p_{ml}\left(t_j|M_{Di}\right) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (5)$$

$$P_{ml}\left(t_j|M_{Di}\right) = \frac{tf\left(t_j, D_i\right)}{dl_{Di}} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (6)$$

$p_{ml}\ (t|M_{Di})$: Maximum likelihood estimate of the probability of term t tunder term distribution of document d .

d) Estimate non-zero values for the absent terms. Thus, if tf (tj, Di) = 0 then:

$$P\left(t_j |\ M_{Di}\right) = \frac{Cf_t}{Cs} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (7)$$

Cf_t: number of occurrences of term *t*in collection

cs: number of terms in collection

e) Smoothing-Minimize Risk

Minimize risk in estimation of probabilities to improve effectiveness:

$$R_{t,d}=\left[\frac{1.0}{1.0+f'}\right]*\left[\frac{f'}{1.0+f'}\right]^{tf_{t,d}} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (8)$$

$R_{t,d}$: Risk for term t in a document d .

$$f' = p_{avg}(t)\times d_{ld} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (9)$$

if tf (t_j, Di) > 0:

p_{avg} : Mean probability of term *t* in documents which contain *t*.

d_{ld} : Total number of terms in documents.

$$p\left(t_j|M_{Di}\right)=p_{ml}\left(t_j|M_{Di}\right)^{(1\text{-}R_{t,d})}\times p_{avg}(t)^{R_{t,d}} \ldots\ldots\ldots\ldots\ldots\ldots\ldots (10)$$

if tf (t_j, D_i) = 0 wee implement Eq (7)

$$P\left(t_j|\,M_{Di}\right)=\frac{Cf_t}{Cs}.$$

And finally Eq (4):

$$SC\left(Q,Di\right)=\prod_{tj\in Q}(t_j|M_{Di})\prod_{tj\notin Q}\left(1\text{-}p\left(t_j|M_{Di}\right)\right)$$

2.3 Performance Measures

Any information retrieval System usually evaluated through efficiency and effectiveness of this system. Moreover, there are two aspects of efficiency, those are: Time and Space. Time is the speed of user's compatibility with Document Descriptions, while space is the needed area of Disk needed by the system. For efficiency we can consider it as the system's ability to recall relevant Documents for the user's recall. As we can say the perfect case of a system is its ability to recall any relevant documents and eliminating the recall of irrelevant ones but there are some difficulties such as determination of Relevance as any determination of any document is a subjective process since person's decision depends on many factors such as experience as any expert might consider any recalled general data maybe irrelevant while rocky one considers it the most relevant Data. In research fields this process maybe seen as an objective process [16], [[17]. The standards used in evaluating the performance of a system are Precision, Recall, average Recall and Precision and Fallout.

Precision: means the system's ability to retrieve documents related to query that can be written according to the following mathematical equation [1], [18]:

$$\text{Precision} = \frac{\text{Number of relevent documents retrieved}}{\text{total number of documents retrieved}}$$

Recall: mean the system's ability to retrieve all related documents of a query. According to the following mathematical equation [1], [18]:

$$\text{Recall} = \frac{\text{Number of relevent documents retrieved}}{\text{total number of relevent documents}}$$

Mean Average Precision: Find the average precision for each query and compute the mean AP over all queries average recall-precision take the mean of the precision at each recall point for all queries together and take average at standard recall points, the following mathematical equation [19]:

$$\overline{P}(r) = \frac{1}{N_q} \sum_{i=1}^{N_q} P_i(r) \ldots\ldots\ldots\ldots\ldots\ldots (11)$$

Where:

$\overline{P}(r)$ = average p at the recall level r

N_q = number of queries that used

P_i = precision p at recall r for the i-th query

Fallout: The proportion of non-relevant documents that are retrieved, out of all non-relevant documents available, the following mathematical equation [18]:

$$\text{Fallout} = \frac{\text{total number of documents retrieved}}{\text{Number of relevent documents retrieved}}$$

Chapter Three

Genetic Algorithm

3.1 Introduction

In nature, individuals that are best suited to the environment will win the competition for scanty resources. An individual's survival capacity is determined by various features that characterize it. The features in turn are determined by the individual's genetic content.

Since only the fittest individuals survive and reproduce, the genes of weaker individuals die out gradually. If the environment doesn't change during the process, we can imagine that finally it will converge to a state that every individual has the fittest (or the best) genes [20].

Inspired by this natural evolution process, the use of analogies of natural behavior led to the development of Genetic Algorithms (GAs). GA has 4 main elements [21]: an encoding element that will be replicated, operators to affect the individuals of a population, a fitness function that indicates how good an individual is, and a selection mechanism. Each individual of the population represents a possible solution to a given problem. Each individual is assigned a fitness score based on the fitness function. A selection mechanism selects highly fit individuals to reproduce the offspring by crossover and mutation techniques [22].

GA is not guaranteed to reach the global optimum, but it is generally good for finding an acceptable solution during an acceptable amount of time. It is mainly design to solve optimization problems. It is so robust that it can be applied to a wide range of problem areas. It also has good performance when solving some difficult problems with no existing specialized techniques can perform well [20], [22].

3.2 How Genetic Algorithms Work

Based on natural phenomenon called “the survival of the fittest”, only the fittest individuals survive and reproduce. The reproduction process happens in the gene pool. New combinations of genes are generated from previous ones by exchanging segments of genetic material among chromosomes (known as “crossover”) [20]. Then a new gene pool is created. Repeated selection and crossover cause the continuous evolution of the gene pool and the generation of individuals that survive better in a competitive environment [23].

GA operates on encoded representations of the solutions, equivalent to those chromosomes of the individuals in nature. It is assumed that a potential solution to a problem may be represented as set of parameters and encoded as a chromosome [23].

A fitness function must be provided for evaluating each string. Each solution is associated with a fitness value, based on the fitness function, to reflect how good it is.

Selection models nature` s survival of the fittest mechanism. In principle, individuals from the population are copied to a "mating pool", with highly fit individuals being more likely to receive more than one copy, and unfit individuals being more likely to receive no copies [24].

The reproduction phase of GA is simulated through a crossover mechanism. The simplest method of crossover is to cut the chromosomes of the two individuals of some randomly chosen position and then exchange their "head" and "tail" segments, known as 1-point crossover [23], [25].

Another operation, called mutation, causes sporadic and random alteration of the bits of strings, which is a direct analogy from the nature and plays the role of regenerating lost genetic materials.

3.3 Components of GA

1. Representation: GA was derived from a study of biological systems. In biological systems evolution takes place on chromosomes-organic devices for encoding the structure of living things. A living being is only a decoded structure of the chromosomes. Natural selection is the link between chromosomes and the performance of the decoded structures. In GA, the design variables or features that characterize an individual are represented in ordered list called string. Each design variable corresponds to gene and the string of the design variables corresponds to chromosomes in biological systems [25], [26].

2. Initialization: GA operates with a set of strings instead of a single string. This set or population of strings goes through the process of evolution to produce new individual strings. To begin with, the initial population could be seeded with heuristically chosen

strings or at random. In either case, the initial population should contain a wide variety of structures [26].

3. Evaluation Function**:** The evaluation function is a procedure to determine the fitness of each string in the population and is very much application oriented. Since GA proceeds in the direction of evolving better fit strings and the fitness values is the only information available to GA, the performance of the algorithm is highly sensitive to the fitness values. In case of optimization routines, the fitness is the value of the objective function to be optimized [23], [27]. GA is basically unconstrained search procedures in the given problem domain. Any constraints associated with the problem could be incorporated into the objective function as penalty function [27].

4. Parent Selection Techniques: When breeding new chromosomes, we need to decide which chromosomes to use as parents. The selected parents must be the fittest individuals from the population but we also want sometimes to select less fit individuals so that more of the search space is explored and to increase the chance of producing promising offspring [27]. The disadvantage of always using, the top few chromosomes is that the population quickly converges to one of these individuals and produces sub-optimal solution known as premature convergence [24]. Two common selection techniques are listed below [28]:

1. Roulette Wheel Selection. The idea behind the roulette wheel selection parent selection technique is that each individual is given a chance to become a parent in proportion to its fitness evaluation [23]. It is called roulette wheel selection as the chances of selecting a parent can be seen as spinning a roulette wheel with the size of the slot for each parent being proportional to its fitness. Obviously those with the largest fitness (and slot sizes) have more chance of being chosen.

The problem with roulette wheel selection is that one member can dominate all the others and get selected a high proportion of times. The reverse is also true. If the evaluation of all the members is very close to one another then they will have an almost equal chance of being selected [28], [20].

2. Tournament Selection: In tournament selection, potential parents are selected and a tournament is held to decide which of the individuals will be the parent.

Using the evaluation to choose parents can lead to problems. For example, if one individual has an evaluation that is higher than all the other members of the population then that chromosome will get chosen a lot and will dominate the population. Similarly, if the population has almost identical evaluations then they have an almost equal chance of being selected, which will lead to an almost random search [21], [22].

In order to solve this problem, each chromosome is sometimes given two values, an evaluation and fitness. The fitness is a normalized evaluation so that parent selection is done more fairly. Some of the methods for calculating fitness are described below [25], [23].

1. Windowing. The windowing evaluation technique takes the lowest evaluation and assigns each chromosome a fitness equal to the amount it exceeds this minimum.

2. Fitness ranking, individuals are sorted in order of raw fitness, and then new fitness values are assigned according to rank. This may be done either linearly or exponentially. Fitness ranking can cease over-compression problem.

3.4 Genetic Operators

1. Crossover**:** Simple Genetic Algorithm (SGA) uses 1-point crossover, where mating chromosomes are cut once. Other crossover techniques have also been devised, often involving more than one cut point [22]. In 2-point crossover, chromosomes are regarded as loops by connecting the ends together. Two cut points decide a segment, and two chromosomes exchange the segment. It performs the same task as 1-point cross over, but more general [24].

- One-point crossover: Uniform crossover involves an average of L/2 crossover points for strings of length L. In uniform crossover, each gene in the offspring is created by copying the corresponding gene from either parent according to a randomly generated crossover mask [26].
- Two-point crossover: Randomly two positions in the chromosomes are chosen. Avoids that genes at the head and genes at the tail of a chromosome are always split when recombined [25]

2. Mutation: Mutation is the process of random modification of the value of a string with small probability. It is not a primary operator but it ensures that the probability of searching any region in the problem space is never zero and prevents complete loss of genetic material through reproduction and crossover [23].

3.5 Genetic Parameters

1. Population size: Population size affects the efficiency of the algorithm. If we have smaller population, it would only cover a small search space and may results in poor performance. A larger population would cover more space and prevent premature convergence to local solutions. At the same time, a large population needs more evaluation per generations and may slow down the convergence rate [28].

2. Probability of Crossover: Probability of crossover or crossover rate is the parameter that affects the rate at which the crossover operator is applied. A higher crossover rate introduces new strings more quickly into the population [28]. For uniform crossover, a higher probability of contributing ones parent's allele lowers the rate of disruption. If the crossover rate is too high, high performance strings are eliminated faster that selection can produce improvements. A low crossover rate may cause stagnation due to the lower exploration rate [28], [29].

3. Probability of Mutation: Probability of mutation or mutation rate is the probability with which each bit position of each string in the new population undergoes a random change after a selection process. A low mutation rate helps to prevent any bit positions from getting stuck to single values, where as a high mutation rate results in essentially random search [23], [27].

3.6 Using GA for IRS

In this 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 genetic algorithm was executed in IR as the following steps:

a. Encoding of a Chromosome: Algorithm begins with a set of solutions (represented by chromosomes) called population. Solutions from one population are taken and used to form a new population. This is motivated by the expectation that the new population will be better than the old one. Solutions which are then selected to form new solutions (offspring) are selected according to their fitness [30].

In IRS the keywords used in the set of user-selected documents were first identified to represent the underlying bit strings for the initial population. A chromosome is formed by gene which represents bit (0 and 1). Each bit represents the same unique keyword throughout the complete GA process. When a keyword is present in a document, the bit is set to 1; otherwise it is set to 0. Each document could then be represented in terms of a sequence of 0s and 1s which is called a chromosome model. At the beginning of a run of a GA a large population of random chromosomes is created. Chromosome models depend on the case

b. Crossover: This is simply the chance that two chromosomes will swap their bits. Crossover is performed by selecting a random gene along the length of the chromosomes and swapping all the genes after that point [23].

E.g. Given two chromosomes

A: 10001001110010010

B: 010100010**01000011**

Choose a random bit along the length, say at position 9, and swap all the bits after that point, so the above become:

A’: 100010011**01000011**

B’: 01010001010010010

c. Mutation: After a crossover is performed, mutation takes place. Mutation is intended to prevent falling of all solutions in the population into a local optimum of the solved problem. Mutation operation randomly changes the offspring resulted from crossover. In case of binary encoding we can switch a few randomly chosen bits from 1 to 0 or from 0 to 1. Mutation can be illustrated as follows [31]:

Original offspring 1	110**1**111000011110
Original offspring 2	110110**0**1001101**1**0
Mutated offspring 1	110**0**111000011110
Mutated offspring 2	110110**1**1001101**1**0

The technique of mutation (as well as crossover) depends mainly on the encoding of chromosomes. For example when we are encoding permutations, mutation could be performed as an exchange of two genes [26], [30].

d. Determination of Population: Determination of population on next generation is based on the fitness' score. The higher level of fitness has a bigger probability to reproduce, while those with lower level of fitness have less probability to reproduce. Usually, this probability is selected by Roulette wheel selection method. In the next generation, all population is evaluated to determine whether they have reached the expected solution [23], [30].

Chapter Four
Literature Survey

There are several studies use GA in information retrieval system:

1. Feras AL-Mashakbeh:

Different GA strategies were used in this study; those strategies are as the following [32]:

GA1: GA that used one-point Crossover and point mutation

GA2: GA that used one point crossover operator and chromosomal mutation

GA3: GA that used restricted Crossover operator and point mutation

GA4: GA that used restricted Crossover operator and chromosomal mutation

GA5: GA that used uniform Crossover operator and point mutation

GA6: GA that used uniform Crossover operator and chromosomal mutation

GA7: GA that used fusion operator and point mutation

GA8: GA that used fusion operator and chromosomal mutation

GA9: GA that used dissociated operator and point mutation

GA10: GA that used dissociated operator and chromosomal mutation

In cosine similarity the study compared different GA approaches by calculating the improvement of each approach over the traditional IR system. We noticed that most approaches (GA1, GA2, GA4, GA5, GA8, GA9, and GA10) gave higher improvement than traditional IR system and that the GA strategy that used one-point Crossover operator and point mutation gave the highest improvement over the traditional with 12.4245% [32].

In Jaccard similarity the study compared different GA approaches by calculate the improvement of each approaches over the traditional IR system. We noticed that most approaches (GA1, GA2, GA4, GA5, GA8, GA9, and GA10) give a high improvement than traditional IR system and that the GA strategy that used one point crossover operator and chromosomal mutation gave the highest improvement over the traditional with 12.476% [32].

In Dice similarity the study compared different GA approaches by calculating the improvement of each approach over the traditional IR system. We noticed that most approaches (GA1, GA2, GA4, GA5, GA8, GA9, and GA10) gave a higher improvement than traditional IR system and that the GA strategy that used dissociated operator and point mutation gave the highest improvement over the traditional with 10.833% [32]

In Inner Product similarity the study compared different GA approaches by calculating the improvement of each approach over the traditional IR system. We noticed that most approaches (GA1, GA2, GA4, GA5, GA8, GA9, and GA10) gave a higher improvement than traditional IR system and that the GA strategy that used gave one-point Crossover and point mutation the highest improvement over the traditional with 11.944% [32].

The study also applied with different mutation strategies and different fitness function (Recall, Precision) on Boolean model, we noticed that GA with point mutation gave a higher improvement than traditional IR system [32].

The study also applied with different mutation strategies and different fitness function (Recall, Precision) on Fuzzy Set, we noticed that GA with point mutation gave a higher improvement than traditional IR system [32].

2. Poltak Sihombing,, Abdullah Embong,, Putra Sumari:

In this paper Horng & Yeh's formulation in Information Retrieval System, (IRS) is imlemented and compared with the Jaccard's formulation and Dice's formulation. In the previous research Jaccard and Dice's formulation is developed in a prototype called the Journal Browser [33].

Each technique had 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,first the number of genes for each individual and the total number of chromosomes (popsize) in the initial population have to be decided. A was basically based on natural biological evolution theory [33].

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 [33].

3. Suhail S. J. Owais, Pavel Kr¨omer, and V´aclav Sn´aˇse:

This study investigated the use of Genetic algorithms in Information retrieval in the area of optimizing a Boolean query. A query with Boolean logical operators was used in information retrieval. For Genetic algorithms, encoding chromosomes was done from Boolean query; where it was represented in the form of tree prefix with indexing for all terms and all Boolean logical operators. Information retrieval effectiveness measures precision and recall used as a fitness function in their work. Other Genetic algorithms operators were used as single point crossover on Boolean logical operators, and mutation operator was used to exchange one of the Boolean operators and, or, and xor with any other one. The goal is to retrieve most relevant documents with less number of nonrelevant documents with respect to user query in Information retrieval system using genetic programming [34].

The results of this study suggest that the final population composed of individuals having the same strength (quality) will have the same precision and recall values. The best individual result was randomly chosen as best.

4. Abdelmgeid A. Aly:

This study presented an adaptive method using genetic algorithm to modify user's queries, based on relevance judgments. This algorithm was adapted for the three well-known documents collections (CISI, NLP and CACM). The method was shown to be applicable to large text collections, where more relevant documents were presented to users in the genetic modification. The algorithm showed the effects of applying GA to improve the effectiveness of queries in IR systems. The goal was to retrieve most relevant documents with less number of non-relevant documents with respect to user's query in information retrieval system using genetic algorithm [35]

This study was based on Vector Space Model (VSM) in which both documents and queries were represented as vectors; the weights were assigned to terms proposed by Salton and Buckle, and the system was evaluated by the precision and the recall formulae [35].

The GA in this study received an initial population chromosomes corresponding to the top 15 documents retrieved from classical IR with respect to that query.

We noticed that the result GA in the CISI documents collection gave a higher improvement than Classical IR system with 11.9%, in the NPL documents collection the GA gave a higher improvement than classic IR system with 11.5% as average values, and in The CACM documents collection GA gave a higher improvement than that with classic IR system 5.13%, as average values [35].

5. José R. Pérez-Agüera

This paper presented how an evolutionary algorithm can help to reformulate a user query to improve the results of the corresponding search. this method does not require any user supervision. Specifically, they have obtained the candidate terms to reformulate the query from a morphological thesaurus, with provides, after applying stemming, the different forms (plural and grammatical declinations) that a word can adopt. The evolutionary algorithm is in charge of selecting the appropriate combination of terms for the new query. To do this, the algorithm uses as fitness function a measure of the proximity between the query terms selected in the considered individual and the top ranked documents retrieved with these terms [36].

In this study carried out some experiments to have an idea of the possible improvement that the GA can achieve. In these experiments they have used the precision obtained from the user relevance judgments as fitness function. Results have shown that in this case the GA can reach a very high improvement [36].

This study also investigated different proximity measures as fitness functions without user supervision, such as cosine, square cosine, and square-root cosine. Experiments have shown that the best results are obtained with square root cosine. However, results obtained with this function do not reach the reference results obtained using the user relevance judgments. This suggests investigating other similarity measures as fitness functions. They have also studied the GA parameters, and see that small values such as a population size of 100 individuals, a crossover rate of 25% and a mutation rate of 1%, are enough to reach convergence [36].

6. Jorng-Tzong Horng, Ching-Chang Yeh

This paper proposed a novel approach to retrieve keywords automatically and then uses genetic algorithms to adapt their weights. The advantage of this approach is that it does not need a dictionary. This approach can retrieve any type of keywords, including types like technical keywords and people's names. The precision of document retrieval through this approach is equal to that of the PAT-tree based approach. However, the approach outlined in this paper requires less time and memory than the PAT-tree based approach does [37].

The genetic algorithm is applied to adapt keywords' weights. The new approach is used to retrieve Chinese documents according to the weights of keywords learned. Several deferent kinds of experiments validate the new approach. The new genetic approach performs better than several traditional information retrieval techniques, including the approach proposed by Yang and Korfhage (1993). This is because our new genetic approach can adjust the weights of keywords according the information from learned documents [37].

This study, also added the relevant feedback mechanism to improve document retrieval performance [37].

7. Abdelmgeid Amin

This paper presented two different proposals based on the vector space model (VSM) as a traditional model in information Retrieval (TIR). The first used evolution strategy (ES). The second used the document centroid (DC) in query expansion technique. Then the results compared; it was noticed that ES technique is more efficient than the other methods [38].

The test databases used in this study are three well-known test collections, which are: the CISI collection (1460 documents on information science), the CACM collection (3204 documents on Communications), and finally the NPL collection (11,429 documents on electronic engineering). One of the principal reasons for choosing more than one test collection is to emphasize and generalize our results in all alternative test documents collections. The Experiments are applied on 100 queries chosen according to each query which does not retrieve 15 relevant documents for our IR system [38].

Chapter Five

Methodology

This research was performed as the following steps:

1. The corpus of 242 Arabic abstracts collected from the proceedings of the Saudi Arabian National conference [6] used in this research.
2. Some text operations have been performed on those documents to determine documents terms, the following procedure is used:

- Extraction of all the words from each document.
- Elimination of the stop-words
- Stemming the remaining words using the porter stemmer [1], this is the most commonly used.

3. Indexing: an inverter file index used in this study.
4. After determining the terms that described the documents, the weights were assigned using the formula proposed by Salton and Buckley [39]

$$a_{ij} = \frac{\left[0.5 + 0.5\frac{tf_{ij}}{\max_{tf}}\right] * \log\frac{N}{n_i}}{\sqrt{\left[0.5 + 0.5\frac{tf_{ij}}{\max_{tf}}\right]^2 \left[\log\frac{N}{n_i}\right]^2}} \text{ ………………………….. (12)}$$

Where a_{ij} is the weight assigned to the term t_j in document D_i, tf_{ij} is the number of times that term t j appears in document D_i, n j is the number of documents indexed by the term *t j* and finally, N is the total number of documents in the database.

5. A traditional approach using (Vector Space Model, Extended Boolean Model, Language Model) a strategy based on an inverted index file, has been used (as shown in Figure 5.1), Then, the following steps have been applied:
   - For each model, each query is compared with all the documents. This yields a list giving the similarities of each query with all documents of the collection
   - This list is ranked in decreasing order of similarity degree.
   - Evaluate the retrieved document using average Recall and Precision formula (Eq.11)

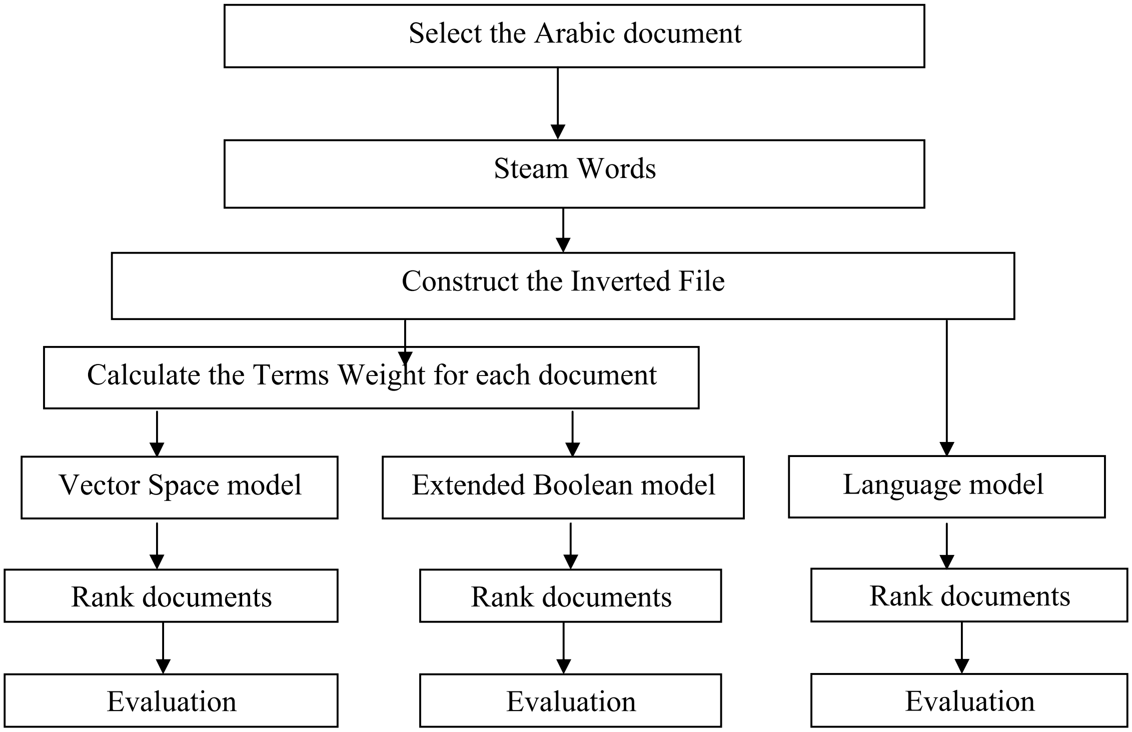

Figure 5.1 Traditional IR Approach

6. Make a training data consisting of the top 15 documents of the list with a corresponding query

7. Automatically, the top 15 documents were retrieved as training data considered as initial population to Adaptive Genetic Algorithms [as shown in Figure 5.2, a, b and c].

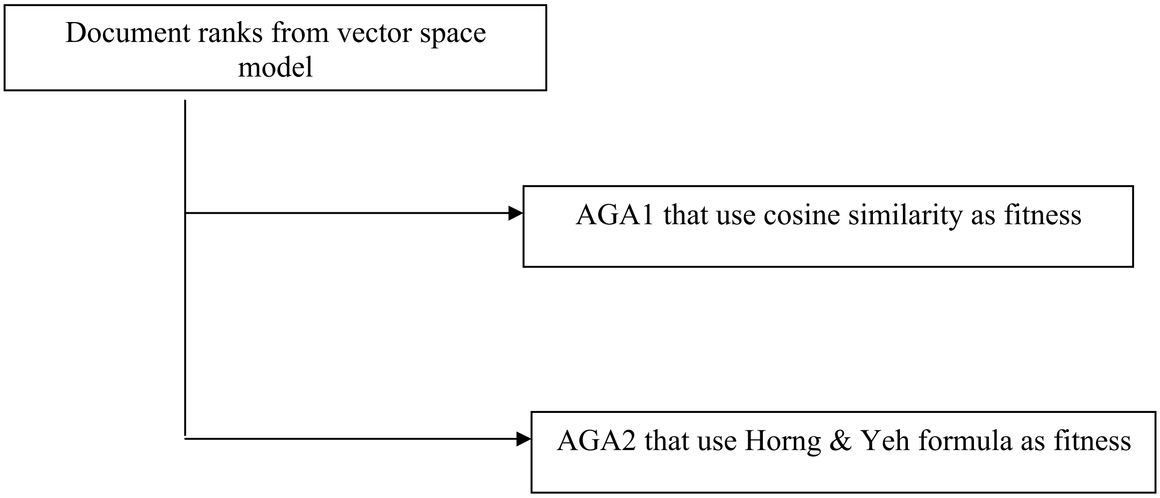
Document ranks from vector space model
AGA1 that use cosine similarity as fitness
AGA2 that use Horng & Yeh formula as fitness

Figure 5.2.a Vector Space Model with AGA Approaches

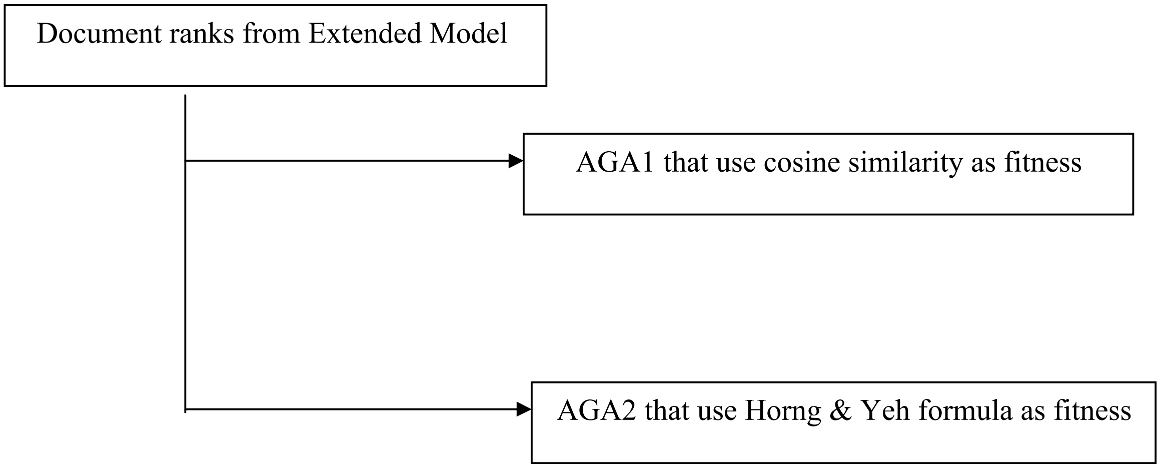
Document ranks from Extended Model
AGA1 that use cosine similarity as fitness
AGA2 that use Horng & Yeh formula as fitness

Figure 5.2.b Extended Boolean Model with AGA Approaches

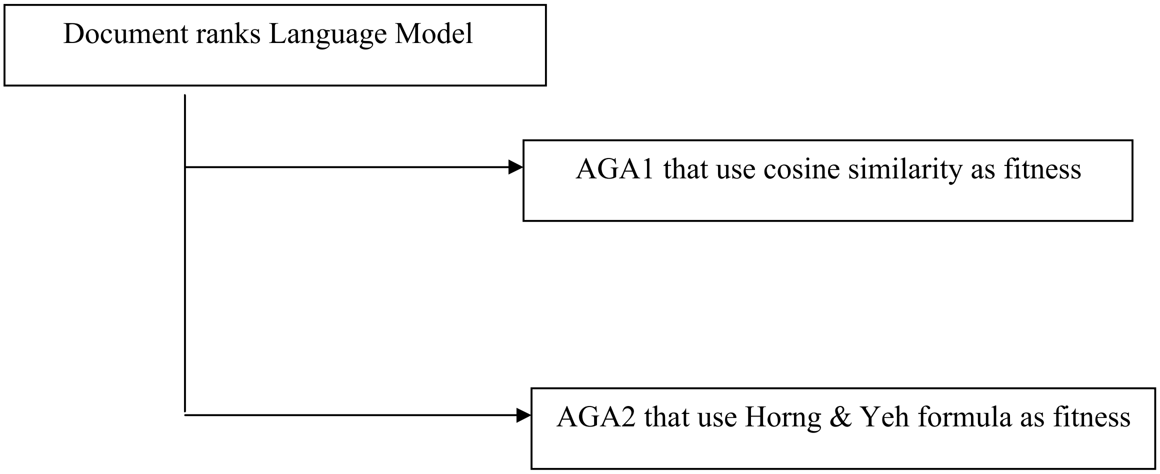
Document ranks Language Model
AGA1 that use cosine similarity as fitness
AGA2 that use Horng & Yeh formula as fitness

Figure 5.2.c Language Model with AGA Approaches

We give some details of the characteristics of the AGA that give the best performance; these characteristics guide the algorithm in its searching process in the following manner:

a) Representation of the chromosomes: AGA work with chromosomes using weights of terms representation, and have the same number of genes (components) as the query and the documents have terms with non-zero weights. First, the set of terms contained in those documents and the query are calculated, and the size of the chromosomes is equal to the number of terms of that set

.

.

b) The population: AGA receives an initial population consisting of the chromosomes corresponding to the relevant documents. The population is represented by terms of weight.

c) Genetic operators: Our algorithm uses the one-point crossover operator and point mutation

d) Control parameters: The values of the control parameters crossover probability (pc) and mutation probability (pm) are variable, the fitness function is the similarity

e) Fitness: The following functions are used in determining the fitness values [32], [3]:

Fitness 1: Horng and Yeh formula

$$F = \frac{1}{|D|}\sum_{i=1}^{|D|}\left(r\left(di\right)\sum_{j=1}^{|D|}\frac{1}{j}\right) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (13)$$

Where $|D|$ is the total number of documents retrieved, and r(di) is the function that returns the relevance of document d, giving a 1 if the document is relevant and a 0 otherwise. We shall refer to this fitness function as fitness 1.

Fitness 2: cosine similarity:

$$F = \frac{\sum_{k=1}^{t}(d_{ik} \bullet q_{k})}{\sqrt{\sum_{k=1}^{t} d_{ik}^{2} \bullet \sum_{k=1}^{t} q_{k}^{2}}} \qquad \ldots\ldots\ldots\ldots\ldots\ldots (14)$$

Where d_{ik} is the weight of term i in document k and q_k is the weight of term i in the query

8. .Evaluate the retrieved document using average Recall and Precision formula
9. Compare effectiveness between different AGA approaches for each model
10. Compare effectiveness between the best AGA with the best GA approach.

Chapter Six

The proposed Algorithm

We used adaptive genetic algorithm (AGA) that has been optimized and adapted for relevance feedback, we next describe the characteristics of this AGA, chosen for having the best performance by using crossover and mutation operators with variable probabilities, where as the traditional genetic algorithm (GA) uses fixed values of those, and remains unchanged during execution. GA is developed to support adaptive adjustment of mutation and crossover probabilities; this allows faster attainment of better solutions, and then we describe two different fitness functions, both based on the order of retrieval, which we used to guide the algorithm in the search process.

6.1 The process of the AGA

1. Representation of the chromosomes:

the chromosomes use binary representation these chromosomes have the same number of genes (components) as there are terms with nonzero weights in the query and in the documents of the feedback. One first calculates the set of different terms contained in those documents and in the query, and the size of the chromosomes is equal to the number of terms in that set.

2. Population**:**

Our AGA receives an initial population consisting of the chromosomes corresponding to the top 15 documents retrieved from traditional IR with respect to that query.

3. Selection**:**

The selection process select two parent chromosomes from a population according to their fitness (the better fitness, the bigger chance to be selected).

4. Genetic operators:

We used one-point crossover as the crossover operator. It is defined as follows: Given two parent chromosomes *C*1 = (a1. . . am) and C2= (b1 . . . bm), one generates two offspring chromosomes H1= (a1. . . ai, bi + 1, . . .,bm) and H2 = (b1, . . . , bi, ai + 1,. . . , am), where *i* is a random number in the interval [1, *m*_ 1] and *m* is the length of the chromosome. Mutation in our algorithm is implemented as a random process. A real random number is generated in a given interval, in our case [0, 1], and that number is taken as the new value for the gene that has to mutate.

5. Control parameters:

Crossover probability Pc and mutation probability Pm play an important role in GA. Crossover causes a randomized exchange of genetic material between chromosomes. Crossover occurs only with some probability Pc which controls the rate at which chromosome is subjected to crossover [26], [27]. The larger value Pc is, the faster is the new chromosome introduced into the population. The smaller value Pc is, the lower the searching process is leading to stagnation. Typical initial value of Pc is in the range 0.5 to1.0. The mutation probability Pm is varied according to the generations. The initial Pm is larger for the global search, and in some generations it is smaller for the local search. Finally it is larger again for avoidance of local optimum. Typical initial value of Pm is in the range 0.005 to 0.05.

We put forward adaptive varied values of P_c and P_m as follows [26], [27]:

$$p_c = \begin{cases} p_{c1} - \dfrac{(p_{c1} - p_{c2}) * (f' - f_{avg})}{f_{\max} - f_{avg}}, f' \geq f_{avg}, \text{.....................(15)} \\ p_{c1}, f' \prec f_{avg} \end{cases}$$

$$p_m = \begin{cases} p_{m1} - \dfrac{(p_{m1} - p_{m2}) * (f - f_{avg})}{f_{\max} - f_{avg}}, f' \geq f_{avg}, \text{....................... (16)} \\ p_{m1}, f' \prec f_{avg} \end{cases}$$

Here, f_{max} is the maximum fitness function of current generation, f_{avg} is the average fitness function of current generation, f' is larger fitness function of the two crossover chromosomes selected, f is the fitness function of mutation chromosome selected, p_{c1}; p_{c2} is crossover probability, and p_{m1}, p_{m2} is mutation probability. The study experimental parameters include: $p_{c1} = 0.9$; $p_{c2} = 0.6$, $p_{m1} = 0.1$, and $p_{m2} = 0.001$.

:

6. End Condition:

GA needs an End Condition to end the generation process. If we have no sufficient improvement in two or more consecutive generations; the number of iteration used in this research is 75 iterations.

7. The Fitness Functions:

We ran the AGA described above with different order based fitness functions**:**

Fitness 1**:** This fitness function, due to Horng and Yeh (2000), is very innovative. As well as taking into account the number of relevant and of irrelevant documents, it also takes account of the order of their appearance, because it is not the same that the relevant documents appear at the beginning or at the end of the list of retrieved documents.

One calculates the similarity of the query vector with all the documents, and sorts the documents into decreasing order of similarity. Finally, calculates the fitness value of the chromosome using (Eq.13).

Fitness 2: Cosine similarity using (Eq.14).

6.2 Pseudocode of Proposed Adaptive Genetic Algorithm

The pseudocode AGA process is expressed as follows:

```
1. [Start] Generate random population
2. [Fitness] Evaluate the fitness of each individual in the population
3. [New population] Create a new population by repeating the following steps until the end condition
    A. [Selection] Select two parents from a population according to their fitness
    B. [crossover rate] Calculate the genetic crossover to find the best rate for individual according to the fitness
    C. [Crossover] Crossing of two individuals by using the parameter of the individual with best fitness to form new offspring
    D. [mutation rate] Calculate the genetic mutation to find the best rate for individual according to the fitness
    E. [Mutation] Mutation of new offspring at each position in chromosome by using the best parameter
    F. [Accepting] Place new offspring in the new population
    G. [Parameter evolution] Create a new set of parameters for each individual in population
        i. [Selection] Select two parameters settings from a population according to their    reinforcement position
```

ii. **[Crossover]** Crossing of two parameters sets by using the parameter of the individual

iii. **[Mutation]** Mutation of new offspring by using the parameter of the individual

iv. **[Accepting]** Place new offspring in the new sets parameters population

4. **[Replace]** Use newly generated population for a further run of the algorithm

5. **[Test]** If the end condition is satisfied, stop, and return the optimized query

6. **[Loop]** Go to step 2

Chapter Seven
Results and Discussion

The dataset used in this research is the 242 Arabic abstracts collected from the proceeding of the Saudi Arabian National Conference [6]. The inverted file indexing is used because it has proved the best indexing method in most studies that deal with IR systems.

The traditional information retrieval systems were built and implemented to handle the Arabic collection using C# NET, and run on acer/ pcs. The following three IR systems were built and implemented:

1. System that used Vector Space Model (VSM) with Cosine similarity
2. System that used Extended Boolean Model (EBM)
3. System that used Language model (LM)

Different adaptive genetic algorithm (AGA) strategies were used in this research. Those strategies are as the following:

AGA1: AGA that use cosine similarity as fitness.

AGA2: AGA that use Horng & Yeh formula as fitness.

System evaluation: Evaluation is a key part of this research, we need to Find the average precision (Eq.11) for each query and compute the mean AP over 59 queries, average precision take the mean of the precision at each recall point for 59 queries together and take average at standard recall points.

7.1 Applying AGA on Vector Space Model

Table 7.1 and Figure 7.1 show the comparison between vector space model with Cosine as fitness VSM (AGA1) and vector space model with Horng as fitness VSM (AGA2), from this table and corresponding figure we notice that the VSM (AGA2) represent the best strategy over VSM (AGA1).

Table 7.1 Average Precision Values for 59 Queries by Applying AGAs on Vector Space Model

	Average Precision	
Recall	**VSM (AGA1)**	**VSM (AGA2)**
0.1	0.46	0.48
0.2	0.53	0.56
0.3	0.57	0.63
0.4	0.65	0.66
0.5	0.75	0.8
0.6	0.75	0.8
0.7	0.85	0.9
0.8	0.9	0.94
0.9	0.9	0.95
Average	0.651	0.736

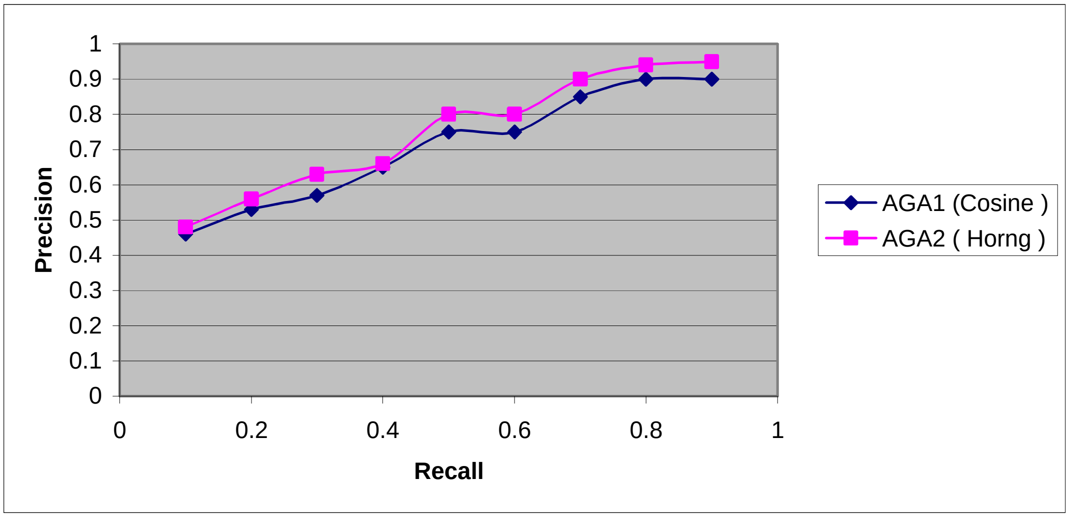

Figure: 7.1 Comparisons Between Deferent AGA in Vector Space Model

7.2 Applying AGA on Extended Boolean Model

Table 7.2 and Figure 7.2 show the comparison between Extended Boolean Model with Cosine as fitness EBM (AGA1) and EBM with Horng as fitness (AGA2), from this table and corresponding figure we notice that the EBM with Horng as fitness (AGA2) represent the best strategy over EBM with Cosine as fitness (AGA1).

Table 7.2 Average Precision Values for 59 Queries by Applying AGAs on Extended Boolean Model

	Average Precision	
Recall	**EBM (AGA1)**	**EBM (AGA2)**
0.1	0.13	0.30
0.2	0.17	0.38
0.3	0.29	0.47
0.4	0.52	0.52
0.5	0.6	0.6
0.6	0.7	0.7
0.7	0.75	0.8
0.8	0.82	0.89
0.9	0.90	0.97
Average	0.542	0.625

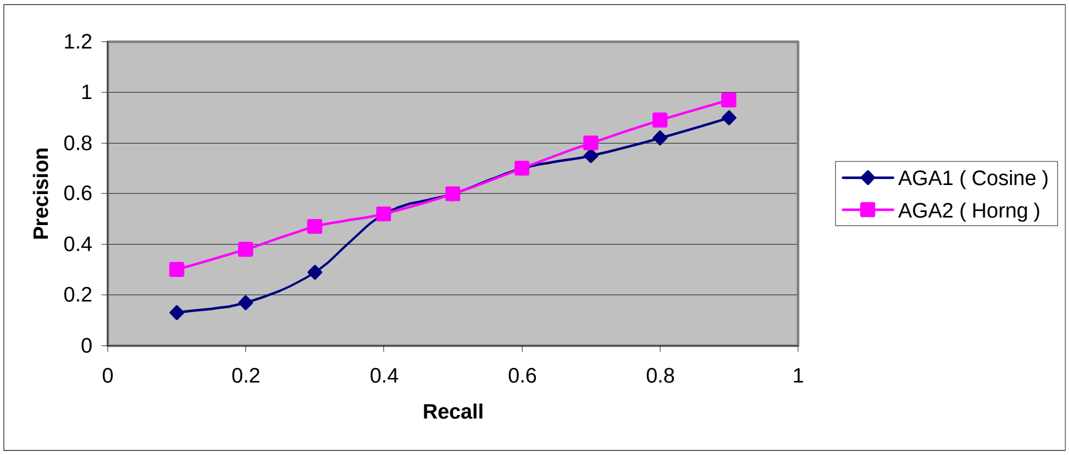

Figure: 7.2 Comparison Between Deferent AGA in Extended Boolean Model

The number of iteration used in this research is 75 iterations, and that explain why some point in AGA approaches have closes together, so if the number of iteration is increased the AGA2 will have the highest improvement over AGA1.

7.3 Applying AGA on Language Model

Table 7.3 and Figure 7.3 show the comparison between Language Model with Cosine as fitness LM (AGA1) and Language Model with Horng as fitness LM (AGA2), from this table and corresponding figure we notice that the LM with Horng as fitness (AGA2) represent the best strategy over LM with Cosine as fitness (AGA1).

Table 7.3 Average Precision Values for 59 Queries by Applying AGAs on Language Model

	Average Precision	
Recall	**LM (AGA1)**	**LM (AGA2)**
0.1	0.1	0.14
0.2	0.11	0.22
0.3	0.14	0.4
0.4	0.19	0.46
0.5	0.33	0.53
0.6	0.59	0.62
0.7	0.66	0.67
0.8	0.77	0.8
0.9	0.9	0.97
Average	0.421	0.534

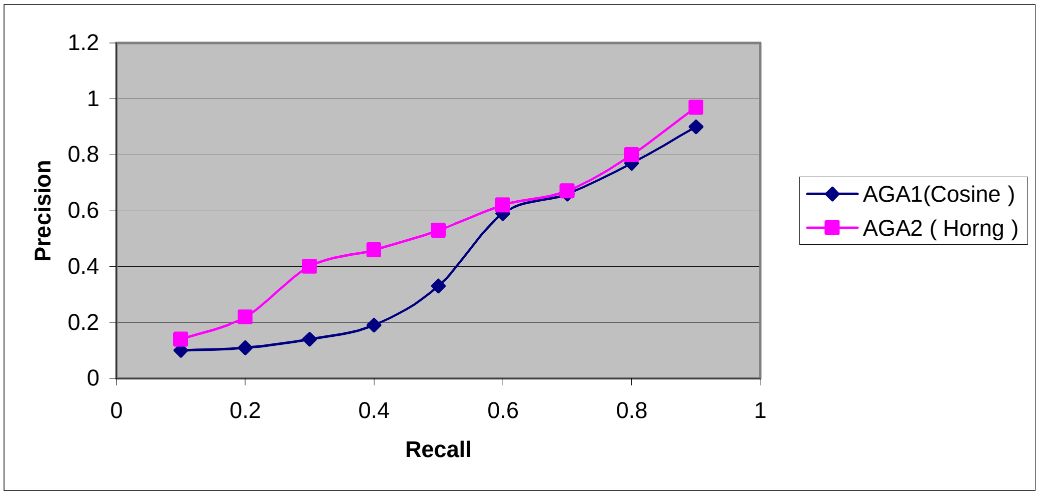

Figure: 7.3 Comparison Between Deferent AGA in Language Model

The number of iteration used in this research is 75 iterations, and that explain why some point in AGA approaches have closes together, so if the number of iteration is increased the AGA2 will have the highest improvement over AGA1.

7.4 AGA Using Cosine Similarity

Table 7.4, and Figure 7.4 show the comparison between VSM(AGA1) ,EBM(AGA1), And LM(AGA1) with Cosine as fitness, from this table and corresponding figure we notice that the VSM(AGA1) represent the best strategy over EBM(AGA1), And LM(AGA1)**.**

Table 7.4 Average Precision Values for 59 Queries by Applying AGAs with Cosine Similarity Fitness

	Average Precision		
Recall	**VSM(AGA1)**	**EBM(AGA1)**	**LM(AGA1)**
0.1	0.46	0.13	0.1
0.2	0.53	0.17	0.11
0.3	0.57	0.29	0.14
0.4	0.65	0.52	0.19
0.5	0.75	0.6	0.33
0.6	0.75	0.7	0.59
0.7	0.85	0.75	0.66
0.8	0.9	0.82	0.77
0.9	0.9	0.90	0.9
Average	0.651	0.542	0.421

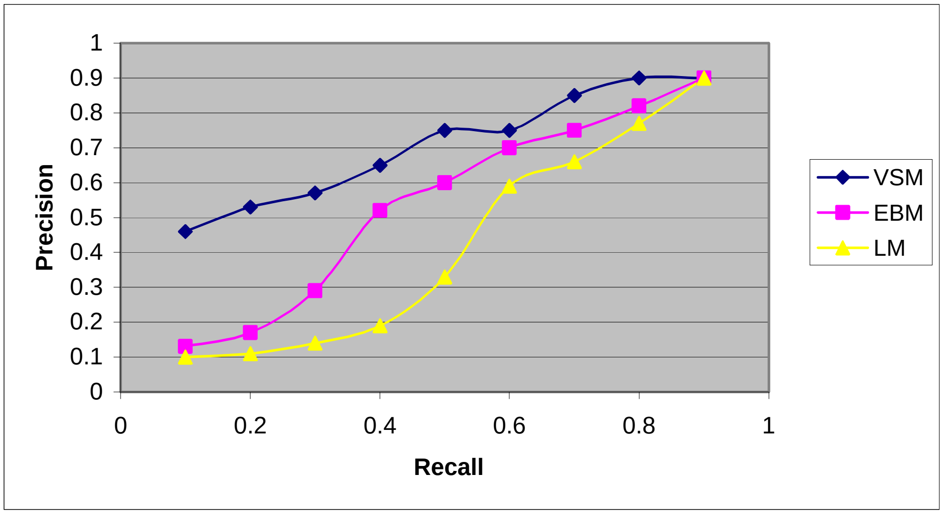

Figure 7.4: Average Precision Values for 59 Queries by Applying AGAs with Cosine Similarity Fitness

7.5 AGA Using Horng & Yeh Formula

Table 7.5 and Figure 7.5 show the comparison between VSM(AGA2) ,EBM(AGA2), And LM(AGA2) with Horng & Yeh formula as fitness, from this table and corresponding figure we notice that the VSM(AGA2) represent the best strategy over EBM(AGA2), And LM(AGA2)

Table 7.5 Average Recall and Precision Values for 59 Queries by Applying AGAs with Horng & Yeh Formula

	Average Precision		
Recall	**VSM(AGA2)**	**EBM(AGA2)**	**LM(AGA2)**
0.1	0.48	0.30	0.14
0.2	0.56	0.38	0.22
0.3	0.63	0.47	0.4
0.4	0.66	0.52	0.46
0.5	0.8	0.6	0.53
0.6	0.8	0.7	0.62
0.7	0.9	0.8	0.67
0.8	0.94	0.89	0.8
0.9	0.95	0.97	0.97
Average	0.736	0.625	0.534

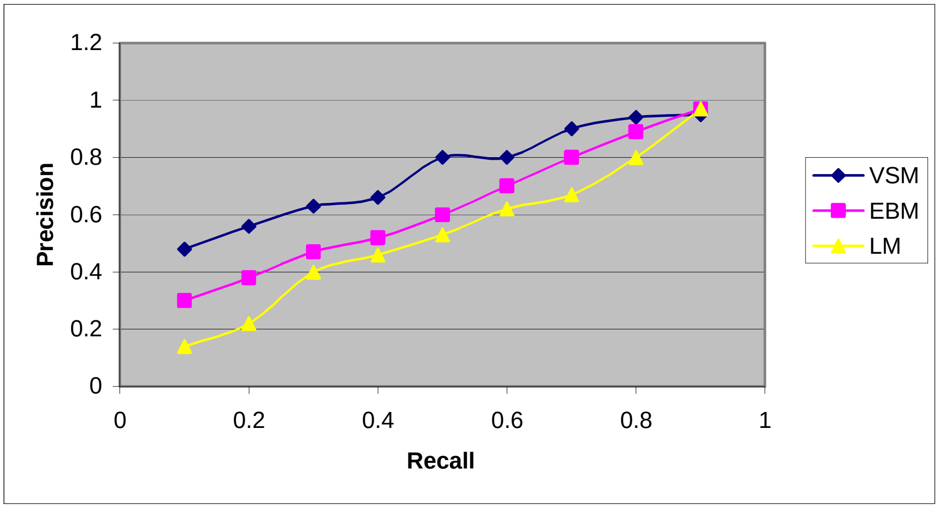

Figure 7.5 Average Precision Values for 59 Queries by Applying AGAs with Horng & Yeh Formula

7.6 Comparison Between Best AGAs Strategy with Traditional GAs

In this comparison best AGAs strategy compares with traditional GAs that based on the thesis done by feras Mashakbeh [32].

The results for the AGAs are shown in table 7.6, table 7.7, figure 7.6, and figure 7.7 using the average Recall and Precision relationship. From those tables and corresponding figures, we notice that VSM with Horng as fitness VSM (AGA2) compare with VSM with Cosine as fitness under traditional Genetic Algorithm VSM (GA) gives the highest improvement over VSM (GA) with **55.1%**., and EBM with Cosine as fitness EBM (AGA1) compare with Boolean Model with precision as fitness under traditional Genetic Algorithm BM (GA) gives the highest improvement over BM (GA) with **42.1%**.

. Table 7.6: Comparison between VSM (AGA2) and VSM (GA)

Recall	**Average Precision**		**AGA Improvement %**
	VSM (GA)	**VSM (AGA2)**	
0.1	0.16	0.48	32
0.2	0.17	0.56	39
0.3	0.18	0.63	45
0.4	0.18	0.66	48
0.5	0.19	0.8	61
0.6	0.2	0.9	7
0.7	0.2	0.9	7
0.8	0.24	0.9	66
0.9	0.25	0.9	65
Average	0.196	0.736	55.1%

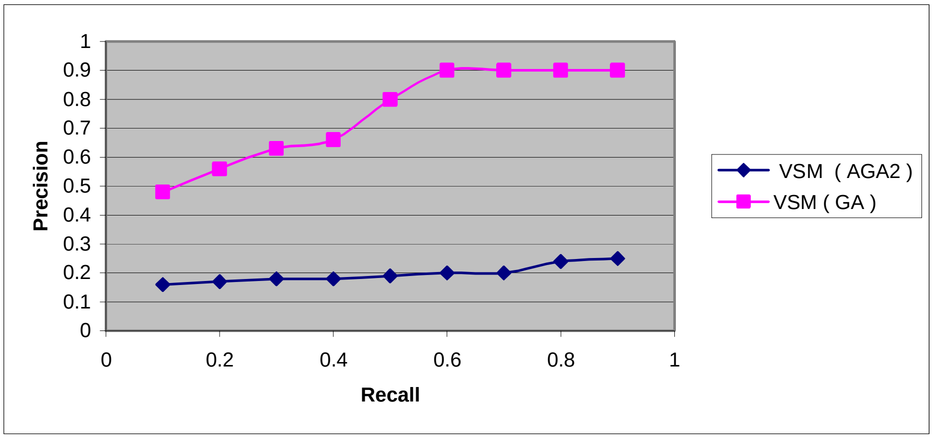

Figure 7.6: Average Precision Values for VCM (AGA2) and VCM (GA)

Table 7.7 Comparison between EBM (AGA2) and BM (GA)

Recall	Average Precision		AGA Improvement %
	BM (GA)	EBM (AGA2)	
0.1	0.17	0.30	13
0.2	0.17	0.38	21
0.3	0.18	0.47	29
0.4	0.19	0.52	33
0.5	0.2	0.6	40
0.6	0.23	0.7	47
0.7	0.24	0.8	56
0.8	0.26	0.89	63
0.9	0.27	0.97	70
Average	0.212	0.625	42.1%

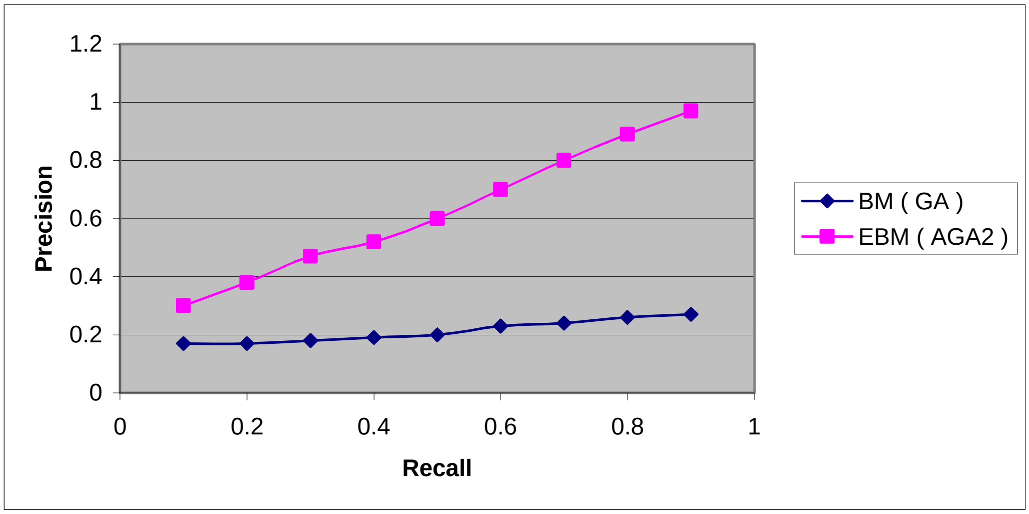

Figure 7.7 Average Precision Values for EBM (AGA2) and EB (GA)

7.7 Optimized Query

In the research, AGA employs query concept to find the most similar term to the query and adds this term to the query to form another query terms. This process is repeated until there is no more similar term to the query concept.

From table 7.8 and table 7.9 wee notices that the AGAs are able to add terms which improve the system performance, the AGA is able to recover the original query, and we can observe that the most optimized query representative terms related to the topic

Table 7.8 Optimization Query Using AGA1

Query	Optimized Query
استرجاع المعلومات	رجع نظم خدم تقن درس وصل
التعليم بمساعدة الحاسب	علم وطن عرب نظم صنع
علوم الحاسب و المعلومات	علم وطن نظم عبد حرف درب دور وسط
الذكاء الاصطناعي	ذكا علم حسب شكل درس كلم ميز رجع
هندسة الحاسب الالي	هند حسب وطن عبد وسط دور صغر
محاكاة الحاسب الالي	حاك حسب علم عبد ندس وطن شكل خدم بان جمل

Table 7.9 Optimization Query Using AGA2

Query	Optimized Query
استرجاع المعلومات	رجع علم حسب وطن درس عبد دور ذكا
التعليم بمساعدة الحاسب	علم نحسب درس خطط
علوم الحاسب و المعلومات	علم حسب عبد تقن جمع لغو خبر رمز خطة كمل وسط
الذكاء الاصطناعي	ذكا وطن عبد حسب قعد سعد ركب خدم وصل
هندسة الحاسب الالي	هندس حسب
محاكاة الحاسب الالي	حاك علم وطن كرم وصل تقن درس دار

Chapter Eight

Conclusion and Future Work

8.1 Conclusion

The research apply Adaptive Genetic Algorithm(AGA) with different fitness functions(Cosine and Horng) and variable operators rate(crossover and mutation) on vector space model, extended model, and language model .

In vector space model, the research compares different adaptive genetic algorithm strategies by calculating evaluation using average recall formula .from table 7.1 and figure 7.1, we noticed that the vector space model with Horng as fitness represent the best strategy over vector space model with Cosine as fitness.

In Extended Boolean Model, the research compares different adaptive genetic algorithm strategies by calculating evaluation using average recall formula .from table 7.2 and figure 7.2, we noticed that the Extended Boolean Model with Cosine as fitness represent the best strategy over Extended Boolean Model with Horng as fitness.

In Language Model, the research compares different adaptive genetic algorithm strategies by calculating evaluation using average recall formula .from table 7.3 and figure 7.3, we noticed that the Language Model with Horng as fitness represent the best strategy over Language model with Cosine as fitness

The research compare between the best adaptive genetic algorithm strategies by calculating the improvement over the traditional genetic algorithm. From 7.6, table 7.7, figure 7.6, and figure 7.7, we noticed that the vector space model with Horng as fitness compare with vector space model with Cosine as fitness under traditional Genetic Algorithm gives the highest improvement over vector space model with Cosine as fitness under traditional Genetic with **55.1%**and EBM with Cosine as fitness EBM (AGA1) compare with Boolean Model with precision as fitness under traditional Genetic Algorithm BM (GA) gives the highest improvement over BM (GA) with **42.1%.**

8.2 Future Work

This research apply AGA approaches on the corpus of 242 Arabic abstracts collected from the proceeding of the Saudi Arabian National Conferences, in the future AGA approach may be applied on other Arabic corpus and larger ones.

In this research we constrain our attention to three families of models: Vector space, extended Boolean and language models, in the future different models may be used such as fizzy set model, probability model.

And in this research also we constrain our attention to the use of Adaptive Genetic Algorithm (AGA) that use crossover and mutation operators with variable probability; in the future AGA with different operators may be used such as type of operators, adaptive fitness, and variable size of chromosome.

References

[1] Christopher D.Manning and Prabhakar Raghavan**, An Introduction To Information Retrievalk** , Cambridge University Press, Cambridge, England, 2008.

[2] Nasredine Semmar, Faïza Elkateb-Gara and Christian Fluhr, **Using a Stemmer in a Natural Language Processing system to treat Arabic for Cross-language Information Retrieval**, Proceedings of the 5th Conference On Language Engineering, pp. 1-10, 2005.

[3] Egidio Terra and Robert Warren, **Poison Pills: Harmful Relevant Documents in Feedback**, Proceedings of the 14th ACM international conference on Information and knowledge, pp. 319 – 328, 2005.

[4] Cristina Lopez-Pujalte, Vicente P. Guerrero-Bote and Felix de Moya-Anegon, **Genetic algorithms in relevance feedback: a second test and new contributions**, Proceedings in Information Processing and Management, Vol .39, PP .53-51, 2003.

[5] Bangorn Klabbankoh, **Applied Genetic Algorithm in Information Retrieval**,Proceedings of the International journal of the computer, the internet and management, Vol. 7, No. 3, pp. 60-66, 1999.

[6] Hmeidi. I, Kanaan. G , and Evens. M, **Design and Implementation of Automatic Indexing for Information Retrieval with Arabic Documents**, Proceedings of the Journal of the American Society for Information Science,Vol. 48(10), pp. 867–81,1998.

[7] Ronan Cummins and Colm O'Riordan, **Determining general term weighting schemes for the Vector Space Model of Information Retrieval using Genetic Programming**, Proceedings of the 15th Artificial Intelligence and Cognitive Science Conference (AICS), pp.43-57, 2004.

[8] hanandeh E, **Evaluation of Different Information Retrieval models and Different indexing methods on Arabic Documents**, PhD Thesis, Faculty of

Information System and Technology , The Arab Academy for Banking and Financial Science , Amman , Jordan, 2008 .

[9] Graham Bennett Falk Scholer Alexandra Uitdenbogerd,**A Comparative Study of Probabilistic and Language Models for Information Retrieval**, Proceedings of Nineteenth Australasian Database Conference (ADC), Vol. 75, pp 1- 10, 2008.

[10] Andrew Trotman, **An Artificial Intelligence Approach to Information Retrieval**, Proceedings of the 27th Annual International ACM SIGIR Conference on Research and development in Information retrieval, pp. 603 – 608, 2004.

[11] Ramesh Nallapati, Bruce Croft and James Allan, **Relevant Query Feedback in Statistical Language Modeling** ,Proceedings of the 12th international conference on Information and knowledge management, pp. 560 – 563,2003.

[12] Xiaohua Zhou, **Semantics-based Language Models for Information Retrieval and Text Mining,** PhD thesis, Drexel University, 2008.

[13] Paul Ogilvie, Jamie Callan, **Language Models and Structured Document Retrieval**, Proceedings of the Initiative for the Evaluation of XML Retrieval Workshop, pp. 211-224, 2003.

[14] Jianfeng Gao Jian-Yun Nie, Guangyuan Wu, Guihong Cao,**Dependence Language Model for Information Retrieval**, Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval, pp.265-278, 2004.

[15] Jen-Yuan Yeh, Jung-Yi Lin, Hao-Ren Ke, Wei-Pang Yang**, Learning to Rank for Information Retrieval Using Genetic Programming**, Proceedings of ACM SIGIR Workshop on Learning to Rank for Information Retrieval, Vol.11, pp.176-215, 2007.

[16] Ronan Cummins, Colm O'Riordan, **Using Genetic Programming for Information Retrieval: Local and Global Query Expansion**, Proceedings of the 9th annual conference on Genetic and evolutionary computation, pp.2255-2258, 2007 .

[17] Bangorn Klabbankoh, **Applied Genetic Algorithm in Information Retrieval**, Proceedings of International journal of the computer, the internet and management, Vol.7, No.3, pp. 60-66, 1999.

[18] Horng & C.C Yeh, **Applying genetic algorithms to query optimizations in document retrieval**, Proceedings of the Information Processing and Management, Vol 36, pp.737-759, 2000.

[19] Herrera-Viedma, **An Information Retrieval Model With Ordinal Linguistic Weighted Queries Based On Two Weighting Elements,** Proceedings of International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems , Vol.2, No. 2, pp 1-11, 2001.

[20] M. Boughanem, Chrisment and Tamine **"using genetic algorithms for multimodal relevance optimization in information retrieval"**, proceeding of the Journal of American Science on Information and Technology,Vol.12, pp.367-382, 2004.

.[21] Desjardins and Godin,R, **Combining Relevance Feedback and Genetic Algorithm in Internet Information Filtering Engine**, Proceedings of the RIAO Content-Based Multimedia Information Access, Vol.13, pp.67-94, 2000.

[22] Stuart J.Russel and Peter Norving , **Artificial Intelligence a modern approach** , Prentice Hall , 2004

[23] Mike Sewell, Jagath Samarabandu, Ranga Rodrigo, **The Rank-scaled Mutation Rate for Genetic Algorithms**, proceeding of International Journal of Information Technology, Vol .3 No.1, pp 32-36.2006.

[24] Shengxiang Yang, **Adaptive Crossover in Genetic Algorithms Using Statistics Mechanism**, Proceedings of the 8th international conference on Artificial life PP.182-185, 2002.

[25] Nga Lam Law and K.Y. Szeto, **Adaptive Genetic Algorithm with Mutation and Crossover Matrices,** Proceedings of the 20th international joint conference on Artificial intelligence, pp. 115-120,2007.

[26] Wang Lei and Shen Tingzhi, **An Improved Adaptive Genetic Algorithm and its application to image segmentation,** Proceeding of 5th International Conference on Artificial Neural Network and Genetic Algorithms, PP.112–119,2004.

[27] Mauro Annunziato and Stefano Pizzuti, **Adaptive Parameterization of Evolutionary Algorithms Driven by Reproduction and Competition,** Proceedings of Genetic and evolutionary computation conference, pp.1597-1598, 2005.

[28] Eric Pellerin, Luc Pigeon and Sylvain Delisle, **Self-Adaptive Parameters in Genetic Algorithms,** proceeding of Data Mining and Knowledge Discovery Conference,Vol.53, pp. 61-100, 2004.

[29] Sima (Etaner) Uyar, Gulsen (Cebiroglu) Eryigit and Sanem Sariel**, An Adaptive Mutation Scheme in Genetic Algorithms for Fastening the Convergence to the Optimum,** proceeding of 3rd Asian Pacific International Symposium on Information Technologies APIS, pp.1257-1264 , 2005.

[30] Imtiaz Korejo, Shengxiang Yang and ChangheLi**, A Comparative Study of Adaptive Mutation Operators for Genetic Algorithms**, proceeding of Met heuristics International Conference(MIC),Vol.8 ,pp.18-46 , 2009.

[31] Sonja Novkovic, and Davor Sverko, **A Genetic Algorithm With Self-Generated Random Parameters**, proceeding of Journal of Computing and Information Technology (CIT) ,Vol.11, pp.271 -283 ,2003.

[32] Feras AL-Mashakbeh , **Evaluate the Effectiveness of Genetic Algorithm in Information Retrieval Based on Arabic Documents,** Unpublished PhD Thesis , Faculty of Information System and Technology , The Arab Academy for Banking and Financial Science , Amman , Jordan, 2008 .

[33] Poltak Sihombing,, Abdullah Embong and Putra Sumari**, Comparison of Document Similarity in Information Retrieval System by Different Formulation,** proceeding of the 2nd IMT-GT Regional Conference on mathematics ,pp244-287, 2006.

[34] Suhail S. J. Owais, Pavel Kr¨omer and V´aclav Sn´aˇse**, Query optimization by Genetic Algorithms,** Proceedings of the Dateso Annual International Workshop on Databases, pp.125–137, 2005.

.

[35] Abdelmgeid Aly, **Applying Genetic Algorithm In Query Improvement Problem**, Proceedings of International Journal "Information Technologies and Knowledge", Vol.1, pp.309-316, 2007.

[36] José R. Pérez-Agüera, **Using Genetic Algorithms for Query Reformulation,** Proceedings of Future Directions in Information Access conferences ,vol. 23, pp. 215-231, 2007.

[37] Jorng-Tzong Horng and Ching-Chang Yeh, **Applying genetic algorithms to query optimization in document retrieval**, Proceedings of Information Processing and Management Conferences, Vol.36, pp. 737-759, 2006

[38] Abdelmgeid Admin**, Enhancing Information Retrieval By Using Evolution Strategies**, Proceedings of International Journal "Information Theories & Applications" ,Vol.15, pp. 369-367, 2008.

[39] G, Salton, C. Buckley, Improving **Retrieval Performance by Relevance Feedback**, Proceedings of Journal of the ASIS, Vol. 41, pp. 288-297, 1990.

الملخص

تستخدم الخوارزميات الجينية في البحث عن الخيار الأمثل من مجموعة حلول متوفرة لتصميم معين، وتعتمد مبدأ داروين في الاصطفاء حيث تقوم هذه المعالجة الوراثية بتمرير المزايا المثلى من خلال عمليات التوالد المتعاقبة، وتدعيم هذه الصفات، وتكون لهذه الصفات القدرة الأكبر على دخول عملية التوالد، وإنتاج ذرية أمثل وبتكرار الدورة الوراثية تتحسن نوعية الذرية تدريجياً.

الخوارزمية الجينية بدأ تطبيقها في نظام استرجاع المعلومات من أجل تحسين الاستعلام ، في الدراسات السابقة اعتمدت الخوارزمية على معايير تحكم ثابتة مثل احتمالية حدوث التهجين و احنمالية حدوث الطفرة .

تم في هذه الدراسة اقتراح خوارزمية الجينات التكيفية و التي تعتمد في أدائها على استخدام معايير تحكم متغيره لاحتمالية حدوث التهجين و الطفرة, طبقت هذه الخوارزمية على نظام استرجاع المعلومات باستخدام نماذج مختلفه وهي : امثلة المتجهات الفضائية , النموذج المنطقي الموسع ، ونموذج اللغة في استرجاع المعلومات .

أظهرت نتائج هذه الدراسة ان الخوارزميه الجينيه التكيفية كانت افضل أداء مقارنة مع الخوارزميه الجينية. كما اظهرت نتائج هذه الدراسة أن الخوارزمية الجينية التكيفية المستخدمة لدالة Horng كانت أفضل أداء من الخوارزمية الجينية التكيفية المستخدمة لدالة Cosine.

Appendix A: Sample of data set

1. Abstract 1:

صنف البرمجة
فهر ح . ح
نوع مؤتمر
عنو تنفيذ لغة برمجة عربية على حاسب آلي مصغر
مؤل خياط , محمد غزالي
جهه جامعة البترول والمعادن , الظهران
ممو معهد الادارة العامة , الرياض
عنم المؤتمر والمعرض الوطني السابع للحاسبات الالكترونية 18 - 22
ربيع الثاني 1404 هـ : سجل البحوث
صفح 94 - 103
عدم 20
نشر 1404 هـ
ناش اللجنة الفرعية للبحوث والبرامج , معهد الادارة العامة , الرياض
لغه العربية
ملخ ان استخدامات الحاسبات الآلية الحالي باللغة العربية تعتمد اساسا
على برامج مكتوبة بلغة برمجة اجنبية . وعلى هذا فان ذلك يمثل تقدما
سطحيا في مجال تعريب استخدامات الحاسبات الآلية . ولذا فانه يتوجب
تطوير لغة برمجة عربية لتمكين المستخدم والمبرمج من استخدام الحاسبات
الآلية باللغة العربية دون اللجوء الى استعمال لغة اجنبية . نقدم في
هذا البحث تصميم وتنفيذ مترجم للغة برمجة عربية . لغة الضاد تم
تطويرها بدعم من المركز الوطني للعلوم والتكنولوجيا (مشروع ات 455
) ويقوم المترجم بترجمة برامج مكتوبة بلغة البرمجة المفتوحة الى سلسلة
من التعليمات التي يمكن ان يقوم الحاسب الآلي بتنفيذها . ويجب ان
تتوافر في المترجم عدة صفات من اهمها الوضوح وسهولة وانتظام التركيب

2. Abstract 2:

نوع مؤتمر
عنو كيف تختار جهاز آلي مصغر
مؤل خياط , محمد غزالي , الدجاني , عبدالباسط
جهه جامعة البترول و المعادن , الظهران
ممو معهد الادارة العامة , الرياض
عنم المؤتمر والمعرض الوطني السابع للحاسبات الالكترونية 18 - 22 ربيع الثاني 1404 هـ : سجل البحوث
صفح 114 - 129
نشر 1404 هـ
ناقش اللجنة الفرعية للبحوث والبرامج , معهد الادارة العامة , الرياض
لغه العربية

ملخ يقدم هذا المقال طريقة علمية لاختيار جهاز حاسب آلي صغير , وتتلخص هذه الطريقة في مقارنة ثلاث عوامل رئيسية وهي : - الخصائص التقنية للجهاز . - مدى تدعيم الشركة البائعة للجهاز . - تكاليف الجهاز . وتشمل الخصائص التقنية تركيب الحاسب الآلي والطرفيات والطابعات والبرامج بكافة انواعها . اما العوامل التي تحدد مدى تدعيم الشركة البائعة فهي مستوى الصيانة ومدى تعاون الشركة البائعة وقدرة تحمل الاجهزة وفترة الضمان . وتتضمن التكاليف اسعار الاجهزة والتخفيضات وتكاليف الصيانة وتقدير نسبة السعر / القدرة . ان اختيار جهاز حاسب آلي مصغر على اسس علمية يضمن للمشتري الحصول على جهاز بالمواصفات المطلوبة باقل تكاليف . كما انه يقلل من الصعوبات والمشاكل التي قد تواجه المشتري بعد شراء الجهاز .

مطب نسخ ورقية .
وفر المصدر المركز .

Appendix B: Sample of relevant document

1.
استخدام الحاسب الآلي
2,5,6,13,22,24,29, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105,174,162, 176, 185, 191,195,202,204,206,207,208,209,210, 211,212,213,214,215,216,217,218,219,220,221,222,223,225,226,227,228,229,230,231,234,235,237.

2.
استرجاع المعلومات
30,36, 79, 80, 81, 82, 89, 90, 91, 92, 93, 95, 96, 97, 98, 100, 105,106,121,136,144,145, 201,202,203,205,206,208,209,210, 213,214,215,221,222,224.

3.
الادارة و التخطيط
20,35,38, 71,72, 74, 79, 80, 97, 98,174,175, 204,205,206,210, 211,213,214,217,222.

4.
التدريب و التعليم
5,14,16,18, 79, 82, 86, 94,162,164,175, 176, 177, 179, 184,210, 234.

5.
الترميز و التشفير
44,50, 74, 81, 88, 89, 96, 97,148,151,156,158,174, 194,201, 214,219,220,221,235.

6.
التعليم بمساعدة الحاسب
5, 16, 77, 81, 82, 93, 94,162,164,169,172,175, 176, 177, 178, 180, 183, 184, 185, 190, 194,198, 234.

7.
التعليم بواسطة الحاسب
5,16, 77, 81, 82, 93, 94,162,163,196,174, 176, 177, 185,195,198,204,205,207,208,209,210, 234.

8.
الحاسب الآلي
3, 19, 24, 28, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 52, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 108, 111, 123, 128, 134, 143, 144, 156, 159, 161, 176, 177, 178, 179, 180, 181, 183, 184, 185, 186, 187, 188, 189, 190, 191, 192, 193, 194, 204, 206, 207, 208, 209, 210, 211,212,213,214,215,216,217,218,219,220,221,224,225,226,227,228,229,230,231,234,235,237.

9.
الحاسبات الصغيرة
3,45, 73, 75, 76, 77, 78, 89, 90, 100, 101, 102, 103, 104, 105,159, 177, 178, 179, 180, 181, 183, 184, 185, 186, 187, 188, 189, 190, 191, 192, 193, 194, 195, 196, 197, 198, 199, 200, 201, 202, 203, 204, 205, 206, 207, 208, 209, 210, 216.

10.
الحاسبات المتناهية الصغر
75, 76, 77, 78, 89, 100, 101, 102, 103, 104, 105, 177, 178, 179, 180, 181, 183, 184, 185, 186, 187, 188, 189, 190, 191, 192, 193, 194, 195, 196, 197, 198, 199, 200, 201, 202, 203, 204, 205, 206, 207, 208, 209, 210, 216.

30,36,74, 80, 90, 100, 103,130,157,170,174, 178,181,184,185,187,198,207, 211,212,213,214,215,222.

11.
قواعد المعلومات
5,43, 79, 80, 81, 91, 97, 98,130,136,144,145,153,170, 177,178,180,182,185,186,187,189,191,198,202,208,209,210, 213,214,215,217,222,224.

12.
لغة برمجة عربية
2,4,15, 84, 85, 86, 87, 88, 90, 94, 101, 102, 103,104, 105,108,113,143,117,170, 176, 177, 178, 179, 180, 181, 182, 183, 184, 185, 186, 187, 188, 189, 190, 191, 192, 193, 194,195,196.

13.
مجتمع المعلوماتية
5,6, 10, 11, 12, 18,33, 84, 85, 103, 145, 164, 174, 177, 178, 201, 202, 212, 237.

14.
محاكاة الحاسب الالي
21, 23, 25, 26, 27, 31, 52, 100, 144, 176, 177, 178, 179, 180, 181, 182, 183, 184, 185, 186, 187, 188, 189, 190, 191, 192, 193, 194, 200, 204, 206, 207, 208, 209, 210.

15.
مهارات استخدام الكمبيوتر
71,78, 79, 177, 181, 188, 192, 198, 200, 205, 208, 237.

16.
نظم الخبرة
26,43, 77, 78, 79, 82, 89, 90, 142,174, 188, 194, 200, 201, 202, 203, 217,238.

17.
نظم المعلومات
13, 32, 48, 74, 76, 77, 78, 79, 80, 81, 82, 89, 90, 91, 92, 95, 96, 97, 98, 103, 105, 139, 170, 177, 178, 180, 183, 185, 186, 187, 189, 191, 194, 200, 201, 202, 203, 212, 214, 215, 221, 222, 224, 231.

18.
هندسة البرامج
71, 72, 76, 77, 84, 85, 94, 99, 101, 103, 178, 179, 180, 181, 182, 183, 184, 185, 186, 203, 207, 209, 214, 215, 220.

19.
هندسة الحاسب الالي
4,14, 24,47, 53,57, 72, 75, 93, 103,126, 178, 179, 180, 181, 182, 183, 184, 185, 186, 203, 204, 206, 207, 208, 209, 210, 211, 216, 218, 219, 230,240.

20.
هندسة الحاسوب
14, 24, 53, 71, 72, 75, 77, 93, 103, 156, 178, 179, 180, 181, 182, 183, 184, 185, 186, 203.
End

Appendix C: Sample C# Code

LM:

```
   private void caculateQuerySimilarity()
        {
            foreach (DocumentTerm document in listDocument)
            {
                document.calculateLanguageSimilirity();
            }
        }


        public List<DocumentTerm> rankingDocument()
        {
            List<DocumentTerm> sortedDocs = new List<DocumentTerm>();
            IOrderedEnumerable<DocumentTerm> sortedCollection;

            sortedCollection = from k in listDocument orderby
k.LanguageSimilirity descending select k;
            foreach (DocumentTerm item in sortedCollection)
            {
                sortedDocs.Add(item);
            }

    return sortedDocs;


private void calculateRiskFunction()
        {
            foreach (DocumentTerm document in listDocument)
            {
                document.TermMean = termMean;
                document.calculateRiskFunction();
```

EBM:

```
 private void calculateBinaryWeight()
        {
            string[] sparse = {" "};
            string[] parseQuery = qd.BoolQuery.Split(sparse,
StringSplitOptions.RemoveEmptyEntries);
            List<string> lQuery = new List<string>();
            foreach (string item in parseQuery)
            {
                lQuery.Add(item);
            }

            foreach (DocumentTerm document in listDocument)
            {
                document.BooleanQuery = lQuery;
                document.calculateBooleanWeight();
            }
        }
```

VSM

```
private void calculateCosineSimiliraty()
        {
            foreach (DocumentTerm document in listDocument)
            {
                double cosineSimiliraty = document.DotProduct /
qd.DocumentVectorLine * document.DocumentVectorLine;
                document.CosineSimiliraty = cosineSimiliraty;
            }
        }


        public List<DocumentTerm> rankingDocument()
        {
            List<DocumentTerm> sortedDocs = new List<DocumentTerm>();
            IOrderedEnumerable<DocumentTerm> sortedCollection;

            sortedCollection = from k in listDocument orderby
k.CosineSimiliraty  descending select k;
            foreach (DocumentTerm item in sortedCollection)
            {
                sortedDocs.Add(item);
            }
            return sortedDocs;
```