



Search Engine Optimization for E-Business

Website

محرك البحث الامثل لمواقع الاعمال الالكترونية

Prepared by: Nancy Sharaf

Student ID: 401020098

Supervised by: Dr. Raed Hnanandeh

A THESIS PROPOSAL SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENTS FOR THE DEGREE OF

Master OF Electronic-Business

Faculty of Business

E-Business Department

Middle East University

Amman-Jordan

January, 2013

تفويض

أنا نانسى شفيق محمد سهلى شرف أفوض جامعة الشرق الأوسط بتزويد نسخ رسالتي
المعنونة بـ " محرك البحث الأمثل لمواقع الأعمال الالكترونية" للمكتبات الجامعية أو
المؤسسات أو الهيئات أو الأشخاص المعنيين بالأبحاث والدراسات العملية عند طلبها.

الاسم: نانسى شرف

التوقيع:

التاريخ:


16 / 05 / 13

Middle East University for Graduate Studies

Authorization Form

I ,The Undersigned (*Nancy Shraf*), authorize the Middle East University for Graduate Studies to provide copies of my thesis to all and any university libraries and/or institutions or related parties interested in scientific researches upon their request


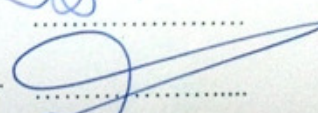
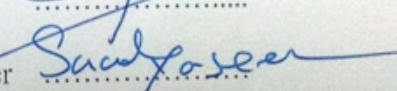
Name : Nancy Shafiq Sharaf

Signature : 

Date : May 16, 2013

Discussion Committee Decision

This thesis has been discussed under the title Search Engine Optimization for E-Business Website "This is to certify that the Thesis entitled was successfully defended and approved May 24th 2010.

Examination Committee Members		Signature
1- Dr. Raed Hanandah	Supervisor	
2- Dr. Soud Al-Mahameid	Internal Member	
3- Dr. Saed Ghaleb	External Member	

ACKNOWLEDGEMENTS

This thesis is the product of an educational experience at MEU, various people have contributed towards its completion at different stages, either directly or indirectly, and any attempt to thank all of them is bound to fall short.

To begin, I would like to express my whole hearted and sincere gratitude to Dr. Raed Hanandeh and Dr Sharef Jad for their guidance, time, and patience, for supporting me and this thesis during every stage of its development.

I would like to extend my special thanks to my family and my work family, without whose encouragement and support; I wouldn't have been here completing my degree's final requirements.

Sincerely Yours,

DEDICATIONS

To

My father and mother soul

My Brothers and sisters

And to all my friends

I dedicate this effort.

Sincerely Yours,

Nancy Sharaf

Table of Contents

Subject	Page
Authorization	II
Discussion Committee Decision	III
Acknowledgements	IV
Dedication	V
Table of contents	VI
List of Table	VIII
List of Figure	VIII
Abstract	IX
Chapter one General Framework	
Introduction	2
Study Problem	4
Significance of the Study	6
Objectives of the Study	7
Model and Methodology	8
Study Limitations	11
Research Importance	12
Chapter Two Previous Studies	
Introduction	15
E-Business	16
Search engine optimization	18
Information Retrieval	21
Previous Studies	23
Study contribution to Knowledge	35
Project Plan	4
Chapter Three Method and Procedures	

Introduction	48
Study Methodology	49
Study Design	49
Implementation	61
Study Population and Sample	81
Study Tools and Data Collection	82
Reliability and Validity	87
Chapter Four Analysis of Results & Tests experiment	
Introduction	91
Descriptive analysis of design model	91
Study Tests	93
Chapter Five Results Discussion	
Results	100
Conclusions	101
Recommendations	102
References	

LIST OF Table

NO.	Subject	Page
Chp2 (2-1)	Organic Ranking Visibility (PRWeb, 2011)	23
Chp2 (2-2)	Project Plane	45
Chp2 (2-3)	Risks Analysis	46

LIST OF Figure

No.	Subject	Page
(1-1)	General search engine architecture . (Xiao-Baili and Sumit Sarkar 2012)	9
(1-2)	Simple context diagram for search engine ranking	11
(2-1)	SEARCH ENGINE OPTIMIZATION	21
(2-2)	Context diagram	25
(2-3)	Gantt chart	47
(2-4)	Project Plane	48
(3-1)	Crawler ERD	51
(3-2)	Indexer ERD	56
(3-2)	Flow Chart Diagram	61

Search Engine Optimization for E-Business Website

Supervised by: Dr. Raed Hnanandeh

Prepared by: Nancy Sharaf

ABSTRACT:

Since the 80's of the last century, the emergence of IT advancements has been regarded as the first motivated aspect for developing a new type of systems called information systems (abbreviated as IS). These systems aim at providing multiple services for multiple fields .

Variety number of information systems has been developed to react to several problematic issues; the aims of these systems are to reduce the amount of time required to achieve certain tasks and to decrease the number of errors that might be fallen into during performing the tasks in manual system.

The most conventional systems being introduced are those systems that are related to business and economical activities. Banking systems, inventory systems and hospital systems are instances of information systems that provide means of system analytical tools.

Search engine optimization of web pages techniques to provide a more significant and restricted set of pages, keywords, and engine query by using internet as the final result of a user search. The proposed system is a web site helps his users who need to search for Arabic topics on the internet, the Arabic search engine is an information system that will give those people the ability to enter Arabic queries and retrieve results not based on exact words, it'll filter out unnecessary words, and rank the retrieved URLs based on their relevancy. The systems allow users to search through a simple graphical user interface GUI which is located on the web site. In the implementation we try to produce our system to be considered as an open system, this system is maintainable, dependable, efficient and usable.

محرك البحث الامثل لمواقع الاعمال الالكترونية

اعداد الطالبة : نانسي شرف

اشراف الدكتور : رائد الهنادة

الملخص

في الثمانينيات من القرن الماضي بدأ ظهور تكنولوجيا المعلومات , واصبح يعتبر كدافع مهم لتطوير نوع جديد من الانظمة تسمى انظمة المعلومات , وتهدف هذه الانظمة الى تقديم خدمات عديدة في مجالات مختلفة. تطورت العديد من انظمة المعلومات لتكون كحل للعديد من القضايا الاشكالية مثل تقليل الوقت المطلوب لتحقيق المهام وايضا تقليل الاخطاء الممكن الوقوع بها عند تنفيذ هذه المهام في الانظمة التقليدية او النظام اليدوي. هناك العديد من الانظمة المعلوماتية كأنظمة الاعمالوالانشطة الاقتصادية والانظمة المصرفية وانظمة الجرد والمستشفيات وتميزت بوجود ادوات تحليلية للنظام. واحد اشكال انظمة المعلومات هو محرك البحث لصفحات الوب والذي يهدف الى تزويد مجموعة من الصفحات الاكثر اهمية والتي يبحث عنها المستخدم .

و النظام المقترح ضمن هذه الدراسة هو موقع الكتروني يبحث عن استعلامات المستخدم باللغة العربية ويسترجع مجموعة من المواقع مرتبة من الاكثر اهمية وعلاقة مع الشيء المستعلم عنه وذلك بازالة الكلمات غير المهمة من كلمات البحث. ويطبق هذا النظام على واجهة مستخدم سهلة تنفذ البرنامج كنظام مفتوح قابل للتعديل والاضافات وليس كالنظام المغلق مثل نظام القران الكريم. ومن خصائص هذا النظام انه يعمل بطريقة فعالة وسهل الصيانة والاستخدام .

CHAPTER ONE

General Framework

- (1-1): Introduction
- (1-2): Study Problem
- (1-3): Significance of the Study
- (1-4): Objectives of the Study
- (1-5): Model and Methodology
- (1-6): Study Limitations
- (1-7): Research Importance

(1-1): Introduction

E-business is the conducting of business on the internet not only buying and selling but also serving customer and collaborating with other business partners, the e-business is widely applied , E-business is the better to meet the requirements of the users by using Search Engine Optimization (SEO) which is the process of affecting the visibility of a website, which makes the e-business approachable to a good number of possible and existing customers, it will increase the revenue. (T. Fagni, 2006).

E-business differs from a physical business in a lot of ways. Even people who are good businessmen in the "real world" might not understand anything about an e-business. It starts with the way people find you. For example: When you have a physical business, people can see you when they are shopping or when they are on their way home from work, but if you have an e-business people don't just bump into you! In addition to that, online-marketing offers a lot of new facilities, which don't exist in offline-marketing.

Each E-Business website poses unique challenges and therefore requires a customized search engine optimization strategy to get best results. The varying dynamics of our research segment (competition, business focus, product offerings, pricing, brand positioning, target audience, geographical location and service reach, depth of website content and the promotional budgets make it impossible to devise a 'one size fits all' search engine optimization is the solution. (Fragfornt, 2006).

Many great Search Engine are perfectly working on the World Wide Web, though the searching process seems to be working like a smooth machine, beyond the simple query

request and results retrieval, there's a big world of operations run behind the scenes (T. Fagni, 2006).

One of these operations is done prior to building the huge search engine database, which is indexing. indexing the documents means building structures that allow the search process to become much faster, plus it makes ranking the results much easier process to, but my concern here is about the way indexers work; indexers don't leave the documents as they're retrieved, it takes the important words to get them to their original forms, this makes it possible to retrieve documents which contains any form of the words in the search criteria or query,

Search engine ranking refers to the position at which a particular site appears in the results of a search engine query. (Fragfornet, 2006)

A site is said to have a high ranking when it appears at or near the top of the results list, The webpage listed at the top of the results page has been selected by the search engine as the most likely to provide the content the user is seeking . i.e. Andrieu(2009)

The webpage listed at the top of the results page has been selected by the search engine as the most likely to provide the content the user is seeking (Leuski, 2001; Zamir, , Etzioni & Madani, 1997)

Many information seekers use a search engine to begin their Web activity. In this case, users submit a query, typically a list of keywords, and receive a list of Web pages that may be relevant, typically pages that contain the keywords. In this text we discuss the building good search engines, and describe some of the techniques that are useful.

Many of the search engines use well-known information retrieval (IR) algorithms and techniques. However, IR algorithms were developed for relatively small and coherent collections, on the other hand, are massive, much less coherent, changes more rapidly, and are spread over geographically distributed computers. This requires new techniques, or extensions to the old ones, to deal with the gathering of the information, to make index structures scalable and efficiently updateable, and to improve the discriminating ability of search engines. For the last item, discriminating ability, it is possible to exploit the linkage among Web pages to better identify the truly relevant pages. (Salton 1989; Faloutsos 1985)

For example, in a study of over half a million pages over 4 months, it was found that about 23% of pages changed daily. In the .com domain 40% of the pages changed daily, and the half-life of pages is about 10 days (in 10 days half of the pages are gone, i.e., their URLs are no longer valid.) (Cho and Garcia-Molina 2000)

(1-2): Study Problem

Existing web search engines often calculate the relevancy of web pages for a given query by counting the search keywords contained in the web pages. This method works well when users' queries are clear and specific. However, in real world, web search queries are often short and ambiguous and containing a lot of stop words , and web pages contain a lot diverse and noisy information. These will very likely lead to the

deteriorating of the performance of web search engines, due to the gap between query space and document. (Berman and Katona, 2010)

Accessing topical information through existing search engines requires the formulation of appropriate queries, which is highly challenging, and then the appropriate selection of query is an optimizing problem and the purpose is to obtain the best query to get the information through the web automatically (Joachims, 2003).

However, retrieving sufficient and relevant information online is difficult for many people because they may not be familiar with the search context, they use too few keywords to search or use improper keywords, or search engines are not able to receive the users' real meaning through their given keywords (Bernard, 1998).

Another problem faces many of search engine users is that due to the huge growth of www there are a lot of information available on the internet, when the search engine is searching for the required information, the search engine returns many of web pages related to the search query. The user does not have time to check all of the returned results, and then he can check a few numbers of these web pages and ignore the others. However the required information meet the query might be found in these pages which ignored. As a result, users cannot find the specific information they really want. (Nicolas, 2000).

The internet delivers way too much information about the user query and people don't know anymore which information is relevant and which not according to huge list of search result when multiple sub topics of the given query are mixed together or when the search engine search for low significant words so this is a time consuming the problem

arises when we deal with Arabic documents , which doesn't have much concern of big search companies , because the lack of knowledge in the Arabic language .

The study problem

- Many of current search engine not very interested in the Arabic languages because no existence of the rooting algorithm.
- Most search engine wasting time in the search for the low scientific words.
- Also many search engines follow the derived word in storing words into database which take more space in database.

(1-3): Significance of the Study

Delivering high-valuable information is by far the cheapest way to get people into your e-business (T. Fagni, 2006). And that is why search engines play such an important role. The significance of the current study arises from the important role of Ranking in Arabic Search Engine. Also the significance of the current study demonstrated from three dimensions:

Our project comes to solve these problems which we mentioned before that faced the current system:

- ✓ This proposed search engine using a good stemming or rooting algorithm for Arabic languages.
- ✓ This proposed search engine doesn't search for stop word list so exploited all time in useful search.

- ✓ Follow the rooting method when storing words into database.

(1-4): Objectives of the Study

The sample study of our research based on creating search engine optimization program includes optimization for indexing and rooting algorithm for Arabic search engine, by using stemming and good rooting algorithm to return useful result list to the user meet his requirement need of information.

This project proposes a new information system that is deployed on the internet. The designed system has to accomplish the following objectives:

1. Crawl the web and get Arabic documents
2. Index these documents into the index file
3. Take in consideration a good Arabic stemming and rooting algorithms
4. Allow user to search the web using simple web application interface.
5. When finding query results, the system shall use vector model structure to calculate similarity of retrieved documents and rank those documents considering their relevancy to the input query.

control module may be implemented by the crawlers themselves.) The crawlers also pass the retrieved pages into a page repository. Crawlers continue visiting the Web, until local resources, such as storage, are exhausted. This basic algorithm is modified in many variations that give search engines different levels of coverage or topic bias. Once the search engine has been through at least one complete crawling cycle, the crawl control module may be informed by several indexes that were created during the earlier crawl(s). The crawl control module may, for example, use a previous crawl's link graph (the structure index in Figure 1.1) to decide which links the crawlers should explore, and which links they should ignore.

The indexer module extracts all the words from each page, and records the URL where each word occurred. The result is a generally very large "lookup table" that can provide all the URLs that point to pages where a given word occurs (the text index in Figure 1.1). The table is of course limited to the pages that were covered in the crawling process. As mentioned earlier, text indexing of the Web poses special difficulties, due to its size, and its rapid rate of change. In addition to these quantitative challenges, the Web calls for some special, less common kinds of indexes. For example, the indexing module may also create a structure index, which reflects the links between pages. Such indexes would not be appropriate for traditional text collections that do not contain links.

The collection analysis module is responsible for creating a variety of other indexes. The utility index in Figure 1 is created by the collection analysis module. For example, utility indexes may provide access to pages of a given length, pages of a certain "importance," or pages with some number of images in them. The collection analysis

module may use the text and structure indexes when creating utility indexes. During a crawling and indexing run, search engines must store the pages they retrieve from the Web.

The page repository in 1.1 represents this—possibly temporary—collection. Sometimes search engines maintain a cache of the pages they have visited beyond the time required to build the index. This cache allows them to serve out result pages very quickly, in addition to providing basic search facilities. Some systems, such as the Internet Archive, have aimed to maintain a very large number of pages for permanent archival purposes. Storage at such a scale again requires special consideration.

The query engine module is responsible for receiving and filling search requests from users. The engine relies heavily on the indexes, and sometimes on the page repository. Because of the Web's size, and the fact that users typically only enter one or two keywords, result sets are usually very large.

The ranking module therefore has the task of sorting the results such that results near the top are the most likely ones to be what the user is looking for. System will introduce search algorithms that take advantage of the Web's interlinked nature. When deployed in conjunction with the traditional IR techniques, these algorithms scientifically improve retrieval precision in Web search scenarios. The crawler, (S. Brin, 2008)

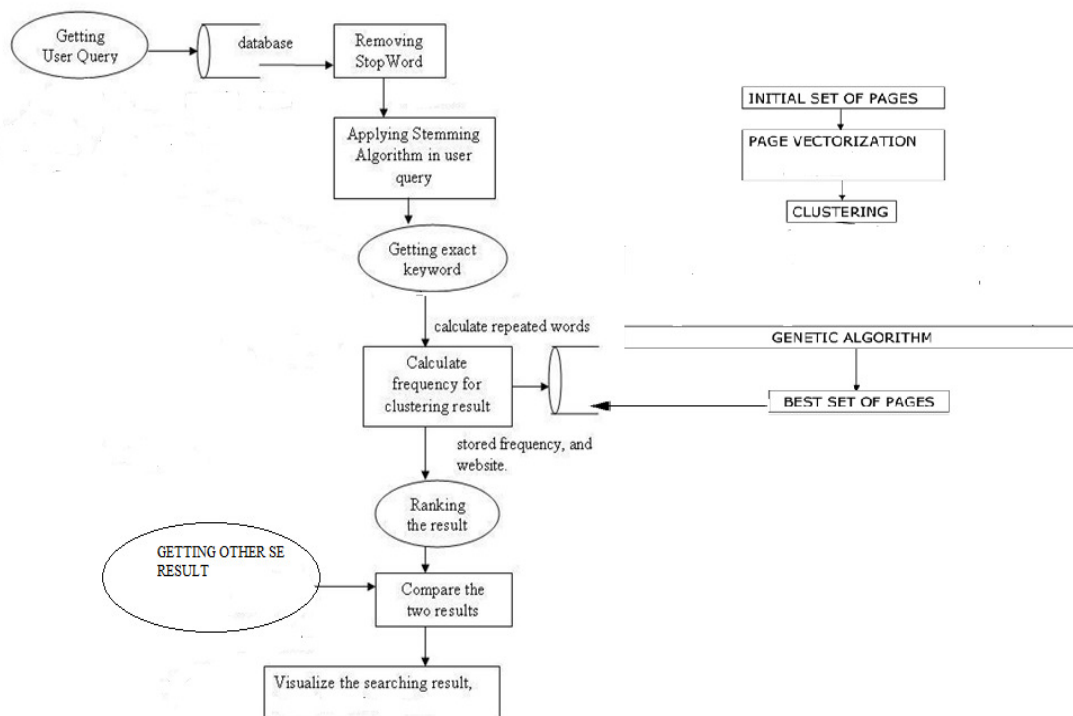


Figure (1-2): Simple context diagram for search engine ranking

(1-6): Study Limitations

Human Limitations: the culture awareness regarding the current study includes Ranking in Search Engine to Optimize E-Business Website by Using Data Mining.

Place Limitations: Web engine

Time Limitations: The time absorbed to study accomplishment.

Scientific Limitations: technologies measuring the ranking In Search Engine to Optimize E-Business Website by Using Data Mining (Isik, 2010).

(1-7): Research Importance:

In our proposed system we will have devised a flexible ranking search engine optimization model explain how to customize it to meet individual needs of information. Sometimes, the users wish to start at a basic search engine optimization level and prefer to increase the intensity as they begin to reap the benefits of enhanced ranking, traffic and revenues. Other times, they choose to go full steam right from the start not only to get bigger results early but also to corner a larger market share and lead the competition. The proposed system overcome problems for " Arabic search engine" by selecting a small subset from the search results which have high scores and semantically related to the user query which are different from each other and are chosen from different regions of some topics where the pages are represented.

In this proposed system will be to reduce the results in a short list by applying artificial intelligence techniques such as genetic algorithm. the existing search engine such as Google , yahoo often return a long list of search results ranked by their relevancies to the given query so the users have to go through the list and examine the titles to identify their required result (hHua-jun zeng learning to cluster web search results 2008),

Our search engine optimization program is structured into several processes and techniques. Each process is designed to match extensive research on search engine behavior and search engine optimization industry. The option to add-on modular services mentioned below will help us to customize the search engine optimization roadmap based on campaign objectives.

- Crawl the web and get Arabic documents
- Index these documents into the index file
- Take in consideration a good Arabic stemming and rooting algorithms
- Allow user to search the web using simple web application interface.
- When finding query results, the system shall use vector model structure to calculate similarity of retrieved documents and rank those documents considering their relevancy to the input query.

CHAPTER TWO

Previous Studies

(2-1): Introduction

(2-2): E- Business

(2-3): Search engine optimization

(2-4): Information Retrieval

(2-5): Previous Studies

(2-6): Study contribution to Knowledge

(2-7): Project Plan

(2-1): Introduction

Many information seekers use a search engine to begin their Web activity. In this case, users submit a query, typically a list of keywords, and receive a list of Web pages that may be relevant, typically pages that contain the keywords. There is no question that the Web is huge and challenging to deal with. Several studies have estimated the size of the Web and while they report slightly different numbers, most of them agree that over a billion pages are available. Given that the average size of a Web page is around 5–10K bytes, just the textual data amounts to at least tens of terabytes. The growth rate of the Web is even more dramatic. According to the size of the Web has doubled in less than two years, and this growth rate is projected to continue for the next two years. Aside from these newly created pages, the existing pages are continuously updated. (intelligent information processing and web mining, Mieczyslaw 2004) .

(2-2): E- Business

According to Chaffey (2002), e-business is described as “all the electronically mediated information exchanges, both within an organization and with external stakeholders, supporting the range of business processes.”

Boone and Ganeshan (2007) define e-Business technologies as “the use of Internet or any digitally enabled inter- or intra- organizational information technology to accomplish business processes”.

Electronic business (e-business) using web sites and enable companies to link their internal and external data processing systems more efficiently to better satisfy the needs of the customers-business is sometimes conceived only as internet stores, is an approach to conducting business on the internet. These activities are referred to e-commerce.

E-business has a larger sense. Many other activities, whose aims are support and effectiveness increasing of business processes, come under e-business.

E-business presents all enterprise activities that are supported of information systems and information communication technologies (ICT) by using web sites for each company to indicate the address so each company like to reach to the optimum ranking over the internet. E-business presents alternative, which has address specific place nowadays and significantly supports profitability and company strategy towards competitive advantage production and business subjects. We can obtain so many

interests in electronic business from the increasing numbers of business transactions all over the world in different sectors; E-business is sometimes conceived only as internet web sites. Many other activities, whose aims to support and effectiveness increasing of business processes, come under e-business. E-business presents all enterprise strategies and activities that are supported ranking search engine optimization based on the availability of digital communication. (Boone and Ganeshan (2007))

(2-2-1): Impact of e-Business on websites Markets

As any other disruptive technology e-Business has affected financial markets in a variety of ways. An important characteristic that Bev (2008) examines is Disintermediation, which refers to the non-existence of intermediaries in a supply chain. Internet allows full disintermediation because of the market's transparency. Disintermediation is also examined by Evans and Wurster (2000). According to their work, the new form of disintermediation that was developed along with the Information Technology, "allows for the traditional richness curve to be displaced, allowing new players to offer greater reach and greater richness simultaneously".

E-Businesses are created, which aim to play exactly the role of the intermediate between the manufacturers and the customers. This is called reinter mediation. In his conclusion, Bev (2008) mentions some of the advantages and the disadvantages of disinter mediated channels. The main advantages are reduction to search costs,. Although Bev (2008) slightly mentions some of them, a more careful and thorough research should point out that one of the most important challenges of these channels is to create a safe business transactional environment. Being the intermediate between the producer and the

customer involves many security risks, which should be thought in depth, before building an e-Business.

(2-3): Search engine optimization

Search engine optimization refers to all actions that are appropriate to achieve a better position in the editorial search engine results page (Lammanett, 2006). Spent time, money, and effort developing an enticing website, but that don't necessarily mean that you'll attract visitors. On the other hand, improving your site's ranking in search engine results will definitely enrich your web position!

2.3.1 Search Engines Handle Arabic Queries

The performances of general and Arabic search engines were compared based on their ability to retrieve morphologically related Arabic terms. The findings highlight the importance of making users aware of what they miss by using the general engines, underscoring the need to modify these engines to better handle Arabic queries.

(Haidar Moukdad School of Library and Information Studies, Dalhousie University Halifax, Nova Scotia 2008)

Information retrieval, as a language-dependent operation, is greatly affected by the language of documents and how a search engine handles the characteristics of this language. Linguistic characteristics that typically have impact on the accuracy and relevancy of Web searches are mainly related to the morphological structures of words and to morphological word variants. Thus it is not surprising that the most common

linguistic features provided by search engines are automatic stemming (conflation of morphologically related words) and truncation. While the positive effect of stemming on English information retrieval has yet to be empirically proven (Harman, 1991 and Hull 1996), languages with more complex morphology are more susceptible to the advantages of stemming (Popović and Willett, 1992 and Savoy, 1991). As opposed to Arabic and other morphologically complex languages, the English language has morphological rules that can be easily treated in computational and information retrieval environments.

English words tend to be formed on the basis of a limited and relatively straightforward number of rules, allowing for simple stemming rules in order to retrieve variants of search terms. Conversely, Arabic has a large number of rules that makes retrieving word variants a challenging task and, consequently, stemming and other techniques absolute necessities

2.3.2 How Search Engines work

"A search engine is a web application designed to hunt for specific keywords and group them according to relevance." (Clay, Esparza, 2009) To get a better understanding on how to optimize a website for search engines, it is helpful to first understand how search engines actually work. The thesis doesn't want to go into any details of this, but it wants to give a quick overview.

2.3.3 Components of a Search Engine

Every search engine consists of three major elements:

- Crawler
- Index
- Ranking
- Search Engine Software

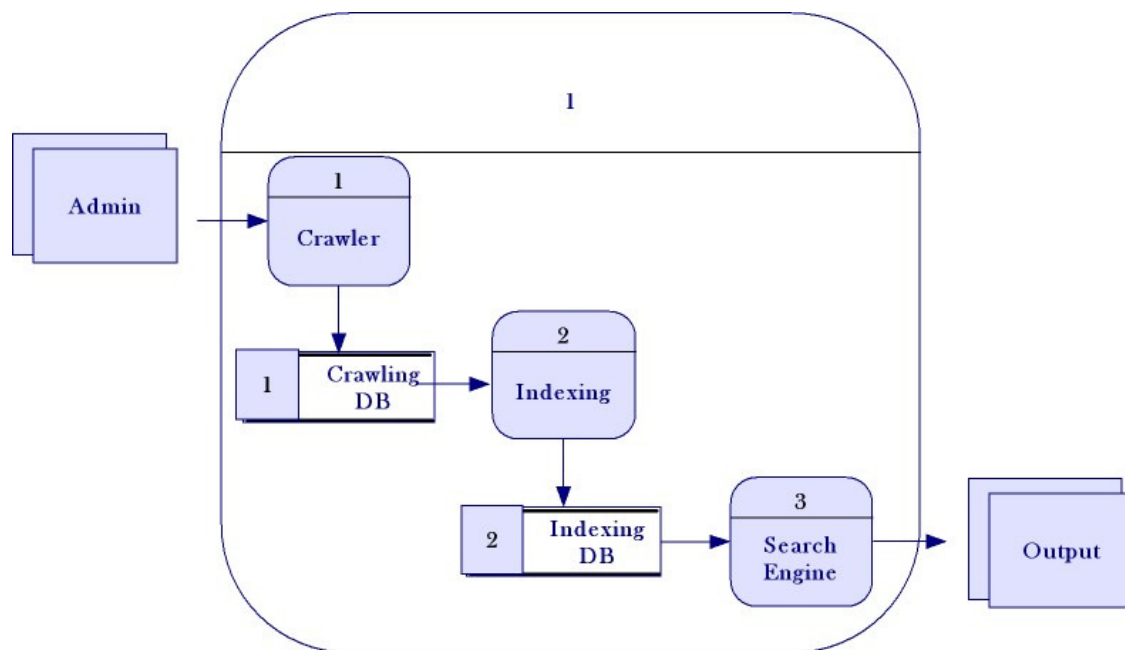


FIGURE (2-1) SEARCH ENGINE OPTIMIZATION

A crawler is also called spider. To get into the search engines' results, a website first needs to be "spidered". That means that a crawler visits the site to get information about it. A crawler doesn't see the website like a human visitor though, which needs to be considered when you optimize a website for search engines. (Alby, Karzauninkat 2007, p.21) Everything the crawler finds goes into the second part, the index. The index is a

huge collection of data, which contains a copy of every webpage that was spidered. A crawler needs to visit the same website over and over again, to notice possible changes and to update the index. Then the search engine software tries to find the most relevant matches to a search and puts them into a ranking. (Search Engine Watch 2010) According to (Baeza-Yates, Ribeiro-Neto 2011, p.468) ranking is the hardest and most important function search engines have to execute. There are two main challenges they have to face:

2.4 Information Retrieval

"Web search has its root in information retrieval (or IR for short); a field of study that helps the user find needed information from a large collection of text documents." (Liu, 2007, p.183) Information retrieval is more complicated than data retrieval in databases, because data in databases are ordered in relational tables, while the websites are unstructured. Another difference is, that websites have hyperlinks, and those are really important for the ranking. (Liu 2007, p.183) .

While a keyword query is pretty simple, a natural language query can get very complex. (Liu, 2007, p.185) One important goal of information retrieval research is to develop models, which optimize the rate of relevant information for a search. (Leibniz Gemeinschaft, 2012) Due to technical restrictions, information retrieval systems cannot

analyze any type of file. This needs to be considered, when doing a search engine optimization. (Bischopinck , Ceyy 2009, p.190)

Unless you are a household name like eBay or Amazon, chances are rare that people will simply type in your URL. The biggest traffic is coming from the search engines. And they are getting more and more powerful. (Clay, Esparza 2009, p.10) Since search engines are often the place, where people enter the World Wide Web, no one can ignore them. Only slightly more than ten percent of all search engine users click on the second page of search results. That is why especially the first ten entries are highly competitive. Only those who are visible are able to guide the user on their site. [Alby, Karzauninkat 2007, p.4) This is also confirmed by a study from Nielsen, NetRatings at 2005. It says that for 79 percent of all internet users, search engines are the most important starting point for a purchase decision. (Lammanett, 2006, p.146) .

Organic Ranking Visibility	
(shown in a percentage of participants looking at a listing in this location)	
Rank 1	- 100%
Rank 2	- 100%
Rank 3	- 100%
Rank 4	- 85%
Rank 5	- 60%
Rank 6	- 50%
Rank 7	- 50%
Rank 8	- 30%
Rank 9	- 30%
Rank 10	- 20%

Table(2-1): Organic Ranking Visibility (PRWeb, 2011)

It shows which areas of the search engines are most likely to be looked at. The first three ranks of the SERP are being read by 100% of all searchers. But rank 10 is only read by 20% of all searchers. (Lammanett, 2006, p.146) All this shows, that search engine optimization has an economic relevance.

(2-5): Previous studies

HaidarMoukdad 2007 under title “How Do Search Engines Handle Arabic Queries”

Presents the search engines and tools. However, the use and spread of other languages are by no means negligible, a fact that contributes to the complexity and importance of investigating information retrieval on the Web. General search engines on the Web are the most popular tools to search for, locate, and retrieve information, and their use has been growing at exponential rates. These engines handle English queries more or less in the same way, but their handling of non-English queries is greatly different from how these queries are handled by non-English search engines—engines that were designed for specific languages. Most general search engines like AltaVista, AlltheWeb, and Google, allow users to limit their searches to specific languages, and some of them even provide local versions, including fully functional interfaces, to better accommodate the information needs of different regional and linguistic groups. Searchers for non-English documents either use the general search

engines or smaller engines specifically designed to handle queries in their respective languages. How the general search engines handle non-English queries is an area that has been largely neglected by research on information retrieval on the Web. The neglect is even more apparent in research on non-Western languages, among which is Arabic, the language covered in this paper.

Bar-Ilan, J. and T. Gutman 2003 under title “**How do search engines handle non-English queries**”. Presents Information retrieval, as a language-dependent operation, is greatly affected by the language of documents and how a search engine handles the characteristics of this language. Linguistic characteristics that typically have impact on the accuracy and relevancy of Web searches are mainly related to the morphological structures of words and to morphological word variants. Thus it is not surprising that the most common linguistic features provided by search engines are automatic stemming (conflation of morphologically related words) and truncation. While the positive effect of stemming on English information retrieval has yet to be empirically proven (Harman, 1991 and Hull 1996), languages with more complex morphology are more susceptible to the advantages of stemming (Popović and Willett, 1992 and Savoy, 1991). As opposed to Arabic and other morphologically complex languages, the English language has morphological rules that can be easily treated in computational and information retrieval environments. English words tend to be formed on the basis of a limited and relatively straightforward number of rules, allowing for simple stemming rules in order to retrieve variants of search terms. Conversely, Arabic has a large number of rules that makes

retrieving word variants a challenging task and, consequently, stemming and other techniques absolute necessities.

Arabic belongs to the Semitic family of languages, which includes Akkadian, Aramaic, Ethiopic, Hebrew, Phoenician, Syriac and Ugaritic. The Arabic script was derived from the Aramaic via the Nabatean cursive script (Hitti 1963). As is the case with all Semitic languages, the script is written from right to left, and this script has traditionally been represented in converted (Romanised) form in Western academic and computerized environments. As opposed to English and other Western languages, vowels have never become a permanent part of the Arabic writing system, allowing for the occurrence of many homonyms in documents. For example, the written word *Scr* could have any of the following meanings: to feel, poetry, hair or to crack. And the word *clm* could mean flag, science or to know. To handle the Arabic script on the Web, a number of encoding systems has been developed. The most common of these systems are Arabic (Windows), Arabic (ISMO 708), Arabic (DOS), and Arabic (ISO). In addition, Arabic is covered by the Unicode encoding system.

Junghoo Cho (1998) under title “Efficient Crawling Through URL Ordering “

Aimed to study in what order a crawler should visit the URLs it has seen, in order to obtain more "important" pages first. Obtaining important pages rapidly can be very useful when a crawler cannot visit the entire Web in a reasonable amount of time. We define several importance metrics, ordering schemes, and performance evaluation measures for this problem. We also experimentally evaluate the ordering schemes on the

Stanford University Web. Our results show that a crawler with a good ordering scheme can obtain important pages significantly faster than one without.

Andreas Paepcke (2000) under title **“Collaborative value filtering on the Web”** presents

Internet search engines help users locate information based on the textual similarity of a query and potential documents. Given the large number of documents available, the user often finds too many documents, and even if the textual similarity is high, in many cases the matching documents are not relevant or of interest. Our goal is to explore other ways to decide if documents are "of value" to the user, i.e., to perform what we call "value filtering." In particular, we would like to capture access information that may tell us—within limits of privacy concerns—which user groups are accessing what data, and how frequently. This information can then guide users, for example, helping identify information that is popular, or that may have helped others before. This is a type of collaborative filtering or community-based navigation. Access information can either be gathered by the servers that provide the information, or by the clients themselves. Tracing accesses at servers is simple, but often information providers are not willing to share this information. We therefore are exploring client-side gathering. Companies like Alexa are currently using client gathering in the large. We are studying client gathering at a much smaller scale, where a small community of users with shared interest collectively track their information accesses. For this, we have developed a proxy system called the Knowledge Sharing System (KSS) that monitors the behavior of a

community of users. Through this system we hope to: 1. Develop mechanisms for sharing browsing expertise among a community of users; and 2. Better understand the access patterns of a group of people with common interests, and develop good schemes for sharing this information.

Andreas Paepcke (2000) under title **Understanding Value-Based Search and Browsing Technologies**, presents search and ranking technologies that are based on document similarity measures. The increase of multimedia data within documents sharply exacerbates the shortcomings of these approaches. Recently, research prototypes and commercial experiments have added techniques that augment similarity-based search and ranking. These techniques rely on judgments about the 'value' of documents. Judgments are obtained directly from users, are derived by conjecture based on observations of user behavior, or are surmised from analyses of documents and collections. All these systems have been pursued independently, and no common understanding of the underlying processes has been presented. We survey existing value-based approaches, develop a reference architecture that helps compare the approaches, and categorize the constituent algorithms. We explain the options for collecting value metadata, and for using that metadata to improve search, ranking of results, and the enhancement of information browsing. Based on our survey and analysis, we then point to several open problems.

Krishna Bharat, George A. Mihaila (2000) under title : **Using Non-Affiliated Experts to Rank Popular Topics** , presents search engine returns a ranked list of documents. If the query is on a popular topic (i.e., it matches many documents) then the returned list is usually too long to view fully. Studies show that users usually look at only the top 10 to 20 results. However, the best targets for popular topics are usually linked to by enthusiasts in the same domain which can be exploited. In this paper, we propose a novel ranking scheme for popular topics that places the most authoritative pages on the query topic at the top of the ranking. Our algorithm operates on a special index of "expert documents." These are a subset of the pages on the WWW identified as directories of links to non-affiliated sources on specific topics. Results are ranked based on the match between the query and relevant descriptive text for hyperlinks on expert pages pointing to a given result page. We present a prototype search engine that implements our ranking scheme and discuss its performance. With a relatively small (2.5 million page) expert index, our algorithm was able to perform comparably on popular queries with the best of the mainstream search engines.

Jenny & Kevin & John (2001) under title **“an Adaptive Model for Optimizing Performance of an Incremental Web Crawler“** aimed to study crawlers have been written that distribute the load of crawling across a cluster, but they generally distribute the work in different ways. Due to the competitive nature of the Internet indexing and searching business, few details are available about the latest generation of crawlers. The first generation Google crawler is apparently designed as a batch crawler, and is only partially distributed. It uses a single point of control for scheduling of URLs to be

crawled. While this might appear convenient, it also provides a bottleneck for intelligent scheduling algorithms, since the scheduling of URLs to be crawled may potentially need to touch a large amount of data. Mercator supports incremental crawling using priority values on URLs and interleaving crawling new and old URLs. The scheduling mechanism of the Web Fountain crawler resembles Mercator in that it is fully distributed, very flexible, and can even be changed on the fly. This enables efficient use of all crawling processors and their underlying network. The base software component for determining the ordering on URLs to be crawled consists of a composition of sequencers. Sequencers are software objects that implement a few simple methods to determine the current backlog, whether there are any URLs available to be crawled, and control of loading and data structures to disk. Sequencers are then implemented according to different policies, including a simple FIFO queue or a priority queue. Other Sequencers are combiners, and implement a policy for joining sequencers. Examples include a round robin aggregator, or a priority aggregator that probabilistically selects from among several sequencers according to some weights.

Arvind Arasu, Junghoo (2001), under title “ **Searching the Web** “ presents

an overview of current Web search engine design. After introducing a generic search engine architecture, we examine each engine component in turn. We cover crawling, local Web page storage, indexing, and the use of link analysis for boosting search performance. The most common design and implementation techniques for each of these components are presented. We draw for this presentation from the literature, and from our own experimental search engine testbed. Emphasis is on introducing the

fundamental concepts, and the results of several performance analyses we conducted to compare different designs.

Sanjay Ghemawat, Howard Gobioff (2003) under title “**The Google File System**”, aimed to designed and implemented the Google File System, a scalable distributed file system for large distributed data-intensive applications. It provides fault tolerance while running on inexpensive commodity hardware, and it delivers high aggregate performance to a large number of clients. While sharing many of the same goals as previous distributed file systems, our design has been driven by observations of our application workloads and technological environment, both current and anticipated, that reflect a marked departure from some earlier file system assumptions. This has led us to reexamine traditional choices and explore radically different design points. The file system has successfully met our storage needs. It is widely deployed within Google as the storage platform for the generation and processing of data used by our service as well as research and development efforts that require large data sets. The largest cluster to date provides hundreds of terabytes of storage across thousands of disks on over a thousand machines, and it is concurrently accessed by hundreds of clients. In this paper, we present file system interface extensions designed to support distributed applications, discuss many aspects of our design, and report measurements from both micro-benchmarks and real world use.

Caramia & Felicib & Pezzolic (2003) under title “**Improving search results with data mining in a thematic search engine**“ presents the search context and the user profile are used to extract a finite set of significant words or page characteristics that are used to create, from all pages in P , a vector of characteristics (page vectorization). Such vectorization represents a particular way of “looking” at the page, specific of each context/profile. Second, the vectorized pages are analyzed by a clustering algorithm that partitions them into subsets of similar pages. This induces a two-dimensional ordering on the pages, as each page p can now be ordered according to the original score within its cluster. At this point the objective is to provide the user with a reduced list that takes into account the structure identified by the clusters and the original score function. This is done in the third step, where the pages that have higher score in each cluster are selected to compose an initial population that is then analyzed by a genetic algorithm. The initial population is formed by subsets of pages, called chromosomes in the genetic algorithms terminology, extracted in turn from the clusters, starting from the pages with higher score; this way the initial population is composed of pages with high score that have a high probability of being different from each other.

John Cuadrado, Maciej Ceglowski, J. Scott Payne (2003) under title **Patterns in Unstructured Data** presents search engines is a presentation suggesting several methods of improving search engine relevancy including latent semantic indexing and multi-dimensional scaling.

Chris Burges, Ted Shaked, Erin Renshaw (2005) under title “**Learning to Rank Using Gradient Descent**” aimed to investigate using gradient descent methods for learning ranking functions; we propose a simple probabilistic cost function and we introduce RankNet, an implementation of these ideas using a neural network to model the underlying ranking function. We present test results on toy data and on data from a commercial internet search engine.

Bev (2008) under title “**From Bertrand Market Expectation to Realistic Disintermediated Marketing in E- Commerce, Available**“ Present e-Business has affected financial markets in a variety of ways. An important characteristic that examines is Disintermediation, which refers to the non-existence of intermediaries in a supply chain. Internet allows full disintermediation because of the market’s transparency. According to their work, the new form of disintermediation that was developed along with the Information Technology, “allows for the traditional richness/reach curve to be displaced, allowing new players to offer greater reach and greater richness simultaneously”. E-Businesses are created, which aim to play exactly the role of the intermediate between the manufacturers and the customers . The main advantages are reduction to search costs, price dispersion, price transparency and price differentiation. Another thing to consider is some important disadvantages of disinter mediated channels. Although Bev (2008) slightly mentions some of them, a more careful and thorough research should point out that one of the most important challenges of these channels, is to create a safe business transactional environment. Being the intermediate between the producer and the customer involves many security risks, which should be thought of in depth, before building an e-Business.

Masoud (2011) under title “**a Multilayer Data Mining Approach to an Optimized E-business Analytics Framework**” appears from the preceding discussions and experimentations that the proposed multilayer data mining approach to an e-business framework may increase overall amount of business intelligence that an enterprise can gain. The concept contains a new methodology and its associated mining structures and mining models. The paper used this novel methodology and introduced an optimized framework called EBAF, to provide intelligence for SMEs and help them to gain competitive advantages. To support the theory, an experimental study consisting of three algorithms each applied on a different mining structure layer presented to provide a better understanding of the concept. The next step of this research is planned to integrate the methodology into multidimensional data and cube structures.

Mallamma (2011) under title **Cross Lingual Information Retrieval Using Search Engine and Data Mining** Presents results from the search engine (URLs and ‘snippets’ provided from the web pages) are retrieved, and translated to the target language. Available lexicons and ontologies are also used in the translation. Further, a heuristic result processing mechanism described below is used to identify the relevance of results retrieved with respect to the source language. The objective of the post processing is to aggregate the results for the given query, rank and reorder the results, and present the results in a new sorted order based not only on the output of the search engine but also on the content of the retrieved documents. Thus, these downloaded documents are indexed and comparable document scores for the downloaded documents are calculated.

The metrics utilized are word relevance, word to document relationship and clustering strategies. Finally, all the returned documents are sorted into a single ranked list along with display mechanisms for helping the user view and decide on the results. These measures have different connotations in traditional data mining. a) Candidate word: words that have high information gain in a given document. b) Information gain: the parametric importance of a word in the retrieved document. c) Pair-wise measure: the correlations between the candidate word and the input keywords are found using the pair wise measure. They also give the relationship between words and help disambiguate similar words. d) Cluster relationship: this gives a measure of the degree of clustering of candidate words in a given document.

Velmurugan & Vijayaraj (2012) under title “Efficient Query Optimizing System for Searching Using Data Mining Technique“ aimed to design and develop tools that abstract away the fundamental complexity of XML based Web services specifications and toolkits, and provide an elegant, intuitive, simple, and powerful query based invocation system to end users. Web services based tools and standards have been designed to facilitate seamless integration and development for application developers. As a result, current implementations require the end user to have intimate knowledge of Web services and related toolkits, and users often play an informed role in the overall Web services execution process .We employ a set of algorithms and optimizations to match user queries with corresponding operations in Web services, invoke the operations with the correct set of parameters, and present the results to the end user. Our system uses the Semantic Web and Ontologies in the process of automating Web services invocation and

execution. Every user has a distinct background and a specific goal when searching for information on the Web. The goal of Web search personalization is to tailor search results to a particular user based on that user's interests and preferences.

(2-6): Study Contribution to knowledge

A **search engine** that is dedicated for Arabic documents searching operations; the existing system suffers from various weaknesses such as the lack of good stemming and rooting algorithms for Arabic texts. In addition, it is difficult for foreign companies to work on Arabic documents comparing to us the people of the language.

The Arabic search engine consider as open system which can show new data and also the size of data become more bigger in the data base and the weighting for each document according to changing in documents and also follow a vector model which retrieve relevant document and of query or not retrieve any things these an advantage of vector model because the Boolean model retrieve just the exact mach.

Before we describe search engine techniques, it is useful to understand how a Web search engine is work.

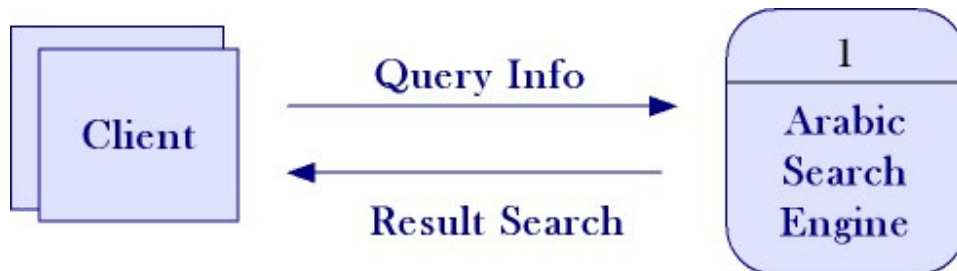


Figure (2.2): Context diagram .

Clarification:

The query module is responsible for receiving search requests or Query from users, and retrieved the ranking result considering their relevancy to the input query there's a big world of operations run behind the scenes.

one of these operations is done prior to building the huge search engine database using crawler to provide the grist for its operation , crawlers are small programs that browse the web on the search engines behalf similarly to how a human user would follow links to reach different pages , the programs are given a starting set of URLs whose pages they retrieve from the web , the Crawler extract URLs appearing in the retrieved pages and give the information to the crawler control module , this modules determines what links to visit next and feeds the links to visit back to the crawler ,crawler continue visiting the web until local resources such as storage are exhausted and indexing . indexing the documents means building structures that allow the search process to

become much faster, plus it makes ranking the results much easier process to, but our concern here is about the way indexers work ; indexers don't leave the documents as they're retrieved , it takes the important words to get them to their original forms, this makes it possible to retrieve documents which contains any from of the words in the search criteria or query.

Related Approaches:

The search engine enable user to search for any information needs in short time and retrieve many view related to the query , so its good for user to decide which result the user need .

This search engine especially for Arabic languages , retrieving Arabic document

In short time, high performance .

System Main Architecture

The plentiful content of the World-Wide Web is useful to millions. Many information seekers use a search engine to begin their Web activity. In this case, users submit a query, typically a list of keywords, and receive a list of Web pages that may be relevant, typically pages that contain the keywords. In this text we discuss the building good search engines, and describe some of the techniques that are useful.

Many of the search engines use well-known information retrieval (IR) algorithms and techniques. However, IR algorithms were developed for relatively small and coherent collections such as news- paper articles or book catalogs in a (physical) library. The

Web, on the other hand, is massive, much less coherent, changes more rapidly, and is spread over geographically distributed computers. This requires new techniques, or extensions to the old ones, to deal with the gathering of the information, to make index structures scalable and update efficiently, and to improve the discriminating ability of search engines. For the last item, discriminating ability, it is possible to exploit the linkage among Web pages to better identify the truly relevant pages.

There is no question that the Web is huge and challenging to deal with. Several studies have estimated the size of the Web, and while they report slightly different numbers, most of them agree that over a billion pages are available. Given that the average size of a Web page is around 5-10K bytes, just the textual data amounts to at least tens of terabytes. The growth rate of the Web is even more dramatic. According to, the size of the Web has doubled in less than two years, and this growth rate is projected to continue for the next two years. Aside from these newly created pages, the existing pages are continuously updated.

For example, in a study of over half a million pages over 4 months, it was found that about 23% of pages changed daily. In the .com domain 40% of the pages changed daily, and the half-life of pages is about 10 days (in 10 days half of the pages are gone, i.e., their URLs are no longer valid). (Junghoo Cho and Hector Garcia-Molina, 2000)

Before we describe search engine techniques, it is useful to understand how a Web search engine is, typically put together.

After reading and through examining previous studies related to the subject of this study, we found that the most important characteristics that distinguish this system from the other previous studies and can be stated as follows:

2.6.1 Functional Requirements:

1. The system shall work under Windows Operating System.
2. The system shall have a flexible and transparent interface so that the entire user at the different levels of skill can access and use the system.
3. This system shall focus on crawling, indexing and searching Arabic documents; these operations are achieved for our project.
4. There are many modules in this system where many of these modules require a number of functional requirements. Modules and requirements of store subsystem
5. The system shall have the ability to update the it's databases regularly.
6. The system shall manage the query over the operation and data flow.
7. In the proposed system, some users such as the end user could not modify anything; this type of users can only view just what they need.

2.6.2 Non Functional Requirements:

1. Performance requirement: The system must have a high performance by deleting duplicate records from the database and removing old records from it.
2. Security requirement: The system must have a high level of security by implementing two or more levels of security.
3. Portable and efficient.
4. Updateable: The system will be easily because it has to be updated frequently.

5. The system is comprised of four subsystems:

- ✓ Crawler subsystem.
- ✓ Stemmer subsystem.
- ✓ Rooting subsystem.
- ✓ Indexer subsystem.
- ✓ Search Engine web site subsystem.

6. The system will be fast enough to load and to execute, but as it a web application, it sometimes relies at the speed of the internet connection, and may rely at the speed of the PC if it was old. In conclusion system shall be loaded depending on the internet speed, and the PC speed.

7. Availability: The system service available 24/7.

8. The system shall contain all necessary information required for information seekers.

9. The system shall utilize the use of the CPU.

10. Throughput: The system retrieves multiple views of the query results.

Design is multi-step process that focuses on data structure software architecture, procedural details, and interface between modules. Design is the place where quality is fostered in software engineering. and is the perfect way to accurately translate a customer's requirement in to a finished software product. The design of an information system produces the details that state how a system will meet the requirements identified during analysis. The emphasis is on translating the performance, requirements into design specifications. With the tremendous growth of information available to end users

through the Web sites, ranking search engines optimization come to play ever a more critical role. Nevertheless, because of their general purpose approach, it is always less

2.7 Project plan :

2.7.1 Feasibility study

In any project, a feasibility study is conducted to check if the project implementation is worthy or not.

In our project we concentrated on getting answers for the following basic questions for any feasibility study:

1- Does the system contribute to the overall objectives of the searching?

As we mentioned before, the system will make an efficient difference in the processes of the searching.

By implementation the proposed system, time, money and effort used to save and will be effectively used in the future. So, the system does contribute to the goals of the overall searching.

2- Can the system be implemented using the current technology?

Considering the current techniques and equipment available , we think we have the required ability , knowledge and resources to implement the Arabic search engine system .

First: Economic feasibility

Project planning:

To determine whether the system is economically feasible we should consider:

- System analyst's time
- Cost of system research
- Cost of employees time for the study
- Estimated cost for hardware
- Estimated cost of packaged hardware and software development.

Determining our project requirement was an easy task and it didn't require extensive research, study and analysis, it is a system that could be understood easily.

In terms of the cost of the hardware and software our project is cost effective because it doesn't need costly hardware and software.

Benefit			
Tangible	<ul style="list-style-type: none">Increased flexibilityCost reductionTime saving		
	Intangible	<ul style="list-style-type: none">Getting one step closer to the e_ future	
	Costs		
Tangible	One – time costs	<u>Hardware:</u> Server Pc	750JD
		<u>Software:</u> C#	5500

		Windows XP User training Site preparation	300 1500
	Recurring costs	<ul style="list-style-type: none"> • Application software • maintenance • Incremental data storage expense 	
intangible	Total costs The need to deal with novice users		14,207

Table 2.1: Project Plane.

Second: technical feasibility

Risk analysis:

1. If additional requirement is demanded, there will be a probability of time bottle neck problem.
2. There will be a probability that is lack of human staff to implement the system.
3. To much high number of records which lead to less performance and space and over load to system.

We summarize the risk in the following table:

<i>Risk factor</i>	<i>Risk type</i>	<i>Possible Risk</i>
Project size	project	Number programmer of the project not enough
Client record	product	Too much high number of records which lead to less performance and space and over load to system
Requirement	Project & product	Change in requirements with time

Table 2.2 Risks Analysis

Third: Operation Feasibility

The system will be used by prospected users, our system screens are clear enough that the user can use the provided services of the system without difficulties, considering the current techniques and equipment available, we think we have the required ability, knowledge and resources to implement our part of the e-department system.

Gantt Chart

A Gantt chart is a horizontal bar chart developed as a production control tool in 1917 by Henry L. Gantt, an American engineer and social scientist. Frequently used in project management, a Gantt chart provides a graphical illustration of a schedule that helps to plane, coordinate, and track specific tasks in a project.

A Gantt chart is constructed with a horizontal axis representing the total time span of the project, broken down into increments (for example, days, weeks or months) and vertical axis representing the tasks that make up the project.

	Task Name	Duration
1	Planing phase	10 days?
2	proplem definition	2 days?
3	project goals	1 day?
4	set: project schedule	1 day
5	project feasibility	4 days?
6	staff the project	2 days?
7	lunch the project	1 day?
8	plaining report	2 days?
9	Analysis phase	13 days?
10	gather all information	5 days?
11	define the scope	1 day?
12	system requirement	5 days?
13	functional requirement	5 days?
14	non functional requirement	2 days?
15	review requirement	3 days?
16	write the documentation	3 days?
17	analysis report	2 days?
18	Design phase	17 days?
19	Design User interfase	5 days?
20	Design DataBase and Model	5 days?
21	Build A prototype	6 days?
22	Review Design	2 days?
23	Design Report	1 day?
24	implementation	32 days?
25	Implement Matching Algorithm	13 days?
26	Writhing the project IMain classes	14 days?
27	Integration The implemented	5 days?
28	testing the project	5 days?
29		

Figure 2.5 Gantt chart

A **PERT** chart is a project management tool used to schedule, organize, and coordinate tasks within a project.

PERT stands for "Program Evaluation Review Technique", a methodology developed by the U.S Navy in the 1950s to manage the Polaris submarine missile program.

A similar methodology, the Critical Path Method (CPM), which was developed for project management in the private sector at about the same time, has become synonymous with PERT, so that the technique is known by any variation on the names: PERT, CPM, or PERT/CPM.

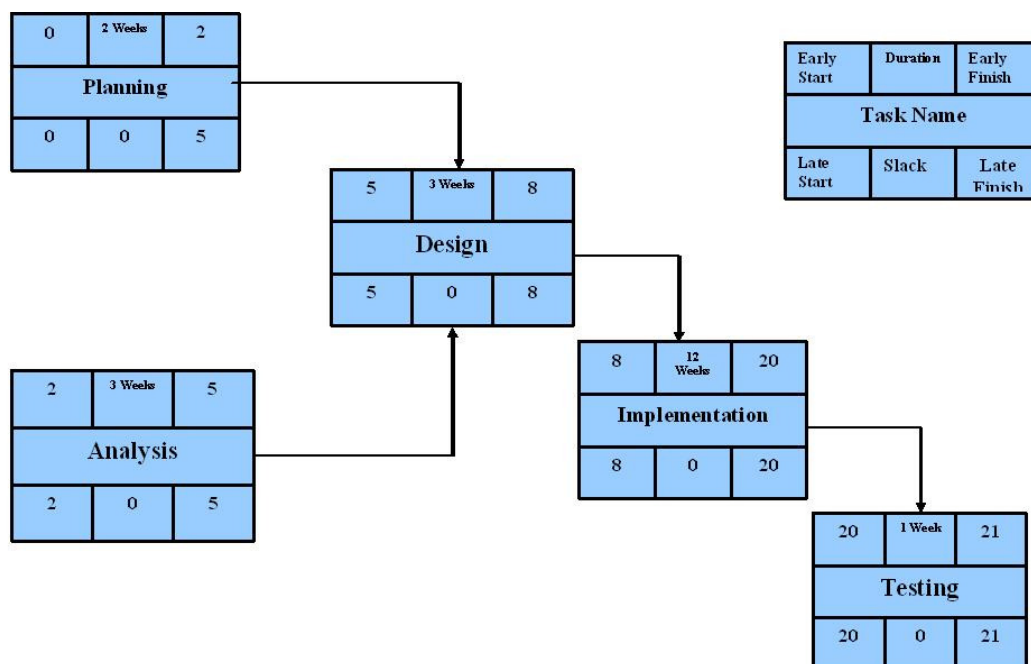


Figure 2.4 Project Plane

CHAPTER THREE

Method and Procedures

(3-1): Introduction

(3-2): Study Methodology

(3-3): Study Design

(3-4): Implementation

(3-5): Study Population and Sample

(3-6): Study Tools and Data Collection

(3-7): Reliability and Validity

(3-1): Introduction

In this chapter we will describe in detail the methodology used in this study , Next, we will design the study model and implemented by programming and explain the study tools and the way of data collections.

After that, we will discuss each subsystem and implement it to make the system optimized , the data set is split into training data and validation data. For each split, the training is done on training data and tested across the validation data. The results are then averaged over the splits. The main disadvantage of this method is some observations may not be selected or some observations may be selected more than once. In other words, validation subsamples may be overlapped.

(3-2): Study Methodologies

Conceptual model or study is a model that exists only in the mind and its used to help us know and understand the subject matter they represent , its referred to model which are formed after a conceptualization process in the mind

Its represent are the necessary means human employ to think and solve problem , some of formalization is usually required for locating the intended semantic to avoid misunderstanding and confusion .

(3-3): Study Design

Entity Relationship Diagram (ERD):

The system has tow databases the first one especially for crawler and the second for indexer we will explain them:

3.3.1 Crawler ERD:

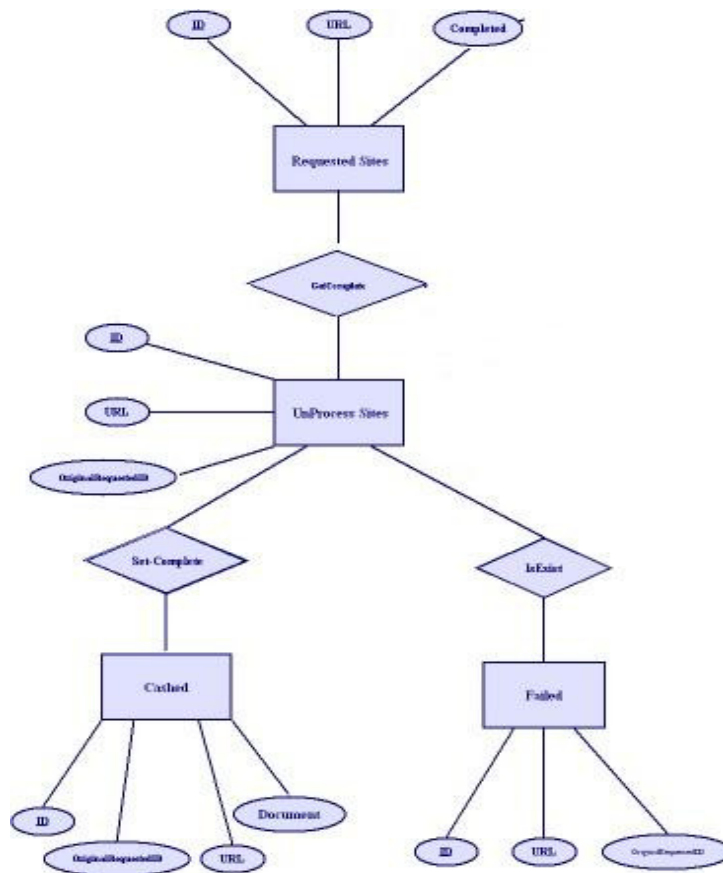


Figure (3.1): Crawler ERD

This process is called Web crawling or spidering. Many sites, in particular search engines, use spidering as a means of providing up-to-date data. Web crawlers are mainly used to create a copy of all the visited pages for later processing by a search engine that will index the downloaded pages to provide fast searches. Crawlers can also be used for automating maintenance tasks on a Web site, such as checking links or validating HTML code.

A Web crawler is one type of bot, or software agent. In general, it starts with a list of URLs to visit, called the seeds. As the crawler visits these URLs, it identifies all the

hyperlinks in the page and adds them to the list of URLs to visit, called the crawl frontier. URLs from the frontier are recursively visited according to a set of policies.

The crawler has four tables

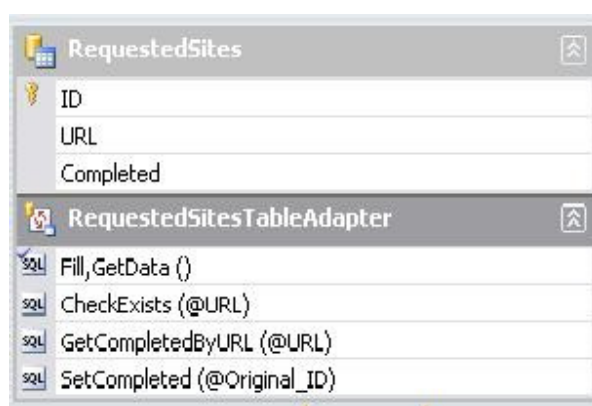
1- Requested table :

This table receives the start site from the user and parsing the link inside it,

Requested Table have three basic queries:

- √ Check Exists
- √ Get completed "Select".
- √ Set completed "update".

First query check exists to check if the website in requested table or not, second get completed to take value from form, third set completed to casting the links cashed or failed if it completed it will give it ID



2- Unprocessed table:

Save the extracted link which comes from the requested table and distribute this link into two tables the cached and the process follows as

While (UNPROCCESSED>0)

- Take and remove from UNPROCCESSED.
- Get webpage of URL and save it in cached table.
- If link failed save it failed pages table.
- If succeeded parse page (extract link).

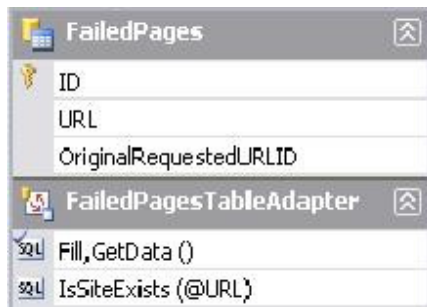


3- Failed Pages:

The pages can not appear it's content when the crawler visit it, it has three attributes ID to give the links sequence number "it's primary key" and type "int" .

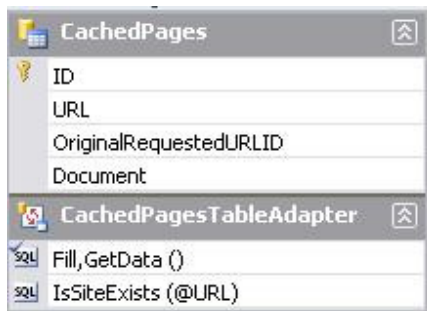
URL to save the links for access the query and type "nvarchar (1000)"

OriginalRequestedURLID type "int".



4- Cashed Pages:

It have filter based on URL used in IsSiteExists query to check if the URL exists or not



3.3.2 Indexer ERD:

the other database for indexer also connected to the crawler database which have many tables on it :

- 1- Ranking table
- 2- Stop Word table : this table have many words remove from the documentation like :

هذا , انا , ان , ماذا , كيف , كان , لم , لعل

- 3- Term_df_idf table : this table calculate the IDF for each term according to this equation ($\log_{10}N/N_i$)

Where N is the number of the documents

And N_i the number of document have the certain term .

4- Tem_tf table: this table calculate the frequency for each distinct term .

- For each page in cached
- Removing a stop word list
- Fined root's for each words from Arabic root
- Create documents terms table

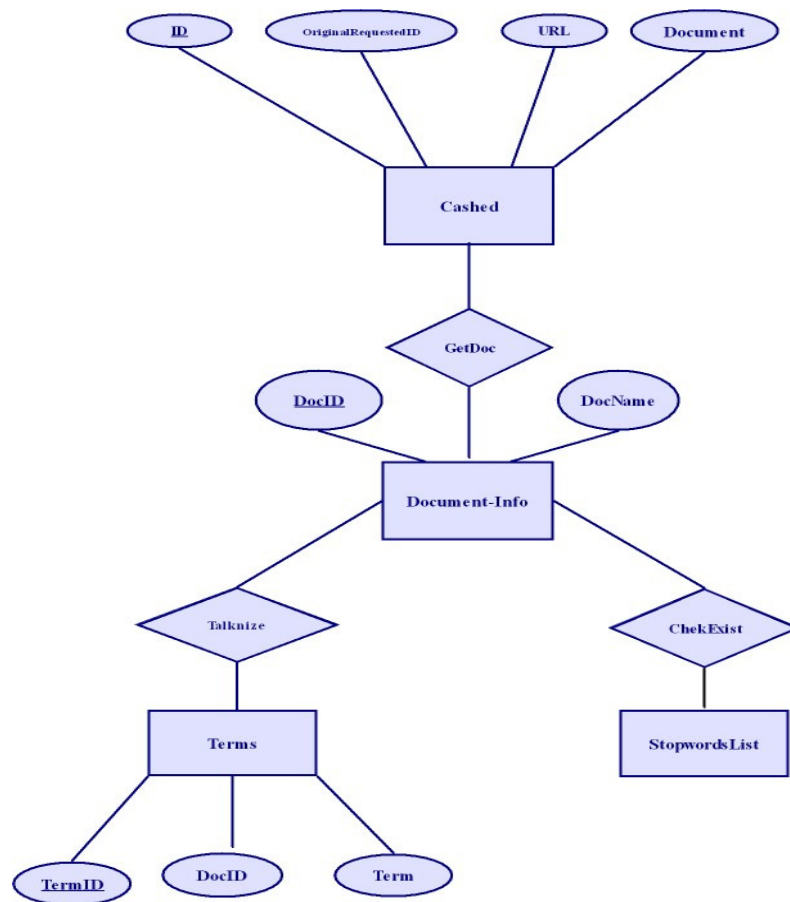
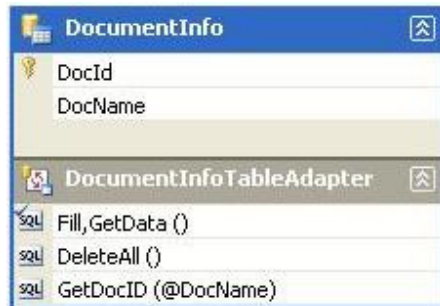


Figure (3.2): Indexer ERD

1-The Document Info table



This table contains two fields **DocId** which have the document Id **DocName** which have the document name or URL and also this table has many queries

- DeleteAll()

This query deletes all information from this table when the message box appears to the user and asks the user to continue the previous indexing process if the user selects no.

- GetDocID(@DocName)

Get documents id from the cached pages

2 Terms table



this table contains all the terms and the term may be iterated more than one time in one document

This table consists three field **TermId** give any term id automatically and **Term** which contains the stem word **DocId** which contains the id for document which contain this term .

Query:

Delete All: delete information from the table if user select no from the message box ask user To (Continue previous indexing process)

stop word table

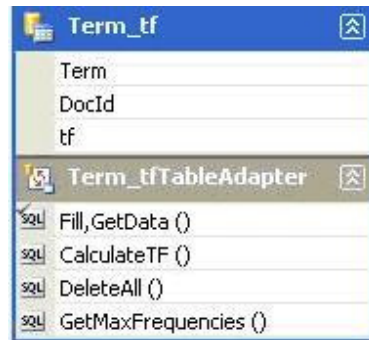


This table the stop word list which contains 1216 words these is useful when the user enter any stop word list or on the query the engine doesn't wasting the time by searching of it , and this filtering the document from the low scientific words so the system exploit tae database efficiency by storing a useful words .

The stop word list contains a collection of categories :

- Question :ماذا , كيف , اين , هل
- Conjunction words:.....و , أو
- And many of low scientific words

a. **Term_tf**



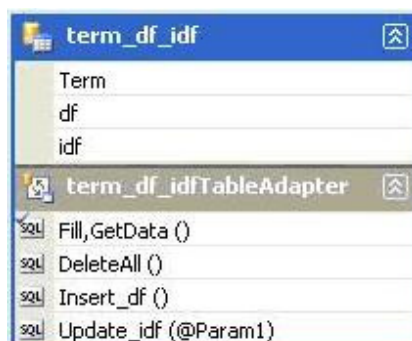
Term_tf table: this table calculate the frequency for each distinct term , the table contains the distinct term and the frequency of it in each document .

Query:

Calculate TF():

This query calculate a term frequency

b. **Term df_idf**



1- This table calculate the IDF for each term according to this equation ($\log_{10}N/N_i$)

Where N is the number of the documents

And N_i the number of document has the certain term.

The processes in indexer are follows as:

- For each page in cashed.
- Removing a stop word list.
- Fined root's for each words from Arabic root.
- Create documents terms table.

The structure of the rooting algorithm as shown bellow

Algorithm : Root extraction for third-based verbs

The proposed algorithm works by executing the following steps:

1. Accept the words from the file.
2. If the word consists of less than three words, then remove the word from the document; it is considered as a stop word.
3. Test the word's letters and insert any letter that is matched with any letter of the word "سألتهمونيها" to a temporary array named as Temp_Array. Other words are stored in an array called Root_Array.
4. if the letter is ع and is followed ل then insert the letter in the Root_Array as ي letter.
5. **Do While (the number of elements in the array is less than 3)**
 - a. If the first letter is م and the last letter is ن then store the letter ن in the Root_Array.

- b.** If the number of elements of the Temp_Array is more than two AND the last letter in the word is ن, and the second last element is one of the following characters (أ, و, ي); store the third last word of the Temp_array in the Root_Array.
- c.** If the number of elements of the Temp_Array is more than two elements, and the last element of that array is ا and the second last element of the array is و AND the last letter of the original word is ا then store the third last word of the Temp_array in the Root_Array.
- d.** If the number of element of the Temp_Array is more than three AND the last letter of that array is Temp_Array is ت or ة AND the second last letter of that array is ل AND the third last letter is ي then store the fourth last letter of the Temp_Array in the Root_Array.
- e.** If number of the word's letters in the Temp_Array is more than two letters and the last letter was ة or و and the second last letter is ي and store the third last element of the Temp_Array in the Root_Array.
- f.** If the number of elements of the Temp_Array is more than one element and that element is one of the following characters (ة, ه, ت, ا, و, ي) and the letter is the last character in the original word then remove that letter from the Temp_Array.
- g.** If the number of elements of the Temp_Array is more than one element and that element is one of the following characters (ا, و, ي, ي) and the letter is the last character in the original word then store the second last letter existed in the Temp_Array in the Root_Array.

- h. If the number of elements of the Temp_Array is more than two characters AND the last letter of that array is ت AND the second last letter is ل and the letter ت is the letter of the original word. Store the third last letter in the Root_Array.
- i. Else store the last letter of the Temp_Array in the Root_Array.
- j. Arrange the letters of the Root_Array as appeared in the original word.

Flow Chart Diagram (FD)

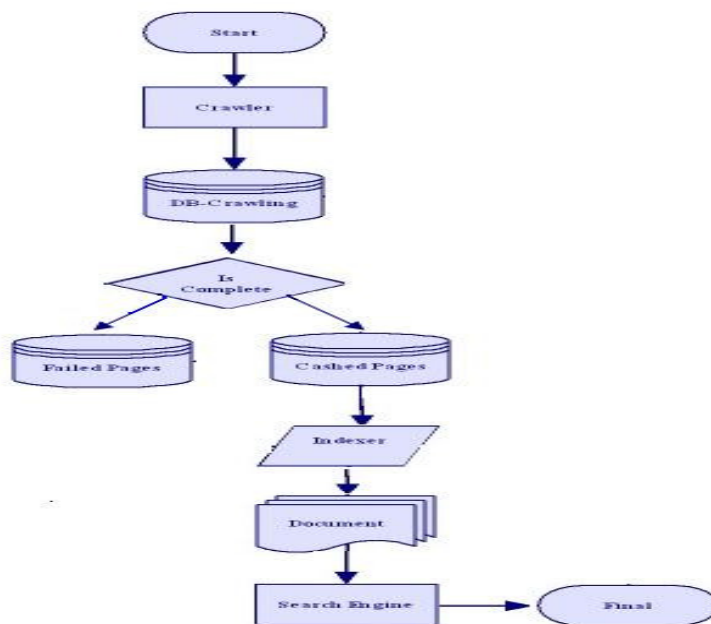
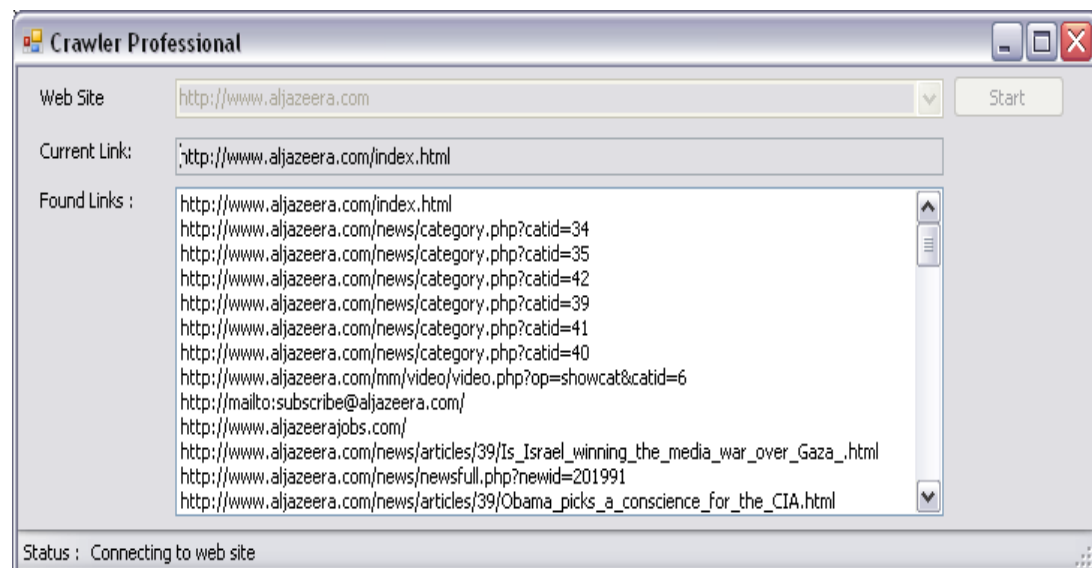
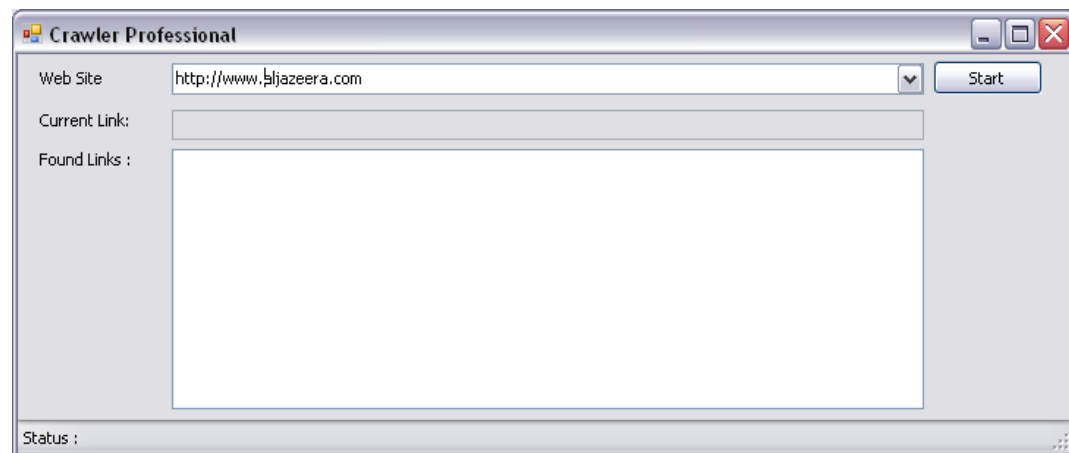


Figure (3.3): Flow Chart Diagram

3.4 Implementation

This phase describe the code of Arabic search engine with the screenshots.

Crawler:



Crawler Code

```
using System;

using System.Collections.Generic;

using System.ComponentModel ;

using System.Data;

using System.Drawing;

using System.Text;

using System.Text.RegularExpressions;

using System.Windows.Forms;

using System.Net;

using CrawlerSystem;

namespace Simple_Crawler
{
    public partial class CrawlerForm : Form
    {
        int CurrentSiteID;

        Crawler crawler = new Crawler();

        public CrawlerForm()
        {
            InitializeComponent();
        }
    }
}
```

```

        crawler.NewURI += new Crawler.NewURIHandler(c_NewURI);

        crawler.IsNewLink += new Crawler.NewLinkHandler(c_IsNewLink);
    }

    void c_NewURI(string URL)
    {
        listBox.Items.Add(URL);

        Application.DoEvents();

        WebSitesDataSet.UnProcessedSitesRow row =
webSitesDataSet.UnProcessedSites.NewUnProcessedSitesRow();

        row.URL = URL;

        row.OriginalRequestedURLID = CurrentSiteID;

        webSitesDataSet.UnProcessedSites.Rows.Add(row);
    }

    bool c_IsNewLink(string URL)
    {
        if (cachedPagesTableAdapter.IsSiteExists(URL, CurrentSiteID).HasValue)
            return false;

        if (failedPagesTableAdapter.IsSiteExists(URL, CurrentSiteID).HasValue)
            return false;

        if (unProcessedSitesTableAdapter.FindByUrl(URL, CurrentSiteID).HasValue)
            return false;

        return true;
    }

```

```
}
```

```
private void Crawler_Load(object sender, EventArgs e)
```

```
{
```

```
    this.requestedSitesTableAdapter.Fill(this.webSitesDataSet.RequestedSites);
```

```
}
```

```
private void btnStart_Click(object sender, EventArgs e)
```

```
{
```

```
    if (requestedSitesTableAdapter.CheckExists(cmbSite.Text).HasValue == false)
```

```
    {
```

```
        requestedSitesTableAdapter.InsertNewSite(cmbSite.Text);
```

```
        CurrentSiteID = requestedSitesTableAdapter.GetIDByUrl(cmbSite.Text).Value;
```

```
        unProcessedSitesTableAdapter.Insert(cmbSite.Text, CurrentSiteID);
```

```
        requestedSitesTableAdapter.Fill(this.webSitesDataSet.RequestedSites);
```

```
    }
```

```
    else
```

```
    {
```

```
        bool? completed = requestedSitesTableAdapter.GetCompletedByUrl(cmbSite.Text);
```

```
        if (completed.Value == true)
```

```
        {
```

```
            MessageBox.Show("Job completed.", "Done Job");
```

```
            return;
```

```
        }
```

```
        CurrentSiteID = requestedSitesTableAdapter.GetIDByUrl(cmbSite.Text).Value;
```

```

    }

    unProcessedSitesTableAdapter.FillByOriginalID(webSitesDataSet.UnProcessedSites,
CurrentSiteID);

    cmbSite.Enabled = btnStart.Enabled = false;

    StartOperation();
}

private void StartOperation()
{
    if (webSitesDataSet.UnProcessedSites.Rows.Count > 0)
    {
        WebSitesDataSet.UnProcessedSitesRow row =
(WebSitesDataSet.UnProcessedSitesRow)webSitesDataSet.UnProcessedSites.Rows[0];

        crawler.CurrentURL = textBox1.Text = row.URL;

        WebPageDownloader Downloader = new WebPageDownloader(row.URL);

        Downloader.NewEvent += new WebPageDownloader.NewEventHandler(downloader_NewEvent);

        Downloader.PageRecieved += new WebPageDownloader.NewEventHandler(downloader_PageRecieved);

        Downloader.Failed += new MethodInvoker(downloader_Failed);

        Downloader.GetWebPage();
    }
    else
    {

```

```

        requestedSitesTableAdapter.SetCompleted(CurrentSiteID);

        lblStatus.Text = "Operation completed successfully";

        cmbSite.Enabled = btnStart.Enabled = true;
    }
}

void downloader_Failed()
{
    failedPagesTableAdapter.InsertNewSite(crawler.CurrentURL, CurrentSiteID);
    webSitesDataSet.UnProccessedSites.Rows[0].Delete();
    unProccessedSitesTableAdapter.Update(webSitesDataSet.UnProccessedSites);
    Invoke(new WebPageDownloader.NewEventHandler(NewStatus), "Download Failed.");
    Invoke(new MethodInvoker(StartOperation));
}

void downloader_PageRecieved(string Message)
{
    Invoke(new WebPageDownloader.NewEventHandler(PageRecieved), Message);
}

void downloader_NewEvent(string Message)
{
    Invoke(new WebPageDownloader.NewEventHandler(NewStatus), Message);
}

```

```

void NewStatus(string Message)
{
    lblStatus.Text = Message;
}

void PageRecieved(string Page)
{
    lblStatus.Text = "Page Downloaded Successfully.";
    cachedPagesTableAdapter.Insert(crawler.CurrentURL,
Encoding.Unicode.GetBytes(Page), CurrentSiteID);
    listBox.Items.Clear();
    Application.DoEvents();
    crawler.ParseUri(Page);
    unProccessedSitesTableAdapter.Update(webSitesDataSet.UnProccessedSites);
    int id =
((WebSitesDataSet.UnProccessedSitesRow)webSitesDataSet.UnProccessedSites.Rows[0]).ID;
    webSitesDataSet.UnProccessedSites.Rows.RemoveAt(0);
    webSitesDataSet.UnProccessedSites.AcceptChanges();
    unProccessedSitesTableAdapter.DeleteByID(id);
    StartOperation();
}

private void cmbSite_TextChanged(object sender, EventArgs e)
{
    btnStart.Enabled = cmbSite.Text.Trim() != "";
}

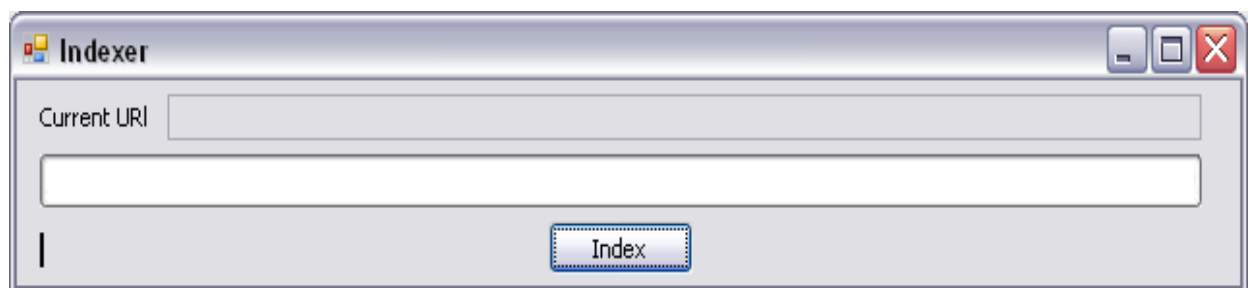
```

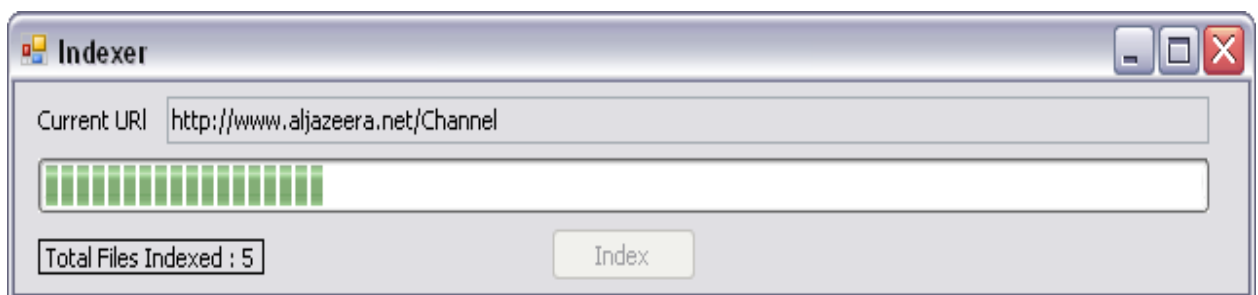
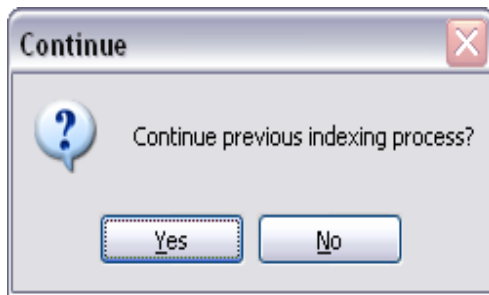
```
}

private void listBox_SelectedIndexChanged(object sender, EventArgs e)
{
    if (listBox.SelectedItem != null)
        txtSelected.Text = listBox.SelectedItem.ToString();
}

private void Crawler_FormClosed(object sender, FormClosedEventArgs e)
{
    Environment.Exit(0);
}
}
```

Indexer:





Indexer Code...

```
using System.Collections;  
using System.ComponentModel;  
using System.Data;  
using System.Drawing;  
using System.Text.RegularExpressions;  
using System.Windows.Forms;  
using System.IO;
```

```

using System.Data.SqlClient;

using System.Text;

using ArabicRouter;

namespace Indexer
{
    public partial class IndexerForm : Form
    {
        public IndexerForm()
        {
            InitializeComponent();

            bool StopWord(string word)
            {
                return stopWordTableAdapter.CheckExists(word) != null;
            }

            void Index()
            {
                progressBar1.Value = 0;

                SqlConnection con = new
SqlConnection(Properties.Settings.Default.sqlDBConnectionString);

                con.Open();

```

```

SqlCommand cmd = new SqlCommand("drop table weights", con);

try
{
    cmd.ExecuteNonQuery();
}

catch
{
    if (MessageBox.Show("Continue previous indexing process?", "Continue",
        MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.No)
        .DeleteA
        {
            documentInfoTableAdapter.DeleteAll();
            term_df_idfTableAdapter.DeleteAll();
            term_tfTableAdapter.DeleteAll();
            termsTableAdapter ll();
        }

        Database1DataSet.CachedPagesDataTable table =
cachedPagesTableAdapter.GetNotIndexed();

        int count = 0;

        progressBar1.Maximum = table.Rows.Count;

        foreach (Database1DataSet.CachedPagesRow CachedRow in table)
        {
            lblTotal.Text = "Total Files Indexed : " + count++;
            txtCurrentURL.Text = CachedRow.URL;

```

```

documentInfoTableAdapter.Insert(CachedRow.URL);

int DocId = (int)documentInfoTableAdapter.GetDocID(CachedRow.URL);

string data =
(string)cachedPagesTableAdapter.GetDocument(CachedRow.ID);

Regex regex = new Regex("[أ-ي]+");

foreach (Match match in regex.Matches(data))
{
    try
    {
        if (StopWord(match.Value) || match.Value.Length < 3)
            continue;

        string stemmed = ArabicWordStem.GetRoot(match.Value);

        termsTableAdapter.Insert(stemmed, DocId);

        Application.DoEvents();
    }
    catch { }
}

cachedPagesTableAdapter.SetDone(CachedRow.ID);

progressBar1.Increment(1);
}

term_tfTableAdapter.CalculateTF();

term_df_idfTableAdapter.Insert_df();

```

```

term_df_idfTableAdapter.Update_idf(cachedPagesTableAdapter.GetCount().Value);

cmd.CommandText = "create table weights (term nvarchar(50)";

sqlDBDataSet.DocumentInfoDataTable docInfo =
documentInfoTableAdapter.GetData();

foreach (sqlDBDataSet.DocumentInfoRow row in docInfo)

    cmd.CommandText += ",D" + row.DocId + " decimal(18,5) ";

cmd.CommandText += ",q decimal(18,5))";

cmd.ExecuteNonQuery();

weightsTableAdapter.InsertTerms();

foreach (sqlDBDataSet.DocumentInfoRow row in docInfo)

{

    foreach (sqlDBDataSet.Term_tf_idfRow row2 in
term_tf_idfTableAdapter.GetTermsWeights(row.DocId))

    {

        cmd.CommandText = "UPDATE weights SET D" + row.DocId + " = " +
row2.Weight + " WHERE (term = '" + row2.Term + "')";

        cmd.ExecuteNonQuery();

    }

}

con.Close();

btnIndex.Enabled = true;

}

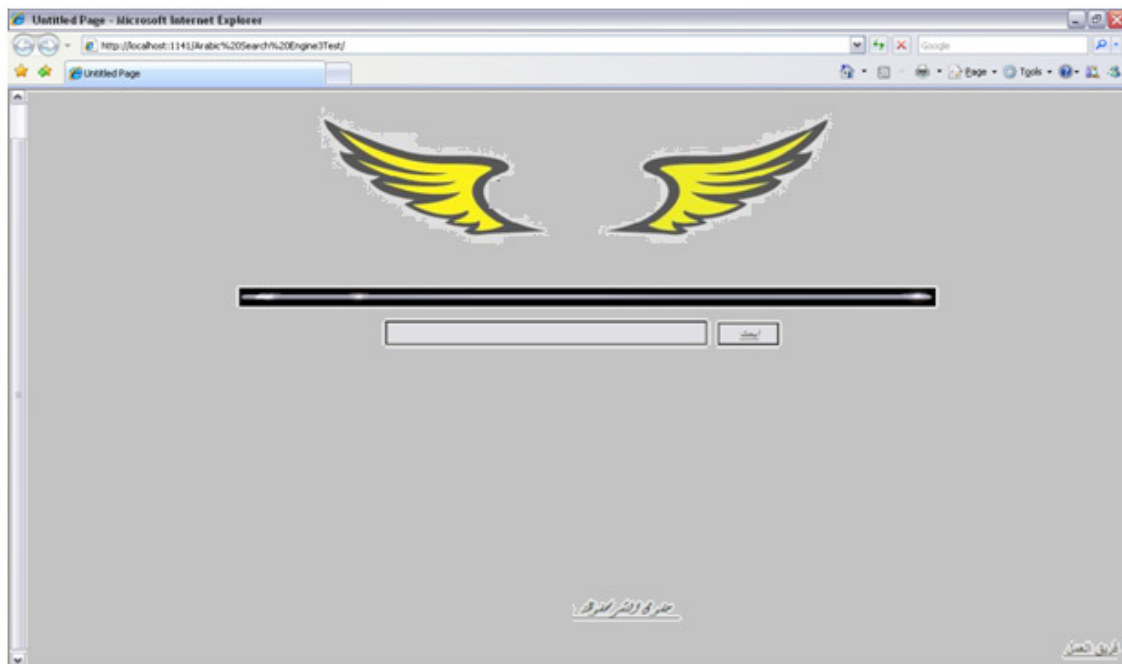
```

```
private void btnIndex_Click(object sender, EventArgs e)
{
    btnIndex.Enabled = false;

    Index();
}

private void VectorModel_FormClosed(object sender, FormClosedEventArgs e)
{
    Environment.Exit(0);
}
}
```

search engine:



Search Engine Code...

```
using System;

using System.Data;

using System.Data.SqlClient;

using System.Configuration;

using System.Collections.Generic;

using System.Web;

using System.Web.Security;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Web.UI.WebControls.WebParts;
```

```
using System.Web.UI.HtmlControls;

using System.Text.RegularExpressions;


public partial class Docs_Default : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
    }

    protected void Button2_Click(object sender, EventArgs e)
    {
        TextBox3.Text = TextBox3.Text.Trim();

        if (TextBox3.Text == "") return;

        string[] words = TextBox3.Text.Split(' ');

        Dictionary<string, int> qweights = new Dictionary<string, int>();

        foreach (string word in words)
        {
            if (StopWord(word)) continue;

            string root;

            try
            {
                {
```



```

        root = ArabicRooter.ArabicWordStem.GetRoot(word);
    }
    catch { continue; }
    if (qweights.ContainsKey(root))
        qweights[root]++;
    else
        qweights[root] = 1;
}

string qterms="";
foreach(string qterm in qweights.Keys)
    qterms+=" "+qterm+" ";
qterms=qterms.Substring(0,qterms.Length-1);

string connectionString = @"Data Source=.;Initial
Catalog=SQLDB.MDF;Integrated Security=True";

SqlDataAdapter da = new SqlDataAdapter();
da.SelectCommand = new SqlCommand("select * from weights where term
in("+qterms+")", new SqlConnection(connectionString));
DataSet ds = new DataSet();
da.Fill(ds, "weights");

DataTable weights = ds.Tables["weights"];

```

```

Dictionary<int, decimal> docSimilarities = new Dictionary<int, decimal>();

foreach (DataRow weightsrow in weights.Rows)
{
    for (int i = 1; i < weights.Columns.Count - 1; i++)
    {
        string docname = weights.Columns[i].ColumnName.Trim('D');
        int docid = int.Parse(docname);
        decimal weight = 0;
        if (weightsrow[i] is DBNull == false)
            weight = (decimal)weightsrow[i] * qweights[(string)weightsrow[0]];
        if (docSimilarities.ContainsKey(docid))
            docSimilarities[docid] += weight;
        else
            docSimilarities[docid] = weight;
    }
}

SortedDictionary<decimal, List<int>> sortedDocSimilarities = new
SortedDictionary<decimal, List<int>>();

foreach (int docid in docSimilarities.Keys)
{
    decimal weight = docSimilarities[docid];
    List<int> list = new List<int>();

```

```

    if (sortedDocSimilarities.ContainsKey(weight))
    {
        list = sortedDocSimilarities[weight];
    }
    else
    {
        sortedDocSimilarities.Add(weight, list);
    }
    list.Add(docid);
}

```

```

List<decimal> Reversed = new List<decimal>(sortedDocSimilarities.Keys);
Reversed.Reverse();
decimal threshold = .4M;

```

```

int t = 0;
while (t < Reversed.Count)
{
    if (Reversed[t] <= threshold)
    {
        Reversed.RemoveAt(t);
    }
    else
    {
        t++;
    }
}

```

```

sqlldbTableAdapters.DocumentInfoTableAdapter docinfo = new
sqlldbTableAdapters.DocumentInfoTableAdapter();

Literal lit = new Literal();

```

```

lit.Text = "<div style=\"text-align: left\">";

Panel1.Controls.Add(lit);

foreach(decimal weight in Reversed)
{
    foreach (int docid in sortedDocSimilarities[weight])
    {
        HyperLink link = new HyperLink();

link.NavigateUrl = link.Text =docinfo.GetURL(docid).ToString()+TextBox3.Text;

        if                (link.Text.Contains("%"))                link.Text                =
link.Text.Substring(0,link.Text.IndexOf("%")+TextBox3.Text;

                // string data;

                // Regex regex = new Regex(@"[!-؟]+");

                // foreach (Match match in regex.Matches(data))

                Panel1.Controls.Add(link) ;

                lit = new Literal();

                lit.Text = "<br /><br />";

                Panel1.Controls.Add(lit);

            }

        }

        lit = new Literal();

        lit.Text = "</div>";

        Panel1.Controls.Add(lit);

    }

```

```

bool StopWord(string word)
{
    sqldbTableAdapters.StopWordTableAdapter sw=new
sqldbTableAdapters.StopWordTableAdapter();
    return sw.CheckExists(word)
}

```

(3-5): Study Population and Sample

To increase credibility, it is important to choose the sample that will represent the population under investigation. The populations of the study are the Arabic search , as the result is a set of web pages ranked according to similarity to the query .

- A similarity measure is a function that computes the degree of similarity between two vectors(Document and Query)
- Using a similarity measure between the query and each document:
 - It is possible to rank the retrieved documents in the order of relevance.
 - It is possible to enforce a certain threshold so that the size of the retrieved set can be controlled.

(3-5): Study Tools and Data Collection

This search engine especially for Arabic languages , retrieving Arabic document

In short time, high performance .

Tools an Techniques:

Tools :

Software support that helps creates models or other requires project components.

- √ Microsoft SQL server 2005.
- √ ASP.net
- √ Microsoft visual studio 2005.
- √ Visio 2007.
- √ Microsoft project.
- √ Microsoft word 2007 to write documentation.
- √ Adobe Photoshop.
- √ Illustrator.

Microsoft SQL server 2005 :

Microsoft SQL Server 2000 is a full-featured relational database management system (RDBMS) that offers a variety of administrative tools to ease the burdens of database

development, maintenance and administration. In this article, we'll cover six of the more frequently used tools: Enterprise Manager, Query Analyzer, SQL Profiler, Service Manager, Data Transformation Services and Books Online. Let's take a brief look at each:

Enterprise Manager:

Is the main administrative console for SQL Server installations?

It provides you with a graphical "birds-eye" view of all of the SQL Server installations on your network. You can perform high-level administrative functions that affect one or more servers, schedule common maintenance tasks or create and modify the structure of individual databases.

Query Analyzer:

Offers a quick and dirty method for performing queries against any of your SQL Server databases. It's a great way to quickly pull information out of a database in response to a user request, test queries before implementing them in other applications, create/modify stored procedures and execute administrative tasks.

SQL Profiler:

Provides a window into the inner workings of your database, you can monitor many different event types and observe database performance in real time. SQL Profiler allows you to capture and replay system "traces" that log various activities. It's a great tool for optimizing databases with performance issues or troubleshooting particular problems.

Service Manager:

Is used to control the MS SQL Server (the main SQL Server process), MSDTC (Microsoft Distributed Transaction Coordinator) and SQL?

Server Agent processes, an icon for this service normally resides in the system tray of machines running SQL Server. You can use Service Manager to start, stop or pause any one of these services.

Data Transformation Services (DTS):

Provide an extremely flexible method for importing and exporting data between a Microsoft SQL Server installation and a large variety of other formats. The most commonly used DTS application is the "Import and Export Data" wizard found in the SQL Server program group.

Hopefully, this article has provided you with a brief introduction to the various tools available to Microsoft SQL Server users. Now get out there and give them a whirl.

Visio 2007

Any software development method is best supported by a tool. The Visio product family is designed to provide the software developer with a complete set of visual modeling tools for development of robust, efficient solutions to real business needs in the client/server, distributed enterprise, and real-time system environment.

Visio products share a common universal standard, making modeling accessible to nonprogrammers wanting to model business processes as well to programmers modeling applications logic. An evaluation version of the Visio tools may obtain at the rational software corporation website as reference.

Microsoft project

Microsoft Project (or MSP) is a project management software program developed and sold by Microsoft which is designed to assist project managers in developing plans, assigning resources to tasks, tracking progress, managing budgets and analyzing workloads.

The first version, Microsoft Project for Windows v1.0 was started in 1987 on contract to a small external company. In 1988 the company was acquired by Microsoft, bringing the development project in-house where it was finished and released in 1990 as part of the company's applications offerings for Microsoft Windows 3.0. Microsoft Project was the company's third Windows-based application, and within a couple of years of its introduction WinProj was the dominant PC-based project management software.

The application creates critical path schedules, although critical chain and event chain methodology third-party add-ons are available. Schedules can be resource leveled, and chains are visualized in a Gantt chart.

Additionally, Project can recognize different classes of users. These different classes of users can have differing access levels to projects, views, and other data. Custom objects

such as calendars, views, tables, filters and fields are stored in an enterprise global which is shared by all users.

Adobe Photoshop and Illustrator

Adobe Photoshop, or simply Photoshop, is a graphics editing program developed and published by Adobe Systems. It is the current and primary market leader for commercial bitmap and image manipulation, and is the flagship product of Adobe Systems.

Adobe Illustrator is a vector-based drawing program developed and marketed by Adobe Systems.

Techniques:

- √ User interviewing technique.
- √ Structured analysis technique.
- √ Structure design technique.

(3-6): Reliability and Validity

Study Tool Reliability

Reliability is fundamentally concerned with issues of consistency of measures. The Internal reliability is concerned with whether or not the indicators that make up the scale of index are consistent, whether or not respondents' scores on any one indicator tend to be related to their scores on the other indicators. The meaning of internal reliability applies to multiple-indicator measures. When you have a multiple-item measure in which each respondent's answers to each question are aggregated form an overall score, the possibility is raised that the indicators do not relate to the same thing. We need to be sure that all indicators are related to each other. If they are not, some of the items may actually be unrelated to design and therefore indicative of something else. To test the internal reliability, we run our system which crawling the Arabic documents and indexing it into database by using a good rooting algorithm and test the search queries by calculates the similarity among web sites.

- 1) Associate a weight to Each Term in document and query.
- 2) Documents and Queries are mapped into term vector space.
- 3) Similarity method is used to compute the degree of similarity between each document stored in the system and user query.
- 4) Documents are ranked by closeness to the query.

Parameters in calculating a weight for a document term or query term:

5) Term Frequency (tf): Term Frequency is the number of times a term i appears in document j (tf_{ij})

6) Document Frequency (df): Number of documents a term i appears in, (df_i).

7) Inverse Document Frequency (idf): A discriminating measure for a term i in collection, i.e., how discriminating term i is. (the term which appear in many documents are not very useful for distinguishing a relevant document from irrelevant one)

$$(idf_i) = \log_{10} (N/ n_i), \text{ where}$$

N = number of documents in the collection

n_i = number of document that contain the term i

Remark: consider n_i/N as a probability of selecting document that contain a query term in it.

we compute the values of the weights w_{ij} ?

- One of the most popular methods is based on combining two factors:
 - The importance of each index term in the document (tf)
 - The importance of the index term in the collection of documents (idf)

Combining these two factors we can obtain the weight of an index term i as :

$$w_{ij} = tf_{ij} * idf_i = tf_{ij} * \log_{10} (N/ n_i)$$

Where:

N = number of documents in the collection

n_i = number of document that contain the term i

- Also called the **tf-idf** weighting scheme

A similarity measure is a function that computes the degree of similarity between two vectors(Document and Query), Using a similarity measure between the query and each document, It is possible to rank the retrieved documents in the order of relevance, also possible to enforce a certain threshold so that the size of the retrieved set can be controlled.

Chapter Four

Analysis of Results & Tests experiment

(4-1): Introduction

(4-2): Descriptive analysis of design model

(4-3): Study Tests

(4-1): Introduction

According to the purpose of the research and the research model presented in the previous chapter, this chapter will describes the results of the data collected according to the research questions and approach to personalized search that involves building models of user context as ontological profiles by assigning implicitly derived interest scores to existing concepts in domain ontology.

(4-2): Descriptive analysis of design model

The System Requirements

Every system has to define two main types of requirements; the functional and non-functional requirements:

4.2.1 Functional Requirements:

1. The system shall work under Windows Operating System.
2. The system shall have a flexible and transparent interface so that the

entire user at the different levels of skill can access and use the system.

3. This system shall focus on crawling, indexing and searching Arabic documents; these operations are achieved for our project.
4. There are many modules in this system where many of these modules require a number of functional requirements. Modules and requirements of store subsystem
- 5- The system shall have the ability to update the it's databases regularly.
- 6- The system shall manage the query over the operation and data flow.
- 7- In the proposed system, some users such as the end user could not modify anything; this type of users can only view just what they need.

4.2.2 Non Functional Requirements:

1. **Performance requirement:** The system must have a high performance by deleting duplicate records from the database and removing old records from it.
2. **Security requirement:** The system must have a high level of security by implementing two or more levels of security.
3. **Portable and efficient.**
4. **Updateable:** The system will be easily because it has to be updated frequently.
5. **The system is comprised of four subsystems:**
 - ✓ Crawler subsystem.
 - ✓ Stemmer subsystem.

- ✓ Rooting subsystem.
 - ✓ Indexer subsystem.
 - ✓ Search Engine web site subsystem.
6. The system will be fast enough to load and to execute, but as it a web application, it sometimes relies at the speed of the internet connection, and may rely at the speed of the PC if it was old. In conclusion system shall be loaded depending on the internet speed, and the PC speed.
 7. **Availability:** The system service available 24/7.
 8. The system shall contain all necessary information required for information seekers.
 9. The system shall utilize the use of the CPU.
 10. **Throughput:** The system retrieves multiple views of the query results.

(4-3): Study Tests

4.3.1 System testing:

Is the systematic assessment of system performance to determine whether goals have been reached or succeeded? It is vital and important part of the implementation phase . it's purpose is to measure the performance and output of a system quantitatively and compute it to the goals established in the planning phase .

If the goals were met, this means that the system was implemented properly . if not , the analysis must be studied or system redesigned

Why testing important?

- To demonstrate to the developer and the customer that the software meets its requirements.
- To discover faults or defects in the software where the behavior of the software is incorrect , undesirable or does not conform to its specification

4.3.2 Verification and validation (V&V):

Verification and validation are not the same thing , although they are often confused Boehm (Boehm ,1979) succinctly expresses the difference between them :

Verification: are we building the right product?

Validation: are we building the product right?

Test case scenario:

Use –case: Search for query

- 1- Information seekers log in the system
- 2- Put the query and select search
- 3- Receive ranking result related to the query

Use-case: collect pages

- 1- Admin log in the system
- 2- Put start site and select start

- 3- The crawler system parsing the link
- 4- Save the URL for the link

Use-case: index document

- 1- System log in the data base
- 2- Retrieve the content of each URL
- 3- Remove stop words
- 4- Stemming the words
- 5- Calculate the weight for each term
- 6- Store into database

Testing method used:

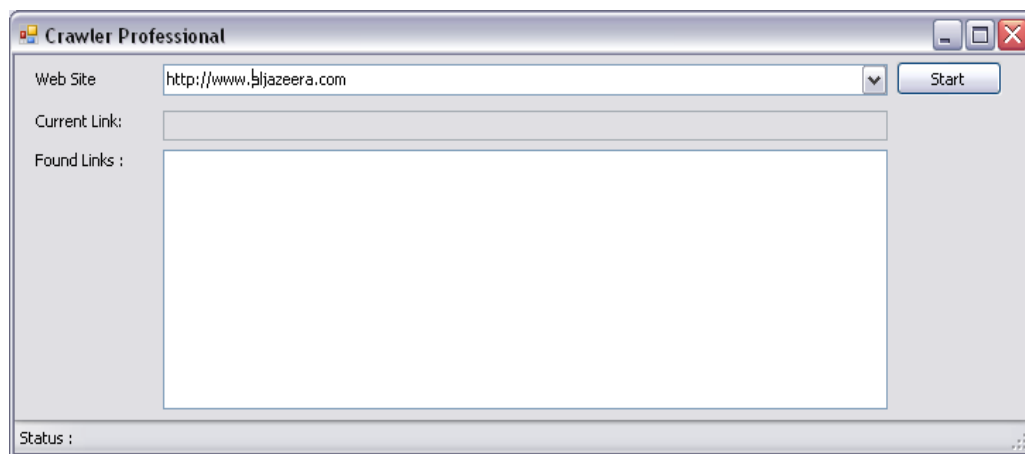
Black box tests are performed to assess how well a program meets its requirements , looking for missing or incorrect functionality. Functional tests typically exercise code with valid or nearly valid input for which the expected output is known . this includes concepts such as boundary values .

Performance tests evaluate response time , memory testing , throughput , device utilization , and execution time : stress tests push the system to or beyond it's specified limits to evaluate it's robustness and error handling capabilities , reliability test monitor

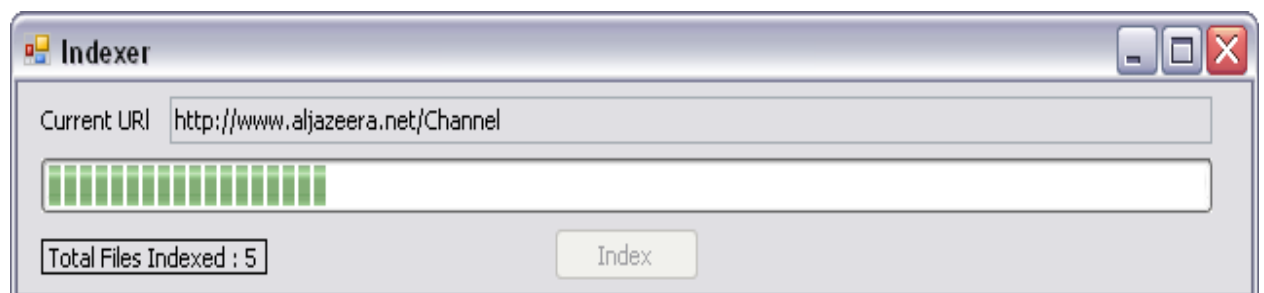
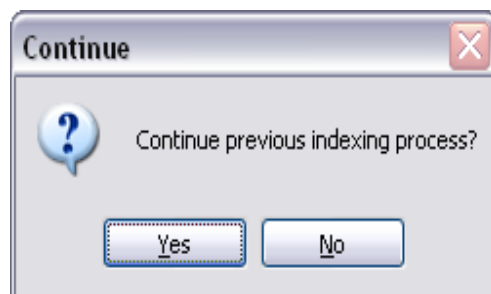
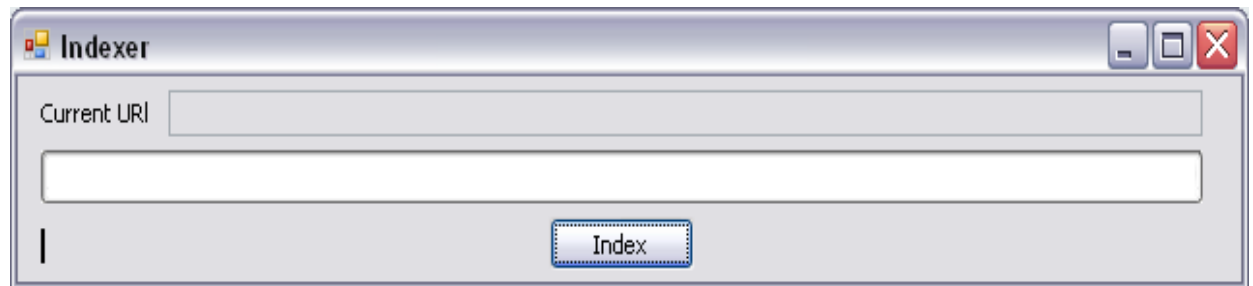
system response to representative user input , counting failures over time to measure or certify reliability .

4.3.3 System integration Test

1. Crawler:



2 Indexer:



search engine:



Chapter Five

Result Discussion and Recommendation

5-1 Result and discussion

5-2 conclusions

5-3 Recommendations

(5-1) Results

- According to the above experiment for the ranking by using Arabic rooting algorithm for indexing the documents and similarity to retrieve the documents to the information seekers we reduce the results in a short list :
- The proposed subsystem crawlers are given a starting set of URLs , The crawler system parsing the link and Save the URL for the link
- The proposed subsystem Indexer extracts all the words from each page, and records the URL where each word occurred that can provide all the URLs that point to pages where a given word occurs Remove stop words and Stemming the words then Calculate the weight for each term then Store into database
- The proposed subsystem The ranking module sorting the results such that results near the top are the most likely ones to be what the user is looking for.
- The proposed system retrieve the most relevant documents of query
- The proposed system ranked the documents according to their relevancy

(5-2): Conclusions

Searching the World-Wide Web successfully is the basis for many of our information tasks today. Search engines are thus increasingly being relied upon to extract just the right information from a vast number of Web pages. The engines are being asked to accomplish this task with minimal input from users, usually just one or two keywords.

We have shown how such engines are put together. Several main functional blocks make up the typical architecture. Crawlers travel the Web, retrieving pages. These pages are stored locally, at least until they can be indexed and analyzed. A query engine then retrieves URLs that seem relevant to user queries. A ranking module attempts to sort these returned URLs such that the most promising results are presented to the user first.

We have shown how the large-scale storage of Web pages in search engines must be organized to match a search engine's crawling strategies. Such local Web page repositories must also enable users to access pages randomly, and to have the entire collection streamed to them.

The indexing process, while studied extensively for smaller, more homogeneous collections, requires new thinking when applied to the many millions of Web pages that search engines must examine. We discussed how indexing can be parallelized, and how needed statistics can be computed during the indexing process.

Fortunately, the interlinked nature of the Web offers special opportunities for enhancing search engine performance. We introduced the notion of Page Rank.

(5-3): Recommendations

According to the experiment model and the previous study

- We recommend to creation of new Arabic rooting algorithm, have a powerful, and we need to implement a new algorithm to give better results.
- We recommend to creation of optimization for ambiguous query

Future Enhancement

A substantial amount of work remains to be accomplished, as search engines hustle to keep up with the ever expanding Web. New media, such as images and video, pose new challenges for search and storage. We have offered an introduction into current search engine technologies, and have pointed to several upcoming new directions.

- The search engine will include the English language
- The ability to searching for image

REFERENCES

- 1- Andrieu, O. (2009), *Re'ussir son re'fe'rencement Web*, 2e` e'd., Eyrolles, Paris.
- 2- Arasu .A, Novak. J, A. Tomkins, and Tomlin..J 2002.” PageRank computation and the structure of the web: Experiments and algorithms”, In Proceedings of the Eleventh International World Wide Web Conference, Poster Track.
- 3- Berman, R. and Katona, Z. (2010), “The role of search engine optimization in search”, TELECOM ParisTech, University of California, Berkeley.
- 4- Bernard J. Jansen , Amanda Spink , Judy Bateman , and Tefko Saracevic. 2009 “Real life information retrieval”
- 5- Bernard J. Jansen , Amanda Spink , Judy Bateman , and Tefko Saracevic. 1998 Real life information retrieval: a study of user queries on the Web, ACM SIGIR Forum, v.32 n.1, p.5- 17, Spring
- 6- Bharat .K. and M. R. Henzinger 1998 . Improved algorithms for topic distillation in a hyperlinked environment. In Proceedings of the ACM-SIGIR.
- 7- Boone, T., Ganeshan, R., 2007. “The frontiers of eBusiness technology and supply chains” , Journal of Operations Management, Vol. 25, p. 1195-1198
- 8- Broder, S. Glassman, M. Manasse, and G. Zweig. 2008 “ Syntactic clustering of the web “ In Proceedings of 6th International World Wide Web

Conference (WWW6) 1997. International World Wide Web Conference (WWW6),.

9- Chaffey, D, 2002, “ E-business and e-commerce management: strategy, implementation and practice”, 4th ed., Pearson Education Limited.

10- Chakrabarti .S, B. Dom, D. Gibson, J. Kleinberg, P. Raghavan, and S. Rajagopalan 1998..” Automatic resource compilation by analyzing hyperlink structure and associated text”. In Proceedings of the Seventh International World Wide Web Conference.

11- Charles L. Viles and James C. French 1995. “Dissemination of collection wide information in a distributed information retrieval system”. In Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval,

12- Craig E. Wills and Mikhail Mikhailov 1999. “Towards a better understanding of web resources and server responses for improved caching. In Proceedings of the Eighth International World-Wide Web Conference”.

13- Chakrabarti .S,. van den Berg, and Dom .B. 1999.” Focused crawling: A new approach to topic-specific web resource discovery”. In Proceedings of the Eighth International World Wide Web Conference.

14- Chien-Kang Huang, Lee-Feng Chien, and Yen-Jen Oyang 2008. “Relevant term suggestion in interactive web search based on contextual information in query session logs”.

- 15- Cutting DR, Karger DR, Pedersen JO, Tukey JW. Scatter/Gather 1992” a cluster-based approach to browsing large document collections” In: Proceedings of the 15th Annual International ACM/SIGIR Conference, Copenhagen,.
- 16- Damanpour, F., 2001. E-business, e-commerce, evolution: perspective and strategy. Managerial Finance, Vol. 27(7), p. 16-33. Available through: Emerald Management Xtra 111 Database. [Accessed 13 April 2011]
- 17- F.C.Cheong1996InternetAgents:Spiders,wanderers,Brokers,andBots,NewRiders,publishing,Indianapolis,Indiana,USA.
- 18- FragforNet (2006), “Understanding biodiversity loss: a workshop on forest fragmentation in South America”, available at:
www.fragforNet.grenoble.cemagref.fr/research-activity/conferences_workshops/events/introduction (accessed on 5 August 2010).
- 19- Golub and Loan. Matrix Computations 1996. The Johns Hopkins University Press, Baltimore,.
- 20- Grimmett and Stirzaker 1989. Probability and Random Processes. Oxford University Press.
- 21- Haveliwala 2002. Topic-sensitive PageRank. In Proceedings of the Eleventh International World Wide Web Conference

World Wide Web Conference, 1999.

22- Haveliwala 1999. Efficient computation of PageRank. Stanford University Technical Report,.

23- Haveliwala and S. D. Kamvar 2003. The second eigenvalue of the Google matrix. Stanford University Technical Report,.

24- Heydon and .Najork. Mercator: 1999 "A scalable, extensible Web crawler". World wide web2 (4):219-229,.

25- Hirai, Raghavan, Garcia-Molina, and. Paepcke. WebBase 2000: A repository of web pages. In Proceedings of the Ninth International World Wide Web Conference,.

26- Isık, Öykü, (2010), "Business Intelligence Success: An Empirical Evaluation of the Role of BI and search engine optimization", Unpublished thesis, University of North Texas.

27- Jeh and Widom 2003. Scaling personalized web search. In Proceedings of the Twelfth International World Wide Web Conference,.

28- Jesus Mena. "Investigative Data Mining for Security and Criminal Detection", First Edition, [amazon.com/Investigative- Mining-Security-Criminal-Detection/dp](https://www.amazon.com/Investigative-Mining-Security-Criminal-Detection/dp).

- 29- Jiawei Han, Micheline Kamber. “Data Mining: Concepts and Techniques”, Morgan Kaufmann Publishers, Champaign: CS497JH, fall 2001, www.cs.sfu.ca/~han/DM_Book.html
- 30- Joachims T. Optimizing Search Engine using Clickthrough Data. In Proceedings of the ACM Conference on Knowledge Discovery and Data Mining, 2002.
- 31- Kamvar, Haveliwala,. Manning, and Golub 1999. Exploting the block structure of the web for computing PageRank. Stanford University Technical Report,.
- 32- Kamvar,. Haveliwala, 2003. Manning, and Golub. Extrapolation methods for accelerating Page Rank computations. In Proceedings of the Twelfth International World Wide Web Conference,.
- 33- Kleinberg. J. 1998. Authoritative sources in a hyperlinked environment. In Proceedings of the ACM-SIAM Symposium onDiscrete Algorithms,
- 34- Leuski A. Evaluating document clustering for interactive information retrieval. In: Proceedings of the 2001 ACM CIKM International Conference on Information and Knowledge Management, Atlanta, Georgia, USA, November 2001. p. 33–44.
- 35- Nicolas J. Belkin, Helping people find what they don't know, Communications of the ACM, v.43 n.8, p.58-61, Aug. 2000..

- 36- Page.L., Brin. S., Motwani, and. Winograd. 1998 The PageRank citation ranking: Bringing order to the web. StanfordDigital Libraries Working Paper,.
- 37- Rafiei and Mendelzon 2000. What is this page known for? Computing web page reputations. In Proceedings of theNinth International World Wide Web Conference,.
- 38- Richardson and Domingos P. 2002. The intelligent surfer: Probabilistic combination of link and content information inPageRank. In Advances in Neural Information Processing Systems, volume 14. MIT Press, Cambridge, MA.
- 39- Patrick Martin, Ian A. Macleod, and Brent Nordin. A design of a distributed full text retrieval system. In Proceedings of the Ninth International Conference on Research and Development in Information Retrieval, pages 131–137, September 1986.
- 40- Sergey Melnik, Sriram Raghavan, Beverly Yang, and Hector Garcia-Molina. Building a distributed full-text index for the web. Technical Report SIDL-WP-2000-0140, Stanford Digital Library Project, Computer Science Department, Stanford University, July 2000. Available at <http://www-diglib.stanford.edu/cgi-bin/get/SIDL-WP-2000-0140>.
- 41- Sergey Melnik, Sriram Raghavan, Beverly Yang, and Hector Garcia-Molina. Building a distributed full-text index for the web. In Proceedings of the Tenth International World-Wide Web Conference, 2001.
- 42- S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. In Proceedings of the 7th World Wide Web Conference (WWW7), 2008.
- 43- S.chakraborti, M.van den Crawling: A new approach to topic-specific web resource discovery”. In the 8th International
- 44- SEOmoz.org (2009), “Search engine ranking factors 2009”, available at: www.seomoz.org/article/search-ranking-factors (accessed on 5 August 2010).

45- Tabatabaei, S. (2009). Evaluation of Business Intelligence maturity level in Iranian banking industry, Unpublished master thesis, Lulea University of technology.

46- T. Fagni, F. Silvestri, S. Orlando, and R. Perego. Boosting the performance of web search engines: Caching and prefetching query results by exploiting historical usage data. ACM TOIS (Transactions on Information Systems), 18:21–36, 2006.

47- Y. Cao, J. Xu, T.-Y. Liu, H. Li, Y. Huang, and H.-W. Hon. Adapting ranking svm to document retrieval. In Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval, pages 186{193, 2006.

48- Y. Zhao and G. Karypis. Empirical and theoretical comparisons of selected criterion functions for document clustering. Machine Learning, 55(3):311{331, 2004.

<http://www.cs.waikato.ac.nz/ml/WEKA/>, online link [dated 2009].

<http://www.oracle.com/technology/products/bi/odm/index.html>

http://www.pentaho.com/products/data_mining/, online link [dated 2010].

<http://www.liacs.nl/~joost/DM/knowledge.htm>

WEKA Machine Learning Software at <http://www.cs.waikato.ac.nz/ml/WEKA/> .

49- Zamir O, Etzioni O, Madani O, Karp RM. Fast and intuitive clustering of web documents. KDD '97, Newport Beach, CA, 1997. p. 287–90.

50- Zdravko Markov, Ingrid Russell. An Introduction to the WEKA Data Mining System

Appendices

The system used tow DLL file the dynamic library link

First one crawler .DLL that's do the following operation

- **Take the fist URL from the administrator**
- **Visit this URL and extract all link inside it**
- **Saving the document related of the collected URL into database**

Second the rooting algorithm also its DLL file this algorithm do the following tasks

Return the word to it's root according to some structure

The structure of the rooting algorithm as shown bellow

Algorithm : Root extraction for third-based verbs

The proposed algorithm works by executing the following steps:

1. Accept the words from the file.
2. If the word consists of less than three words, then remove the word from the document; it is considered as a stop word.
3. Test the word's letters and insert any letter that is matched with any letter of the word "سألتمونيها" to a temporary array named as Temp_Array. Other words are stored in an array called Root_Array.
4. if the letter is ئ and is followed ٰ then insert the letter in the Root_Array as ي letter.
5. Do While (the number of elements in the array is less than 3)
6. If the first letter is م and the last letter is ن then store the letter ن in the Root_Array.
7. If the number of elements of the Temp_Array is more than two AND the last letter in the word is ن, and the second last element is one of the following characters (ي, و, أ); store the third last word of the Temp_array in the Root_Array.
 - a. If the number of elements of the Temp_Array is more than two elements, and the last element of that array is أ and the second last element of the array is و AND the last

letter of the original word is ا then store the third last word of the Temp_array in the Root_Array.

b. If the number of element of the Temp_Array is more than three AND the last letter of that array is Temp_Array is ت or ة AND the second last letter of that array is ا AND the third last letter is ي then store the fourth last letter of the Temp_Array in the Root_Array.

c. If number of the word's letters in the Temp_Array is more than two letters and the last letter was ة or ا and the second last letter is ي and store the third last element of the Temp_Array in the Root_Array.

d. If the number of elements of the Temp_Array is more than one element and that element is one of the following characters (ة, ه, ت, ا, و, ي) and the letter is the last character in the original word then remove that letter from the Temp_Array.

e. If the number of elements of the Temp_Array is more than one element and that element is one of the following characters (ا, و, ي, ى) and the letter is the last character in the original word then store the second last letter existed in the Temp_Array in the Root_Array.

f. If the number of elements of the Temp_Array is more than two characters AND the last letter of that array is ت AND the second last letter is ا and the letter ت is the letter of the original word. Store the third last letter in the Root_Array.

g. Else store the last letter of the Temp_Array in the Root_Array.

h. Arrange the letters of the Root_Array as appeared in the original word.

Names of arbitrators

No.	Name	Specialization	University
1	DR. Mohammed AL-Neiamee	BUSINESS Administration	MEU
2	DR. Hamza Khreem	BUSINESS Administration	MEU
3	DR. Raed Hnandah	BUSINESS Administration	MEU
4	DR. Souod Mhameed	BUSINESS Administration	MEU