CRAWLING THE WEB USING APACHE NUTCH AND LUCENE

A THESIS SUBMITED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED
SCIENCE OF
ÇANKAYA UNIVERSITY

BY

NIBRAS ABDULWAHID

IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE
DEGREE OF
MASTER OF SCIENCE
IN
THE DEPARTMENT OF
MATHEMATICS AND COMPUTER

JULY 2014

Title of Thesis : **Crawling the Web Using Apache Nutch and Lucene**

Submitted by **Nibras ABDULWAHID**

Approval of the Graduate School of Natural and Applied Sciences, Çankaya University.

Prof. Dr. Taner ALTUNOK
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Prof. Dr. Billur KAYMAKÇALAN
Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis of degree Master of Science (M.Sc.) in Mathematics and Computer Science.

Assist.Prof. Dr. Abdül Kadir GÖRÜR
Supervisor

Examination Date: 31.07.2014

Examining Committee Members

Assist.Prof. Dr. Abdül Kadir GÖRÜR (Çankaya University) _____

Assist.Prof. Dr. Reza HASSANPOUR (Çankaya University) _____

Assist.Prof. Dr.Fahd Jarad (Turkish Aeronautical Association Univ.) _____

STATEMENT OF NON-PLAGIARISM PAGE

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name : Nibras ABDULWAHID

Signature :

Date : 31.07.2014

ABSTRACT

CRAWLING THE WEB USING APACHE NUTCH AND LUCENE

ABDULWAHID, Nibras

M.Sc., Department of Mathematics and Computer Science

Supervisor: Assist.Prof. Dr. Abdül Kadir GÖRÜR

July 2014, 61 Pages

The availability of information in large quantities on the Web makes it difficult for user selects resources about their information needs. The good link between the internet users and this information is Search engine. Search engine is kind of Information Retrieval (IR). It works on data collection from the Web by software program is called crawler, bot or spider. Most of Search Engines users don't know the mechanism of action the Search Engine, like how Search Engine works and how it catch information in the Web and how it rank the results to users. For this reason in this thesis used the open-source Search Engine is researched in detail.

In this study, we used each of (Apache Nutch and Lucene) to clarify work of Web crawling open source. They are released under the Apache Software Foundation. Nutch is a web Search Engine working to search and index Web Pages from the World Wide Web (WWW). Nutch is based or built on top of Lucene. It uses in the information retrieval technology. It has more software libraries to indexing of large-size data. Lucene doesn't care about information existing in the Web, like PDF, TEXT, and MS Word. It is working to indexing these documents and convert them to the data can be utilized. The benefit of using both Nutch and Lucene in this study, they are free and we can their development. The Nutch and Lucene are written by Java language, it is a computer programming language. Furthermore, we used Tag Cloud Technology to analysis and view the Lucene content or its index.

Keywords: Web Crawling, Open Source Web Search Engine, Tag Cloud, Apache Nutch, Apache Lucene.

ÖZ

APACHE NUTCH VE LUCENE KULLANARAK WEB TARAMA

ABDULWAHID, Nibras
Yüksek Lisans, Matematik-Bilgisayar Bölümü
Tez Yöneticisi: Yrd. Doç. Dr. Abdül Kadir GÖRÜR
Temmuz 2014, 61 Sayfa

Webde yer alan geniş boyuttaki bilgilerin varlığı, kullanıcıların ihtiyacı olan bilgiyi seçmesini zorlaştırmaktadır. Bu bilgiler ile internet kullanıcıları arasındaki bağlantı yolu arama motorlarıdır. Arama motorları. Crawler, bot veya örümcek adı verilen yazılımlar aracılığıyla web'deki veri koleksiyonları üzerinde çalışır. Birçok arama motoru kullanıcısı arama motorlarının çalışma mekanizmasını bilmezler. Örneğin arama motorları nasıl çalışır veya web üzerinde bilgiyi nasıl yakalar yahut bilgiyi nasıl sıralar. Bu çalışmada açık kaynak tabanlı arama motorlarının nasıl çalıştığını detaylı incelenmiştir.

Bu çalışmada, açık kaynak kod tabanlı Web Crawler programlarını izah ederken apache nutch ve lucene yazılımlarını tek tek kullanılmıştır. Bunlar Apache yazılım kurumu tarafından yayınlanmıştır. Nutch bir web crawler olup, world wide web üzerinde indeksleme yapabilmektedir. Nutch bir lucene mimarisi üzerinde geliştirilmiştir. Bilgi erişimi teknolojileri kullanır. Büyük boyuttaki verileri indeksleyebilmek için birçok yazılım kütüphanesi mevcuttur. Lucene web üzerinde var olan PDF, TEXT veya MS WORD gibi bilgiler ile ilgilenmez. Bu dökümanları indeksleyerek, faydalı olabileceği türe dönüştürür. Bu çalışmada Nutch ve Lucene'nin birarada kullanılmasının faydası, birbirinden bağımsız olmalarının yanısıra Nutch ve Lucene'nin ikisinin de Java ile geliştirilmesidir. Ayrıca Lucene içeriğini veya indeksini görüntülemek ve analiz edebilmek için Tag Cloud Technology'i kullanılmalıdır.

Anahtar Kelimeler: Web Crawling, Açık Kaynak Kodlu Web Arama Motoru, Tag Cloud, Apache Nutch, Apache Lucene.

ACKNOWLEDGEMENTS

I would like to express my gratitude toward Assist. Prof. Dr. Abdül Kadir GÖRÜR for his supervision, special guidance, suggestions and encouragements through out of my thesis progress and development. I would like to thank to Prof. Dr. Taner ALTUNOK for all of his supports.

I would like to appreciate to the Minister and the Employees of Ministry of Higher Education and Scientific Research who have helped enriching my knowledge and for all their support and encouragement. Finally I would like to thank my Husband and my family for all their support and belief in me and for being a knack boosting morale during rough times.

TABLE OF CONTENTS

STATEMENT OF NON PLAGIARISM … iii
ABSTRACT. iv
ÖZ… v
ACKNOWLEDGEMENTS. vi
TABLE OF CONTENTS. vii
LIST OF FIGURES. xi
LIST OF TABLES. xiv
LIST OF ABBREVIATIONS. xv

CHAPTERS:

1. INTRODUCTION. 1
1.1. Web Search: Users Face Problems with Search on the Web. 1
1.2. Aim of the Thesis... 1
1.3. Thesis Structure... 2
2. BACKGROUND. 3
2.1. What is the Internet. 3
2.1.1. Getting to the Internet: (Browsers). 4
2.1.2. The Internet: Uniform Resource Locator (URLs). 5
2.1.3. Other Typical Domain Names Include the. 6
2.1.4. Where did the Internet Come From. 6
2.2. History of World Wide Web (WWW). 6
2.2.1. What is the World Wide Web (WWW). 7
2.2.2. The Web and the Internet are Different. 7
2.3. Question: What is the Difference Between the Internet and Web. 7
2.3.1. What is the Content of the Internet. 8
2.3.2. What is the Content of the Web. 8
3. SEARCH ENGINE AND WEB CRAWLING. 9
3.1. History of Search Engine. 9

3.2. What is a Search Engine (SE). 10
3.2.1. Search Engine System Architecture. 11
3.2.2. Where are we Searching. 11
3.2.3. The Search Engine has three Stages Process. 12
3.3. Search Engine Features and Services. 12
3.3.1. Use the Boolean Logic. 12
3.3.2. What are Boolean Operators. 13
3.3.3. Combining Terms. 13
3.3.4. Search Engines and Boolean Operators. 14
3.4. Web Crawling. 14
3.4.1. Why Crawlers. 15
3.4.2. Working of Web Crawler. 15
3.4.3. Crawling Techniques… 17
3.4.4. The Relationship Between Searches Engines and Web Crawler. .. 17
4. LITERATURE SURVEY. 18
4.1. Introduction. 18
4.2. What are a General Web Crawler and Focused Web Crawler. 18
4.3. First Focused Web Crawling. 19
4.4. Strategies of Focused Crawling. 19
4.5. Algorithms Used in Focused Web Crawlers (FWC). 20
4.5.1. Web Analysis Algorithms. 20
4.5.2. Web Search Algorithms. 21
4.6. More Algorithms of Web Crawling. 24
4.6.1. Depth First Search Algorithm. 24
4.6.2. Genetic Algorithm. 24
4.6.3. Naive Bayes Classification Algorithm. 25
4.6.4. HITS Algorithm. 25
4.7. Google Search Engine and PageRank Algorithm. 25
4.7.1. Google Search Engine. 26
4.7.2. PageRank. 26
4.7.3. Link Structure of the Web. 27

4.7.4. PageRank Votes. 27
4.7.5. High PageRank Backlinks. 28
4.8. Open Source Search Engines and Commercial Search Engines. 29
4.9. Search Process is Mostly Invisible. 29
4.10. Advantages to Use Focused Web Crawling. 30
5. APACHE NUTCH AND LUCENE. 31
5.1. What is a Nutch. 31
5.2. Nutch Architecture. 31
5.2.1. Nutch Crawling Process. 32
5.2.2. Why Use Nutch. 33
5.3. What is a Lucene. 33
5.3.1. Indexing Use Lucene. 33
5.3.2. Creating the Index. 35
6. IMPLEMENTING A NUTCH WEB CRAWLING. 36
6.1. Using Nutch. 37
6.2. Installing the Java Environment. 37
6.3. Selecting a Web Interface. 37
6.4. Installing a Shell Environment. 38
6.5. Web Crawling with Apache Nutch. 39
6.5.1. Preparing Nutch for Crawling. 40
6.5.2. Configuring Apache Nutch Crawl. 41
6.6. Creating a URL List to Fetch. 42
6.7. Performing a Nutch Crawl and Using the Crawl COMMAND. 43
6.8. Show the Result Crawling (Nutch Data). 44
6.9. Searching the Crawl Results 46
6.10. Nutch Web Applications 48
6.11. Use Luke to Analyze the Lucene and Nutch Indexes... 50
6.11.1. Search about keywords in Lucene and Nutch Indexes Use Luke. 50
6.12. Analysis the Lucene Indexing Using Tag Cloud Technology… 51
6.12.1. Analysis the URL and Word Frequency… 52
6.12.2. Analysis the Text and Word Frequency 56

7. CONCLUSION AND FUTURE WORK… 59
7.1. Conclusion… 59
7.2. Future Work... 60
REFERENCES R1
APPENDICES A1
A. INDEX WEB PAGE… A1
B. OPEN SOURCE WEB SITES… B1
C. CYGWIN COMMANDS… C1
D. QUESTIONNAIRE… D1
E. PAGERANK… E1
F. SYSTEM REQUIREMENT..….. F1
G. CURRICULUM VITAE (CV) G1

LIST OF FIGURES

FIGURES

Figure 1 Internet: Collection of millions of computers... 4
Figure 2 The browser: is a part of software (S/W) that runs on our PC... 4
Figure 3 Internet users in the world by geographic regions... 4
Figure 4 This example above explain to us parts of url... 5
Figure 5 The internet contains the world wide web... 8
Figure 6 High level search engine architecture... 11
Figure 7 (A and B).. 14
Figure 8 (A OR B)… 14
Figure 9 (A NOT B).. 14
Figure 10 Web crawler or spider helps search engine to widely index the web pages… 15
Figure 11 Crawling process show us represents the process of fetch and indexing Web pages.. 16
Figure 12 Focused crawling.. 19
Figure 13 Fetch and reject web pages according type of web crawler.. 19
Figure 14 URLs structure… 20
Figure 15 Breadth-First search algorithm.. 21
Figure 16 The Breadth-First search queue.. 22
Figure 17 Breadth-First search code.. 22
Figure 18 The Best-First search queue.. 23
Figure 19 Best-First search code.. 23
Figure 20 Google search engine with order web pages.. 25
Figure 21 Add vote or link to web page.. 26
Figure 22 Backlinks from web pages A and B to web page C.. 27
Figure 23 Web page A has 6 votes and web page B has 4 votes.. 27
Figure 24 Real PageRank.. 28

FIGURES

Figure 25 High PageRank backlinks.. 28
Figure 26 Search process..... 30
Figure 27 Framework of nutch..... 32
Figure 28 Crawl diagram to understand the crawling process… 32
Figure 29 Relationship between lucene and applications.. 34
Figure 30 Document converting…..... 34
Figure 31 Indexing process.. 35
Figure 32 Our system fetch and view result by tag cloud..... 36
Figure 33 Download java product windows x86.. 37
Figure 34 Download "Apache Tomcat" supported the 32-bit windows operating system… 38
Figure 35 Tomcat interface… 38
Figure 36 Installing and updating cygwin environment or packages..... 39
Figure 37 The architecture of nutch crawling… 40
Figure 38 The structure of the "crawl-urlfilter.txt" file …..... 41
Figure 39 Configuration the "nutch-site.xml" file… 42
Figure 40 The urls.txt file…..... 42
Figure 41 Parameters command bin/nutch crawl..... 43
Figure 42 Run crawl command… 43
Figure 43 Urls.txt… 44
Figure 44 Crawl folder.. 44
Figure 45 Directories created… 45
Figure 46 Nutch crawl folders data … 45
Figure 47 Nutch commands… 46
Figure 48 Directory the crawling results…..... 46
Figure 49 Apache tomcat interface… 47
Figure 50 Apache tomcat deploy….. 47
Figure 51 Nutch-1.1 webapps……. 48
Figure 52 Nutch-1.1…… 48
Figure 53 Nutch-1.1 web with hits list…..... 49
Figure 54 Nutch's score explanation page, matching the query "Cankaya"..... 49

FIGURES

Figure 55 Command to read nutch folders content…. 50
Figure 56 Luke result…. 50
Figure 57 Cankaya University web page… 51
Figure 58 Search result about "Üniversite Hakkında" keyword use luke lucene.. 51
Figure 59 Cankaya history web page… 52
Figure 60 List of words….. 52
Figure 61 Python code to indexing words…. 53
Figure 62 Words- frequency…. 53
Figure 63 Stop words list…. 54
Figure 64 Meaningful words and chart… 54
Figure 65 Tag cloud using www.wordle.net... 56
Figure 66 Tag cloud before removing common words… 57
Figure 67 Tag cloud after removing common words …. 57
Figure 68 Tag cloud after removed the black words …. 58
Figure 69 Nutch focused web crawling…. 60
Figure 70 URL with query… 61

LIST OF TABLES

TABLE

Table 1 List of word-frequencies... 55

LIST OF ABBREVIATIONS

URL	Uniform Resource Locator
HTML	Hyper Text Markup Language
PDF	Portable Document Format
MS	Microsoft Office
WWW	World Wide Web
TCP/IP	Transmission Control Protocol/Internet Protocol
S/W	Software
PPT	Power Pint
TXT	Text
DOC	Document
DB	Database
HITS	Hyperlink-Induced Topic Search
BST	Binary Search Tree
FIFO	First-In-First-Out
DFS	Depth First Search
PR	PageRank
JDK	Java Development Kit
HTTP	Hypertext Transfer Protocol
SQL	Structured Query Language
SE	Search Engine
ASF	Apache Software Foundation
PC	Personal Computer
RHG	Red Hat Cygwin
JRE	Java Runtime Environment
DLL	Dynamic Link Libraries

CHAPTER 1

INTRODUCTION

1.1 Web Search: Users Face Problems with Search on the Web

The World Wide Web has become the largest source for the providing of information to Internet users around the world. Because of the considerable increasing of information of the Web makes it difficult for user selects resources about their information needs. User gets this information from the Web by "search engines". At the heart of all "Search Engines" there is a program called "Web Crawler". It works to fetch URLs from the Web in large quantities. For example "Web Crawler" is fetching more topics not relevant, but the user need to fetch few pages of topics relevant.

Open source search engines has more benefits are no cost, can be developed and give the same functionalities of commercial search engines.

1.2 Aim of the Thesis

In this study we used open source search engine is "Apache Nutch", developed by Apache Foundation and it written by java. The Nutch search engine has more advantages are rich crawl, crawling can be biased to fetch "important or related" web pages first and it works up Lucene. Nutch use Lucene to indexing. Lucene open source project and it developed by Apache Foundation. Lucene features are field based indexing and search and use inverted index to store content of crawled documents. Furthermore, Lucene doesn't care about format data in the web, it capable of indexing (HTML, PDF, MS document).

1.3 Thesis Structure

In this section we review thesis structural:-

In Chapter 2, we introduce background about the Internet and the World Wide Web, and history all of them.

In Chapter 3, we present history of search engine, web crawler and the relation between them. Furthermore, we presents how search engine work to retrieval information to users from the WWW and how it sort the result, in this chapter we used PageRank formula of Google to clarify the mechanism the sort the web pages. We present work of web crawler to fetch URLs from the WWW, and work of web crawler and the strategy followed.

In Chapter 4, we introduce overview "Focused Crawler" through previous studies conducted by researchers.

In Chapter 5, we introduce overview about "Apache Nutch and Lucene". We explain Nutch and Lucene architectures and how Nutch and Lucene are working. Furthermore, we explain why used these programs to crawl the web.

In Chapter 6, we introduce requirements for creating an environment for work "Nutch Web Crawling". These requirements are java environment, tomcat server and Cygwin environment, when completed from these requirements then we can run the "Nutch Web Crawler". Finally we display the web crawling result by tomcat server. It has search interface, which user can deal with it easily.

In Chapter 7, we introduce two parts conclusion and future work, in the first part, we have included, the work of each of Search Engine and Web Crawler by use Web Crawler open source and we got functionalities and results close of commercial search engines, and in the second part as you know the Nutch is open source, this means that we can develop it. We are trying to train "Nutch Web Crawling" to become more high-quality to fetch URLs from the web and make it focus in its work to fetch the Web pages from the Web according what we want fetch under specific keywords.

CHAPTER 2

BACKGROUND

2.1 What is the Internet?

We can definition the Internet is a worldwide set of machines or (computer networks) that join very many of state agencies, educational faculties, businesses, and businesses, as shown in the (Figure 1). We can the exchange of information between these the networks in the around the world. Whatever the different computers used in the network [1]. More detail about internet is a global system that includes a wide range of interrelated computers networks that use the standard Internet Protocol Suite (TCP/IP) to cover the serve very many of users worldwide. In the world the internet service connects between very many of computers jointly globally. Which computers in the network can connect with other computers in the other networks they are connected to the Internet? [2]. Despite the differing computers in the world, but users can access any information, and permits your own computer to get pieces of information stored on another computer away, because of standard protocols that allow computers to communicate with each other [3, 4]. Different networks, in terms of the number of computers in the each network. Some networks have thousands of computers. A few networks have only a few computers. They are connecting to the Internet via telephone and cable systems like fiber optic [4]. Huge firms, governments and many organizations own intranets. Intranet is technologies used behind the corporate firewall or in private environment. The computers on an intranet are link to the Internet. However people that are officials at the organization which possess the intranet have an access to it. Other people who use the internet service are incapable of seeing the information on the intranet computers [4, 5]. Figure 1 show us contact the computers by internet.

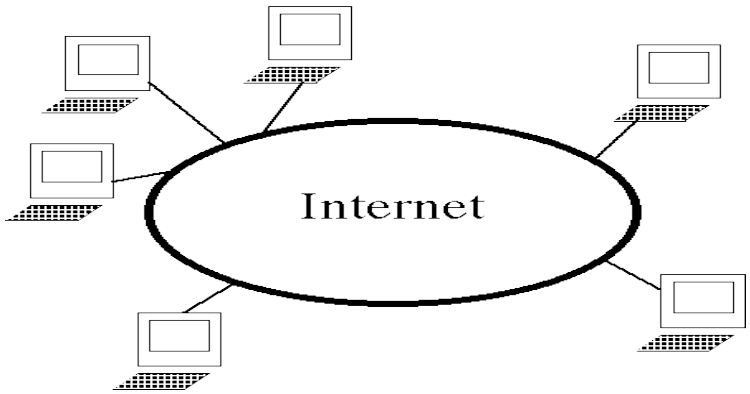


Figure 1 Internet: Collection of millions of computers around the world that all are connected by one another is the internet. We can change information with each other by the internet like E-mail, regardless of the difference computers [6].

2.1.1Getting to the Internet: (Browsers)

A browser is a set of software application. The users use it to display the web pages. The three most well-known browsers are (Google Chrome, Microsoft Internet Explorer, and Firefox) [7], as shown in the Figure 2.

Figure 2 The browser: is a part of software (S/W) that runs on our PC [8].

Figure 3 below shows us Internet users in the World distributors by geographic Regions.

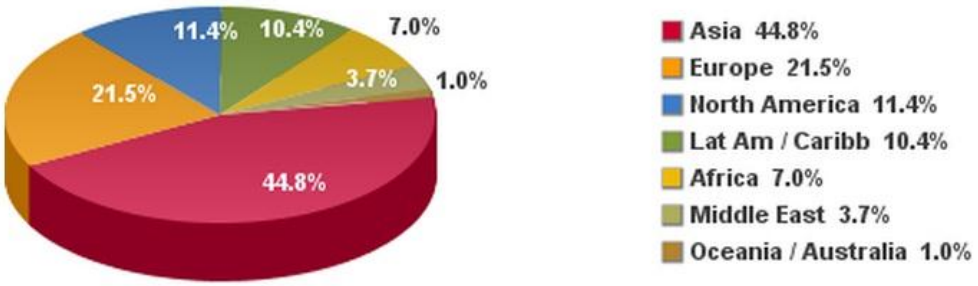


Figure 3 Internet Users in the World by Geographic Regions (Source: http://www.internetworldstats.com/stats.htm accessed on June 30, 2012).

2.1.2 The Internet: Uniform Resource Locator (URLs)

There is an address on the Web for each document like videos, images and text that is called the URL ("Uniform Resource Locator"). Every URL owns sundry parts: the host name, domain name and the protocol.

The URL to locate Shopping is http://www.shopping.com/ [7]. More detail about the URL above:

(http:// and https ://) hypertext transfer protocol identifies the document as a web page. It is the standard used to connect, communicate and transfer data on the Web. Most web browsers will automatically add this prefix to the address. The “s” indicates a “secure” version of http and is usually used by web pages that ask for personal information. (W3) document on the World Wide Web. Some web sites require this. Google host name. (.com) domain name that identifies type of the web page [7], below additional example about URL. Figure 4 below shows parts of URL.

[PROTOCOL]://[HOST]:[PORT]/[PATH][FILE]?[QUERY]
http://edition. cnn.com/search/query=IRAQ

Figure 4 This example above explain to us parts of URL

The "URL" parts have the following meanings [9]:

- The **"PROTOCOL"**: http is a gate to enter to the web
- The **"HOST"**: each URL has a space in the Web Server Side.
- The **"PORT"**: Is the protocol used to communication process
- The **"PATH and FILE"**: it can create via a content management system or either automatically.
- The "QUERY" User Query, about what a user searches the web.

So, each Webpage has unique URL, and it has space in the Web, and Internet users can access to the URLs by use the Web Browser, like Google Chrome, Internet explorer and Firefox.

2.1.3 Other Typical Domain Names Include the [7]:

- (.gov) Abbreviation for the two words (Government Agencies).
- (.edu) Educational institutions.
- (.org) Abbreviation for the word "Organizations" (nonprofit).
- (.mil) Abbreviation for the word "Military".
- (.com) Abbreviation for the two words "Commercial business".
- (.net) Abbreviation for the two words "Network Organizations".
- (.iq) Abbreviation for the word "Iraq".

2.1.4 Where did the Internet Come From?

The first internet has been discover or emerges by the computer network, it has called (ARPANET). Furthermore, developed country like United States of America, also the military sector was the responsible of that invention at the era 1960s. Just a small number of scholars and employers were able to use it and that in period between 1970 and 1980. The year1980, witness approval of the government to use the internet networks at the universities, Schools, libraries, local and state governments, houses. In fact at that time it was not easy to get information from the internet, may you can get only ordinary information by using the computer. In 1980 a British commuter scholar called Timothy Berners-Lee, invents the World Wide Web, as it now known as (WWW) [4].

2.2 History of World Wide Web (WWW)

The W3 has become recently with its history diverse internet users or people in world. A large number of people all over the world utilize the Web daily because of the urgent requirements, Foundations, Ministries and Banks are working to evolve a system is as better as possible that retrieve information the Web, it to contain a wide range of information. Expansion and growth of the World Wide Web (WWW), and of available more information on this network, making difficult for users of this network to determine the sites around specific topics and Topics of benefit. Another growth in the (WWW) includes an increase in the number of users and the amount of several of documents like PDF, PPT, DOC and TXT, making the search and determines interesting topics are difficult. So when the user requests a particular topic, search engines have a difficult challenge for fetch interesting topics and

relevance by user's request. The major and big search engine such as "Google.com" that crawls, more millions of web pages per day takes weeks to crawl the whole web, each page have more links [10, 11, 12].

2.2.1 What is the World Wide Web (WWW)?

The W3 is a big group of web pages, that large software sub of the Internet ad hoc to broadcasting content in the form of (HTML pages). Access the Web through of using free software called Web Browsers like Google chrome, internet explorer and etc.
It was Born in 1989, the Web is based on http, the language that lets us to access (hyperlink) to any other public (web page). It is quite known that there are more billion public web pages of public Web pages nowadays on the Web [4, 13, 14].

2.2.2 The Web and the Internet are Different

We can make comparison between the web and the internet, such kind of comparison can be given through comparing delivery service and streets the cars which are used to deliver service use streets to transport the cargo from place to place. The delivery service is similar to the Web. the internet is as the same as the streets .the information of traffic transfers from the web by using the internet sites include locations in the web by utilizing particular computer programs, internet users create the sites .the web servers is the name for the sites which are stored on the computers, the web pages may include more documents such as video, text, images and so on "uniform resource locator" is denoted for every web page. http://www.shopping.com/ is an instance. As "computer experienced personages" believe that the internet become quite famous simply because of the Web, the internet is more difficult to than the web .More than 80% of all traffic on the internet high way came from the Web, by the end of 2000 [4].

2.3 Question: What are the Difference Between the Internet and Web?

The relationship between the Internet and the Web are very close. The Internet is the big space like (container), and the Web is a part inside the container. To clarify more of the relationship between the "Net" and the "Web", the Web is the most popular dish on the menu and the Net is the restaurant [15], below the definition of "Internet" and "Web":

2.3.1 What is the Content of the Internet?

The Internet is a Big Collection of "Computers and Cables", connected with each via wires made of copper and wireless connections [15].

2.3.2 What is the Content of the Web?

The WWW is a large group collection of interrelated documents and another resource, linked via hyperlinks and Uniform resource reference locator or URLs, that great software group of the Internet devoted to broadcasting content in the form of HTML pages. The "WWW" is one of the service accessible via the internet, along with various others including email file sharing, online gaming and others applications. However, the "internet" and the "Web" are commonly used interchangeably in non-technical settings. [15]. Figure 5 shows the contents of the Internet.

Figure 5 The internet contains the world wide web [15]

CHAPTER 3

SEARCH ENGINE AND WEB CRAWLING

3.1 History of Search Engine

"Archie" is a faster tool used to search on in the internet. At "McGill University" in Montreal, the "Archie" was created by Alan Emtage, who is a student at that University. The purpose of this program is to download the directory listings of all the files available at public anonymous File Transfer Protocol (FTP) sites, creating a database and making information of file names which are searchable. At the "University of Minnesota", in 1991 "Gopher" was created by the student y Mark McCahill. The mechanism of action of each program, "Gopher" indexed plain text documents, "Archie" indexed file names. There are two other programs "Veronica" and "Jughead". The programs search the files stored in Gopher index systems. Wandex was the first search engine created in 1993, by "Matthew Gray" using a Perl script. This search engine was usable for the World Wide Web. Aliweb is another search engine which also appeared in 1993 is (Archie like Indexing for the WEB). The first Web search engine to provide "full text" search was "WebCrawler", 1994. It different unlike its predecessors, WebCrawler let users put query and search for any word in any web pages, "this became the standard for all major search engines ever since". It was also the first one popular among users. Also in 1994, Lycos (which started at Carnegie Mellon University) appeared and became an important commercial endeavor. Soon after, many search engines appeared and became popular in world. These included Info seek, Excite, Northern Light, Income, and AltaVista. To some extent, they competed with popular directories such as Yahoo. Then, the directories added on search engine technology for greater or to improve functionality. In the late 1990s, search engines were also known for the Internet investing widely. Many companies entered the market widely, with record gains during their initial public offerings. "Some have taken down their public search

engine, and are marketing enterprise only editions, for example (Northern Light)". Its success was based partly on the concept relationship with other Web site through. Interdependence and PageRank that uses the premise those good web pages are pointed to more web pages than others. Google, a search engine, has the most popular interface for users in the world. It began in January 1996 as a research project by (Larry Page and Sergey Bring) when they were both PhD students at (Stanford University) in Stanford, California. Google is useful for users of internet, to find all they needed with their related topics. In 2005, Google indexed more Web pages than other search engines, which are about 8 billion pages. It also offers to for its users a number of Web services, like Google Maps, online translation, search images and videos. In 2002, Yahoo! acquired Income (its corporation was a California company that provided software for Internet service providers) and in 2003, Yahoo! acquired Overture, which owned (AlltheWeb and AltaVista). Although, owning its own search engine, Yahoo kept at first, continued using Google search engine to provide its users with information result. In 2004, Yahoo! presented its own search engine which consists of an integrated set of technology. MSN Search is a search engine owned by Microsoft Company. It previously depended on others for its search engine to show listings. In 2005 it started showing its own search results, collected by its own spider. Many other search engines tend to be gates that show the results from another types search engine [16, 17, 18].

3.2 What is a Search Engine (SE)?

The growth in the information in the Web, led to create like or bridge between the Internet users and these information. Search Engine Solve this problem. Search Engine is a software program, help the Internet users to search and access to the data existing in the Web, like PDF, Video, Images, MS word and etc. Search Engine works according the user query, which means search engine take the keyword from the user and search in the Web, then search engine return the results to users in the form of hits. Search engine uses the special software program called Web Crawler. It works to visit the URLs in the Web and index them. So when user search about something in the Web via search engine, just put his/her keyword then search engine send the Web Crawler to collect the information from the data base. Search engine a good tool in the information Retrieval. Available search engines in the World like Google, Yahoo, Bing and etc. And also available open-source search engine like

Nutch, IXE, Web Glimpse and etc. [19, 20, 21, 22, 23]. Search engine has three parts are:

- Crawler, Spider or robot.
- Search engine software.
- Index, catalog or database.

3.2.1 Search Engine System Architecture

In this section we will talk about how the whole system of a search engine works. Before a search engine to answer your request where a files or documents are, it must be found. To search and find information on the hundreds of millions of Web pages that exist in W3, a typical search engine provides or employs special software robots, called spiders or crawlers, to build lists of the words found on Web sites, then search engine displays the results about the information on your keywords. When a crawler is end of the building its lists, the process is called Web crawling [24]. Figure 6 below shows High Level Search Engine Architecture.

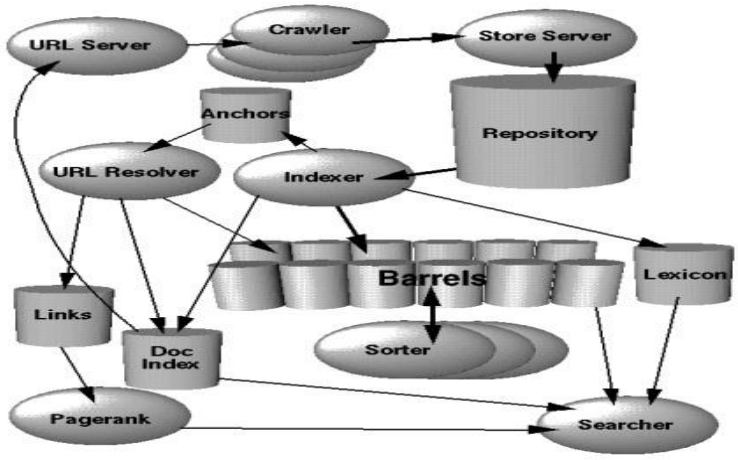


Figure 6 High level search engine architecture (Brin and Page, 1998) [24].

3.2.2 Where are we Searching?

When we use search engines to search on information about a particular topic, actually we are not searching on the Web online, but we are dealing with the search engine's database (off line) of Web pages information. Search engines are working to take copies, analysis and organize all of the information available on the Web into their own efficient database via the Web Crawler, and make this information ready for use by search engines users. This information is ready when requested by the

users of search engines. Search engines can not to fetch and index all the information or Web Pages in the Web, so search engines working to include partial or the largest possible size of Web pages. When we clicking on the hyperlink listed in the search results led us from the search engine host, to the actual Webpage existing in the Web. Perhaps each link leads or takes us to another link.

3.2.3 The Search Engine has Three Stages Process

Building a search engine database is three-stage processes are crawl, index, and information retrieval. Search engines must and important find web pages, organize the information for fast retrieval, and serve up the information based on your request. This is an ongoing process because the search engines databases are constantly change and develop. This constant change happening result to update web pages. In part a Web crawler will talk about these three important stages of in the search engines [25].

1. Stage 1 – Crawling.
2. Stage 2 – Indexing.
3. Stage 3 – Retrieving.

3.3 Search Engine Features and Services

Search engines displays search results to the user, according to input keywords that describe an information need. When users using a search engine to get information, by entering more than one keywords, the space between the keywords has a logical meaning that directly affects the search results. This process is known as (default syntax). Example: when we use one of the search engines like Alta Vista, Info seek and excite, a search of word ‘bird migration’ means that the searcher will get list of documents that contain either word’ Birds’ and the word ‘migration’ or both. The space between the keywords defaults to the (Boolean OR). This is probably not what the searcher will get list of documents that contain both the keywords’ birds’ and ‘migration’. Search engine return list of search results in "schematic order". Most search engine use different standards to contract a term relevancy rating of each hit and present the list of search results in this order. When users use logical words to search about some things in search engines and they use words like (AND, OR, NOT), they are Boolean features that allow retrieval list of

documents that contain all the keywords (AND), any of the keywords (OR), exclude of some words (NOT), or mix of these Boolean operators. The proximity feature searches for phrases or successive words (usually simple search can do this if the words are surrounded by double quotes), example: search about "bottle opener" the result is anything with exact phrase "bottle opener". The search can be done only in particular fields, when a user type into the search box, such as URLs likes (www.Cankaya.edu.tr) rather than of using (Cankaya University). Limits can be imposed on the type of retrieved web pages: date, language, file types. Some search engines also provide or offer services (additional applications): news directories, image and video search, maps (such as Google Maps and Yahoo!), language tools (translation tools in all languages), newsgroup search, and other specialized searches [16, 26, 27].

3.3.1 Use the Boolean Logic [27]

• AND keyword (Requires the entire search terms to appear on a web page).
• OR keyword (Allows any of the search terms to appear on a web page).
• NOT keyword (Requires a search term to not be present on a web page).

3.3.2 What are Boolean Operators?

To linking phrases or two or more words when searching databases or using an internet search engine such as Google, in this case you need to linking tool called "Boolean Operators". Using these operations make searching large databases much more precise [28]. The three main operators are: AND, OR and NOT.

3.3.3 Combining Terms

The use of words and phrases to search for related topics via search engines, maybe we get the results are not relevant to your search needs. For this reason search is a waste of time to review a long list of citations for only a few which are directly relevant to your subject. Linking words and phrases to define the concepts of a search more clearly can reduce the lost time and find related topics, some examples about (AND, OR, NOT) [28]:

- (AND) operator work to the finds items which include both groups or search terms. Use and to combine two or more words or phrases. Figure 7 below shows "AND" operation.

Figure 7 (A and B)

- (OR) operator through use of this operator make broadens the search by finding keywords or items that contain at least one of the search terms anywhere in the record. Figure 8 below shows "OR" operation.

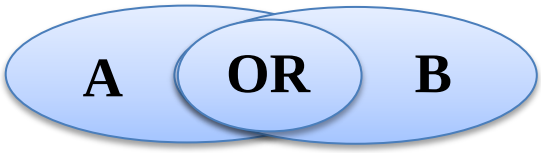


Figure 8 (A OR B)

- (NOT) operator works to find the first word or phrase but excludes any that also contain the second. Figure 9 below shows "NOT" operation.

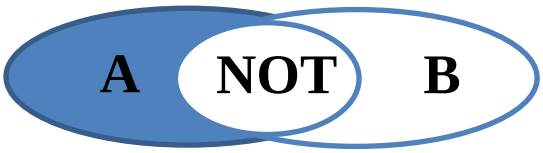


Figure 9 (A NOT B)

3.3.4 Search Engines and Boolean Operators

Google and many search engines also use "Boolean Operators". In "Google" Advanced Search they appear as [28]:

- Use AND with all the words.
- Use OR with at least one of the words.
- Use NOT without the words.

3.4 Web Crawler

Inside each search engine there is what is known as Web Crawler or Spider. Web Crawler is a software program used by search engine to crawl the Web. Web Crawler known a spider or robot. The Web Crawler starts the fetch from the URL seed like Cankaya .edu.tr or http://edition.cnn.com/. Web Crawler works to inject all URLs from the seed URL, and put them in the queue to fetch one by one, search engine follows strategy to fetch the URLs in the queue, like Breadth-First Search algorithm

(First in First out FIFO). After Web Crawler visited to all URLs in the queue, Web Crawler works or put all URLs in the frontier. After that Web Crawler works to extract the important information from the Webpages like Title, Keywords, date and metadata. Web Crawler re-visit to check each Webpage in the frontier if any change on the Webpage like put new hyperlink or put new keywords and etc. Web Crawler does not care about information the Web, it has the ability to catch all information like Videos, Images and other information in the Web. Web Crawler works to copies the pages, which it visited, to speed the retrieval the information to user by search engine. [29, 30].

3.4.1 Why Crawlers? [31]

- Internet has a wide expanse of information, in various fields (e.g., Education, marketing, etc.).
- Finding relevant "information requires" an efficient mechanism.
- Web crawlers provide that scope to search engine, in detail.
- Web crawler or Spider helps search engine to widely index the Web pages existing in World Wide Web.

  Figure 10 below shows result of crawl.

Figure 10 Web crawler or spider helps search engine to widely index the web pages.

3.4.2 Working of Web Crawler

The Web crawler is works on a specific path, it beginning with start set of URLs known as root or seed URLs, like tree. Web Crawler works to inject the URLs from the seed URL, and put them inside queue, after that it works to fetch each URL from the queue and put them inside frontier. After the fetch stage Web Crawler works to extract all the information from the Web Pages and store on local desk

(storage unit), to take advantage of them and make them more flexible to users. Web Crawler re-crawls or renews the visit to the URLs in the frontier, to check any change of each the Web Page, like put new link or put new topic. The extracted URLs from the downloaded Web Page are confirmed to know whether their related documents have already been downloaded or not. If they are not downloaded, the URLs are trying again assigned to Web Crawlers for more downloading. This process is repeated until no more URLs are missing for downloading. Millions of "Web Pages" are downloaded per day by a "Web Crawler" to complete the goal [30]. Figure 11 shows us the web crawling processes.

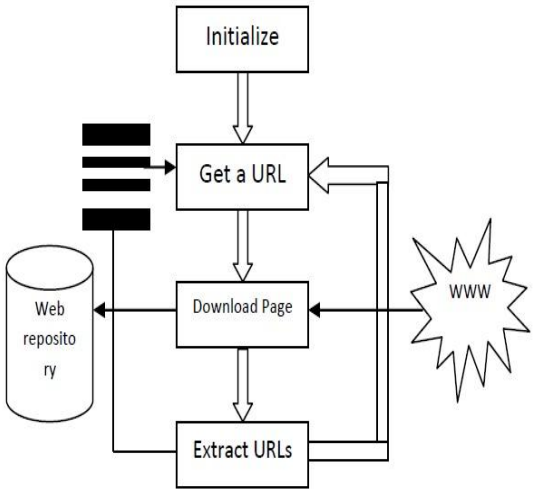


Figure 11 crawling process show us represents the process of fetch and indexing web pages [30]

The working of a "Web Crawler" may be discussed as follows:

- Selecting a starting seed "URL" or "URLs".
- Adding it to the frontier.
- Now selection the URL from the frontier.
- Fetching the "web page" corresponding to that "URL".
- Parsing that "web page" to find new "URL links".
- Adding all the newly found "URLs" into the "Frontier".
- Go to "step 2" and repeat till the "Frontier" is empty.

Thus a "Web Crawler" will repeat keep on inserting newer "URLs" to the database (DB) store of the "Search Engine". So we can see that the major function of a "Web Crawler" is to insert new links into the frontier and to choose a "new URL" from the frontier for more processing after every recursive step [30].

3.4.3 Crawling Techniques

"Web Crawlers" used Crawling Techniques, below few from it [30] [32]:

A- General Purpose Crawling

A general purpose Web Crawler gathers as many web pages as it can from a particular set of URL's and their links. In this case, the crawler is can to fetch very many web pages from various locations. In this type of crawlers, "general purpose crawling" network bandwidth and speed can slow down because it is fetching all the web pages.

B- Distributed Crawling

In this type of crawlers, the "distributed crawling", is a multiple processes is used to fetching and download the "Web Pages" from the Web.

C- Focused Crawling

A "Focused Crawler" is designed to fetching specific web pages related by user query, in this case can reduce the amount of network traffic, and downloads and no loss in time. In "Focused Crawling", the goal is to adapt the behavior of the "Search Engine" to the requirements of a user. "Focused Crawler" is to eclectic look for pages that are appropriate to a pre-defined set of matters, this is purpose of the "focused crawler". It fetch web pages only the relevant area of the "Web" and leadership to important store in place of storage like hardware and network resources.

3.4.4 The Relationship Between Searches Engines and Web Crawler

Before a search engine display you the search result or the information, it must be found and search engine indexes all the contents or words of web pages and adds them to a database, then follows all hyperlinks and indexes and adds that information also to the database. The search engine employs special software program, called crawlers, internet bot and spiders it works inside heart for search engine, it works to find topics or words on very many of Web pages that exist in the Web. When a robot is building its lists of web pages, the method is called Web crawling. Finally, Web crawler provide that scope to search engine and Web crawler, as an important part of search engine, mainly answers for collecting web resources to store, to be the speed of answer and retrieval of data to users faster.

CHAPTER 4

LITERATURE SURVEY

4.1 Introduction

A challenge faced by search engines, the expansion of the World Wide Web, and the large number of web pages, resulting in a search engine using a type of web crawling is a Focused Crawling. The availability of information in large quantities on the Web makes it difficult for user selects resources about their information needs. Search engine works on data collection from the web by software program is called crawler, bot or spider. In this chapter we will cover the types of Web Crawlers like the general Web Crawler and Focused Crawling, the first one fetch the big URLs from the Web, a second one fetch or collects relevant or specific Webpages of interested topics from the Web. It works to reduce the retrieval of web pages, analysis of the interesting topics, so it obtains large number of high quality web pages. In this chapter we will explain or see what the previous topics about general Web Crawler and Focused Web Crawler [33, 64].

4.2 What are the General Web Crawler and Focused Web Crawler?

A general Web Crawler works to fetch a large number of URLs from the Web but without any goal.

Focused Web Crawler a program or tool used to crawl on the web to collect web pages related to a particular topic otherwise rejected the web pages not related. As we know the World Wide Web contains a huge amount of web pages with multiple fields like sport, art, policy and education ...etc.

As we know the World Wide Web contains a huge amount of web pages with multiple fields like sport, art, policy and education ...etc. And each field contains branches as like sport is containing football, handball and basketball. She/he wants fetch only threads about football. Then the search engine focus search and send

Focused Crawling to fetch only the topics related to football. Figure 12 below shows focused crawler works.

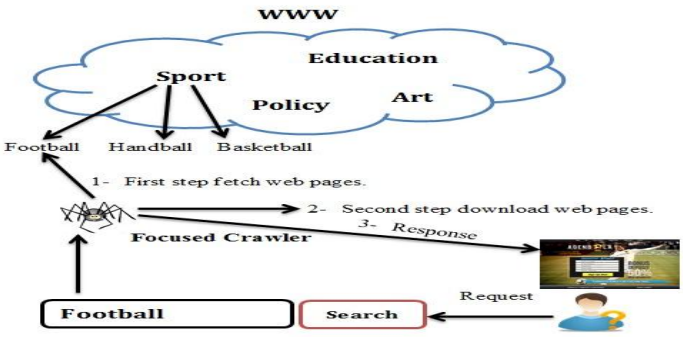


Figure 12 Focused crawling

4.3 First Focused Web Crawling

Chakrabarti, introduced the first focused web crawler. It works to download or fetch related topics only, and avoid catch or download all threads [34].

4.4 Strategies of Focused Crawling

Naming "Focused Web Crawling" was first introduced by "Chakrabarti". It is a program working inside heart of search engine. Their advantages focus on to fetch the web pages that are relevant to a particular topic, according to the user's request. Focused Web Crawler works to avoid fetch the web pages not related to a particular topic. After the end of the fetch process, download the web pages related in a place to be stored and analysis them to be ready for use by the user [34]. Figure 13 below shows different between the Focused Crawling and Regular Crawling.

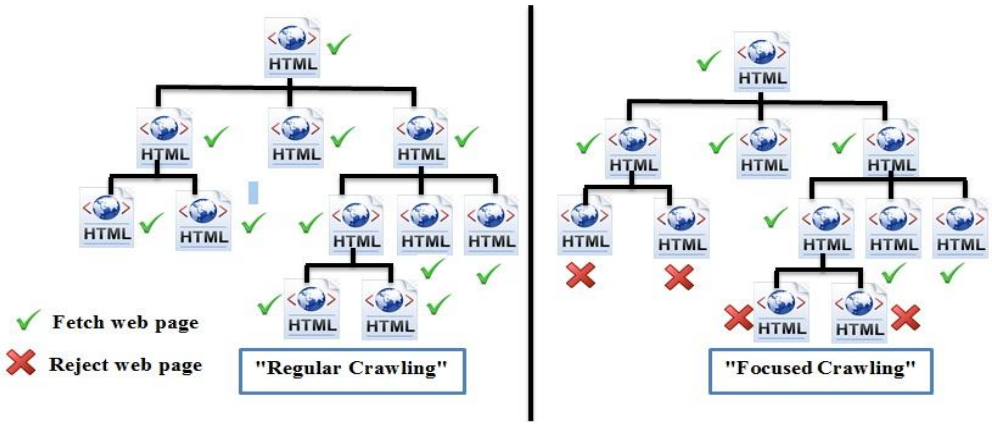


Figure 13 Fetch and Reject Web pages according type of web crawler

4.5 Algorithms used in Focused Web Crawlers (FWC)

Dependent the focused crawlers on two parts of algorithms to crawl on the web and keep the crawling scope within a specific field. The first algorithms are Web analysis, used on an analysis of web pages and quality of the Web pages pointed to by target URLs. The second algorithms are Web search, used to determine and order which the target URLs are visited [35].

4.5.1 Web Analysis Algorithms

These kinds of algorithms can be classification into two parts:

1- "Content based Web analysis algorithms".

2- "Link-based Web analysis algorithms".

The first "content-based analysis algorithms", are working on analysis content the web page to know it among the related topics or not. Most often the URLs content good information on the web page.

For example, http://tubitak.gov.tr/en/announcements/graduate-scholarship-programme-for-international-students/ shows us that URL come from http://tubitak.gov.tr/ and information on scholarship for graduate students. The second algorithms are "Link-based Web analysis", include PR "PageRank algorithm" is part of Google search engine [36] and HITS "Hyperlink-Induced Topic Search algorithm" Is now part of the ask search engine and it works inside the (www.Ask.com) [37]. We can know from the URL structure important details about server, web page language and information. For example the URL below show us the URL come from server in Turkey ".tr" and the web page language is English "/en" .Figure 14 below shows URLs structure.

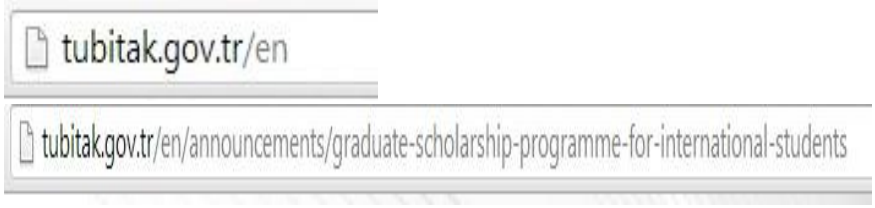


Figure 14 URLs structure

4.5.2 Web Search Algorithms

These algorithms "Breadth-first Search and Best-first Search" are used in focused crawling to determine system credited to visit URLs. These algorithms are used to search trees and graphs; there are two types of ordered trees or graphs and unordered trees or graphs. There are two algorithms used in focused crawling, they are "Breadth-first Search and Best-first Search" are used for unordered trees or graphs. But when we have ordered tree or graph we used binary search tree (BST). Breadth-first Search, its work principle is first-in-first-out (FIFO/Queue). When we use this method to tree search we must start from the root then we scan each node in the level, but first we take left side in search, this means that visit all nodes in the each level without jumping to another level. Figure 15 below shows us (FIFO) method [36, 37, 38].

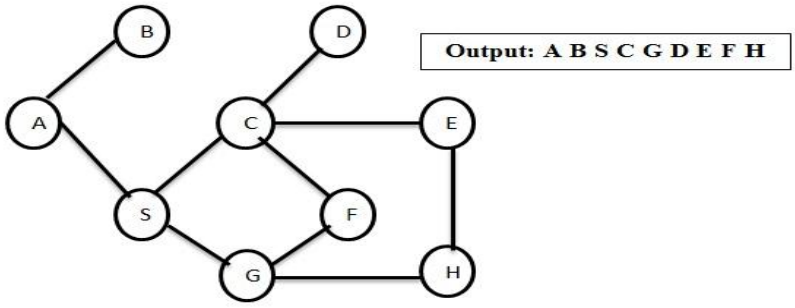


Figure 15 Breadth-First search algorithm

The aim of this algorithm is that it does not focus on which URL to visit next, but trying to collect from URLs in same level, this means that must visit all URLs in same level in the order then jump to next level to visit URLs there. It is considered well to collect web pages for search engines in general. After that other studies have shown to make the Breadth-first search used to build domain-specific collections. Probability here if starting search for relevant URLs in first level then the second level has relevant URLs. Can be fetch web page with domain-specific in breadth-first search with reasonable quality, this appeared in previous studies [40]. Example use "Breadth-First search" to crawl the web. When web crawler visit seed URL, add the URL to queue, then it visit each child in the second level, this means that visiting level after level, and each URLs it visited it works to take them out the queue for no-repetitive [39]. Figure 16 shows the Breadth-First search queue and figure 17 shows Breadth-First search code.

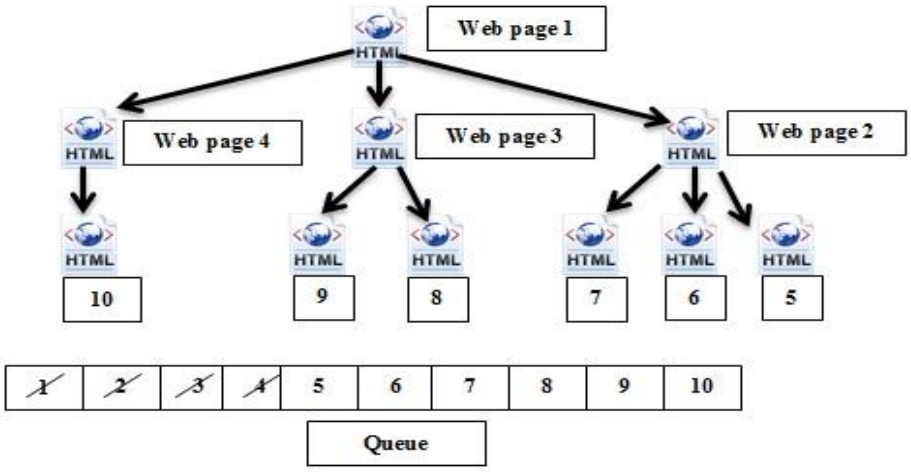


Figure 16 The Breadth-First search queue

```
proc Breadth-First(starting urls)
for each url(starting urls) do
enqueue(frontier, url);
od
while (visited < max pages & frontier not empty) do
url := dequeue(frontier);
page := fetch(url);
visited := visited + 1;
enqueue(frontier, extract links(page));
od
end
```

Figure 17 Breadth-First search code [35]

The crawlers that have been built to fetch small web pages from the web, they cannot fetch large web pages. Breadth-first search algorithm start lose focus to fetch large web pages with noise to collection them. The researchers tried to collect breadth-first and Web analysis algorithms together to build Focused Web Crawling [39]. In their study, breadth-first algorithm search fetch web pages first, then comes the Web analysis algorithm to filtering the related topics or irrelevant Web pages for non-related topics. Summary using breadth-first search alone, this method build and fetch large domain specific collections with less noise. However, Web analysis algorithms fetch irrelevant Web pages for processing during the crawling process; this method has low efficiency [35].

Best-first Search, the principal thought is considering a forefront of URLs, for some assessing standards, the most reliable URL is chosen for the purpose of

creeping (crawling). The forefront can be used as a precedent line. Such equipment is directed by what is extracted from the native database the lexical resemblance of the subject's basic vocabulary and the original sheet of the URL. This rate has been calculated with the help of the Combine crawler. Hence the resemblance of a sheet p and the vocabulary of the subject are exploited to assess the relation that is existed among all the links which belongs to p. Figure 18 shows the Best-First search queue and figure 19 shows Best-First search code [35].

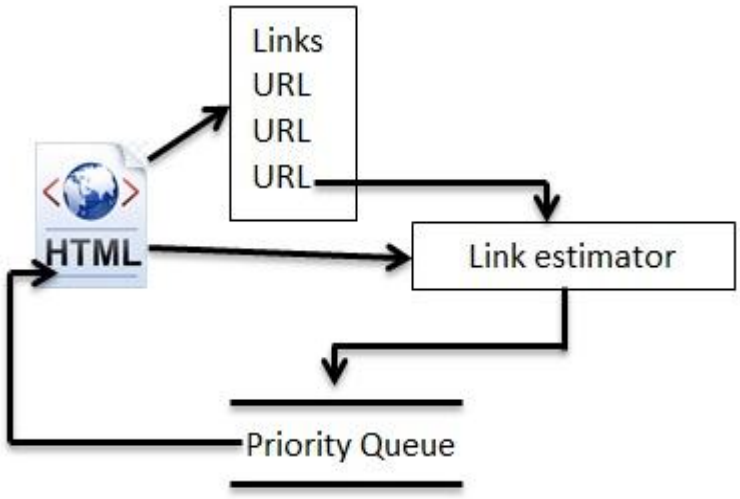


Figure 18 The Best-First search queue

```
proc Best-First(starting urls)
for each url(starting urls) do
enqueue(frontier, url, max score);
od
while (visited < max pages & frontier not empty) do
url := dequeue(frontier);
page := fetch(url);
score := getTopicScore(page);
visited := visited + 1;
enqueue(frontier, extract links(page), score);
od
end
```

Figure 19 Best-First search code [35]

Best-first search, now the algorithm or method most frequently used is focused crawlers. Best-first search method is characterized by breadth-first search, it fetch only relevant web pages and avoids fetching or visiting relevant web pages. Best-

first search has advantages but in same time it has problems. It misses some relevant Web pages during crawl the Web [35].

4.6 More algorithms of Web Crawling

Different search techniques are used in we page search. The goal is to find a sequence of steps that will get us from some root node to some goal node(s) and to cover large number of websites in search engines. The focus here is on computation by exploring the web pages. There are many fields of artificial intelligence, like artificial neural networks, biologically inspired computation, fuzzy intelligence and metaheuristics.

4.6.1 Depth First Search Algorithm:

The big advantage of DFS is that it has much lower memory requirements than BFS, as you only need to store nodes on the current path. The aim of DFS algorithm is to traverses the search by starting at root node and explores as far as possible along each branch through the child nodes. The priority is used in case you have more than one left child, then priority is given to the left most child and continue to go deeper until no more child find then you backtrack to each subsequent parent node and traverse it's children [46]. Stack is used in the implementation of the depth first search to store nodes from the root node to the current node. When a depth-first search succeeds, the path to the objective is on the stack. This algorithm well suited for search problems, but it takes end up in infinite loop when the branches are big [45].

4.6.2 Genetic Algorithm:

Genetic algorithm is adaptive heuristic search algorithm that their techniques inspired by ideas of natural evolution, such as inheritance, mutation and selection. Fitness is what guides the genetic algorithm’s search. Fitness function is used in the algorithm to evaluate the quality of all the proposed solutions to the problem. The idea behind genetic algorithm is to find the best solution from search space solutions in specified time but there is no guarantee to find the optimal solution [47]. Genetic algorithms differ from traditional search methods in genetic algorithms operate on a whole population of solutions while almost all traditional methods search from single point or solution [48,49, 50, 51, 52].

4.6.3 Naive Bayes Classification Algorithm:

Naive bayes classification algorithm represents a statistical method based on applying bayes theorem with independence assumptions [28]. That allows us to capture uncertainty about the model. The efficiency of this algorithm has been proved over many other methods [53]. An efficient way for crawler proposed by Wenxian Wang et al [54]. Peter Flach and Nicolas Lachiche presented Naive bayes classification of structured data [55].

4.6.4 HITS Algorithm:

Hyperlink-Induced Topic Search (HITS) algorithm is a page ranking algorithm used by search engines to perform page rank calculation. This was developed by Jon Kleinberg. It was a precursor to page rank algorithms [56]. In this algorithm, all the web pages are classified into two sets called Hubs and Authorities. This method is not often used [44].

Proposed modification by Joel C. Miller et al on adjacency matrix input to HITS algorithm which give insightful results [57].

4.7 Google Search Engine and PageRank Algorithm

How the Google search engine in the order or sequence of web pages? As in figure 20 below show us order the web pages.

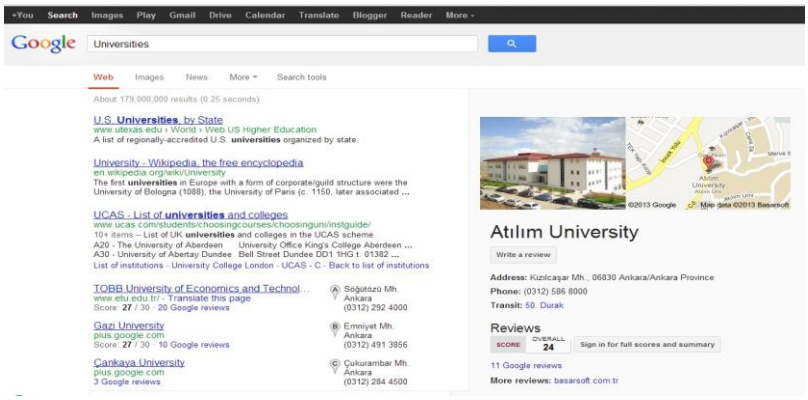


Figure 20 Google search engine with order web pages

4.7.1 Google Search Engine

Is a tool that helps users to access information on the W3. Search engines use keywords or phrase entered by users find or fetch Web sites which contain the information sought [41].

4.7.2 PageRank

Is a “vote”, by all the other web pages on the Web, about how important a web page is? The web page has a number of more votes or links be arranged at the top of the order in list. A link to a web page counts as a vote of support. Increase the rank of the web page on the Web whenever refers to it by a group of web pages [41]. Figure 21 shows add vote or link to web page.

Figure 21 Add vote or link to web page

Published two papers to describe their innovative and patented "PageRank" algorithm by "Larry Page and Sergey Brin" at Stanford University. The PageRank of a web page is essentially its importance (or rank) with respect to the other web pages on the internet. Below formula that calculates the PageRank of each web page [41].

$$\mathbf{PR(A) = PR(T\ 1\ )/C(T\ 1\ ) +\ldots+ PR(T\ n\ )/C(T\ n\ )\ T\ 1\ ,...,T}$$

Below details of this equation:

- d: damping factor, normally this is set to 0.85.
- T1, …, Tn: pages pointing to page A.
- PR(A): PageRank of page A.
- PR(Ti): PageRank of page Ti.
- C(Ti): the number of links going out of page Ti.

4.7.3 Link Structure of the Web

- Every web page has some number of forward links and back links.
- e1 and e2 are Backlinks from webpages A and C of web page C.
- The more backlinks, the more important the web page [41]. Figure 22 below explain the back link.

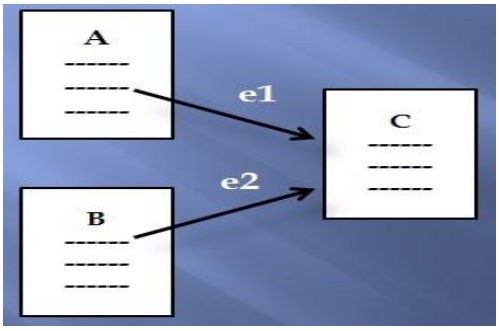


Figure 22 Backlinks from web pages A and B to web page C

4.7.4 PageRank Votes

Is an algorithm used by the Google web search engine to arrange or rank websites in their search engine list results. Increasing rank web page of the more backlink to it. Figure 23 below show you the web page votes and we can deduce from this figure the back link is important of web page [41].

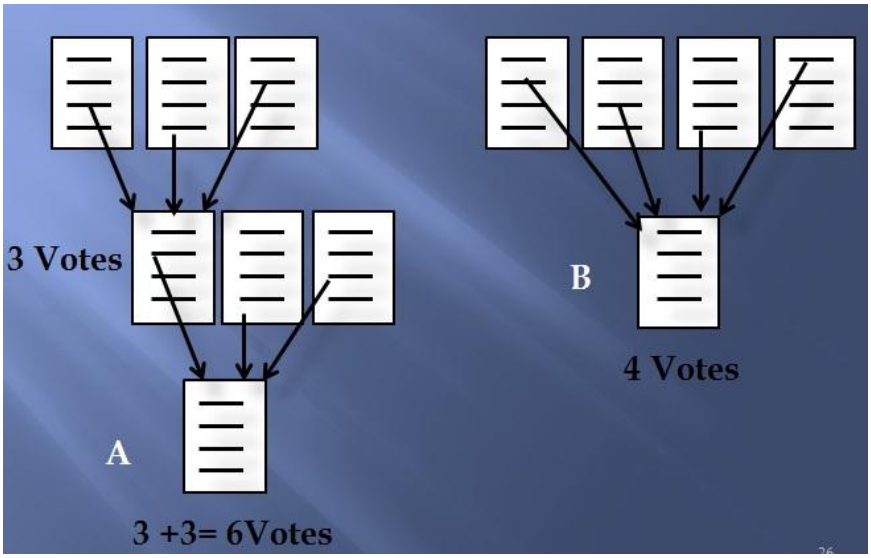


Figure 23 Web page A has 6 votes and web page B has 4 votes

The PageRank display of the Google web search engine toolbar. Toolbar PageRank is given a 1-10 scale. Figure 24 below show you Google real PageRank.

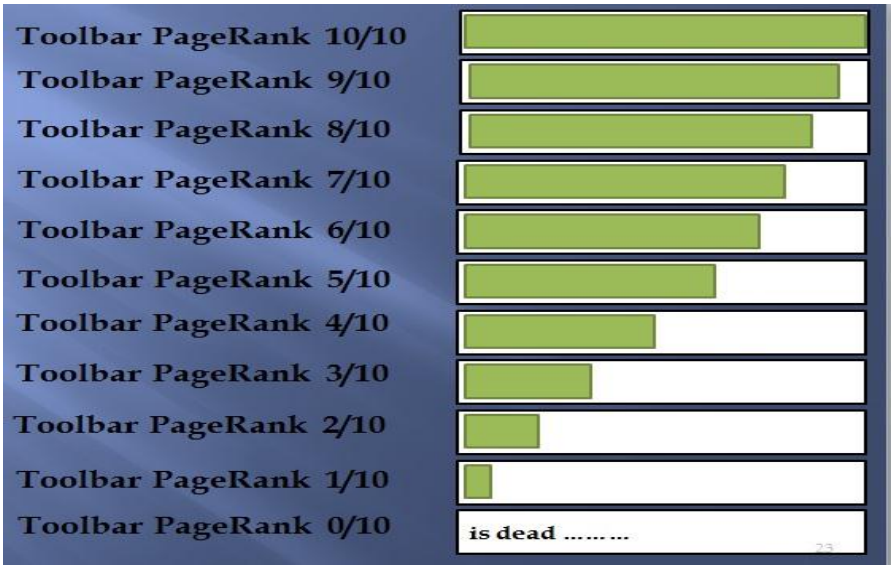


Figure 24 Real PageRank

4.7.5 High PageRank Backlinks

The web page (B) occupies the first rank because it contains a wide range of backlinks from more web pages. For web page (C) occupies the second rank, in the figure 25 we show the web page (C) contain only one backlink but this backlink high rank. These PageRank were calculated via PR(A) = PR(T 1)/C(T 1) +…+ PR(T n)/C(T n) T 1 ,...,T.

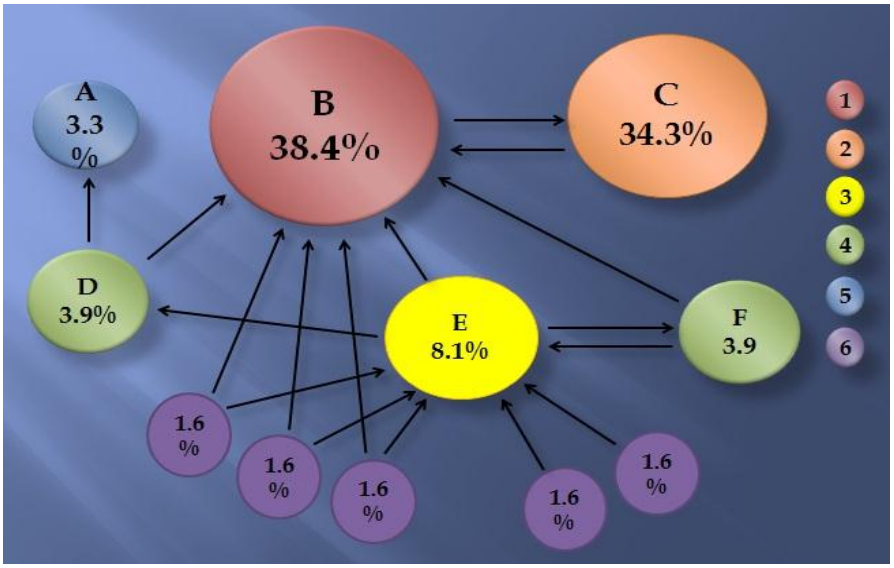


Figure 25 High PageRank backlinks

4.8 Open Source Search Engines and Commercial Search Engines

Search engines are part from Information Retrieval (IR), the rapid growth of information in the Web, led to the increase in commercial search engines like Google, Ask, Yahoo and MSN Search, this led to the increase of Internet users by using this kind of search engines. This led to the generation of reaction in non-commercial search engines designers, to work on the development of non-commercial search engines like Nutch. It open source and it product by Apache Foundation, Nutch written by Java (Programing Language). Nutch contains in its heart a software program is called Lucene, it work like Web crawler, Lucene works to index the documents after the Nutch fetched the URLs from the Web, Lucene works on parser the documents, it works to extract the title, metadata, keywords and author. Nutch and Lucene are free applications available in the internet, and can any internet users or information retrieval technology users, used them. In same time we can their development. Nutch and Lucene requirements are simple to install them in our machine, like Windows 7, Cygwin Linux, and Java environment. Nutch has more benefits are Flexibility, Scalability and Transparency [62]. The availability of information on the web dramatically, leading to difficulty to indexing all these information via the search engines, Nutch is framework for specific works to fetch the Web pages relevant and ignore another Web pages.

4.9 Search Process is Mostly Invisible

Search engines in these days are became important part from Information Retrieval Technology. And because the Search Process is mostly invisible, so more people or Internet users do not know how Search Engines work. In the next chapters, we will explain about using the open-source search engine, to show to the people or Internet users how the search engine and Web crawler work. Figure 26 below shows the Search Process.

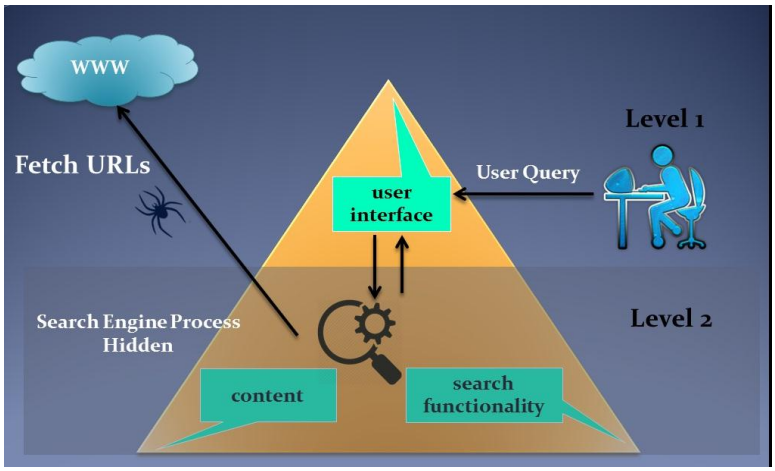


Figure 26 Search Process

4.10 Advantages to Use Focused Web Crawling

Through our study about each of web crawling and focused web crawling, and through the used algorithms to development work the web crawling to make it more focused. We found the focused web crawling best than regular web crawling. Where the focused web crawling focuses only fetch the web pages related to a particular topic. Many methods to training focused web crawling to crawl the web. For example one of focused crawling focuses on the URLs and what its content, another focused web crawling focuses on the content web pages or on the keywords.

In our study we show the work of Nutch Web Crawler, in the framework of a specific, it works to fetch only the related topics and ignore another. The benefits to using this type of Web Crawler, it is free-open source and it index the Web pages via the Lucene. It has more libraries software Information Retrieval.

CHAPTER 5

APACHE NUTCH AND LUCENE

5.1 What is a Nutch

Is a web search engine working to search and index web pages from the Web. Nutch search engine is free/open source based on Lucene, which is an "API" application programming interface for text indexing and searching and they are created by Java. It creates copy of all visited web pages without duplicating. It Developed by Apache Software Foundation. Nutch consists of three stages are fetch, index and search. in figure below shows figure 4.1 Framework of Nutch.

5.2 Nutch Architecture

Functions of Nutch divide into three parts are fetch, index and searcher. The fetch process fetches web pages and turns them into an inverted index. Second stage the index process first, converts the Web pages or other files into the text-document, second divides them into groups called "segments", then process of filtering unwanted stuff, finally configure "inverted index" storing groups from keywords or numbers. Last stage the search process takes the words that are put in the search box by users and information retrieval is based on the basis of the order of these words. It starts from seed of URLs. In the next chapter will talk more about Nutch architecture. In the indexing process the Nutch depends on the Lucene. It written by java, it open source and supported by ASF "Apache Software Foundation". Lucene is information retrieval library for the searching and indexing. It provides a web crawler program, an Index engine and a Query engine [42].

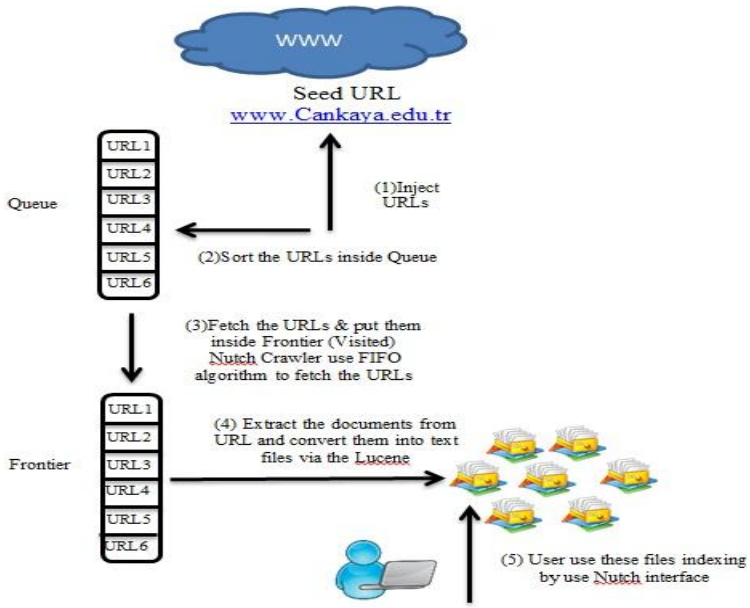


Figure 27 Framework of Nutch

The figure 27 above shows, the Nutch Web Crawler uses the Breadth-First search strategy, when it fetch the URLs from the Web, that mean the Crawler visit all thee Web pages in each level and jump to next level. And so for the levels of other.

5.2.1 Nutch Crawling Process

The diagram below illustrates the sequential process to work Nutch. Three important steps are fetch, indexing and search [42]. In chapter V more detail about Nutch crawler.

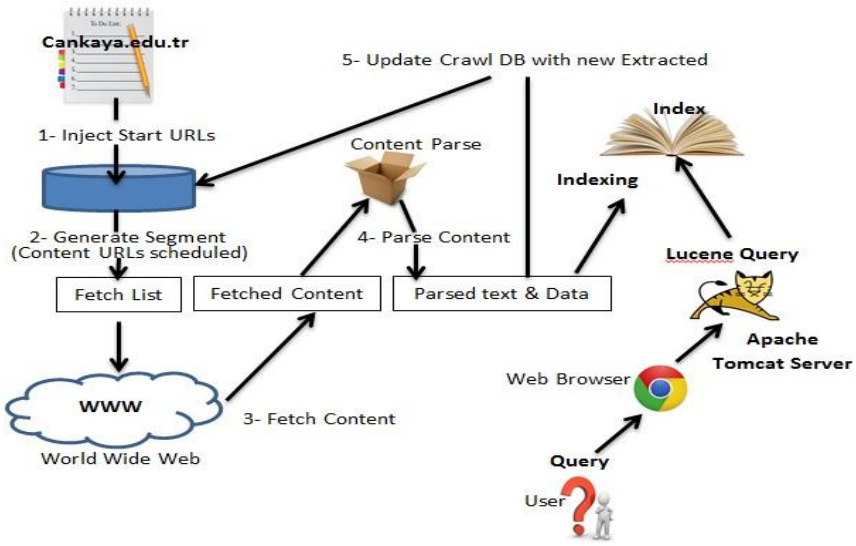


Figure 28 Crawl diagram to understand the crawling process

The Nutch starts the crawling from seeds URLs as shown figure 28 above and then fetch which web page it visited to store it on local disk. Then creating indexing come, it is a program works to convert the web pages into text, then filter some useless information then configure index content key words or inverted index. Finally, search process is a program displays the results to the user according to his/her query [42].

5.2.2 Why Use Nutch

There are a lot of advantages to using use Nutch more than others search engines like Yahoo, Google, Bing and Ask these factors are [43]:

1. Extensibility: - Nutch is flexible. It can be customized and incorporated into your application.
2. Understanding: - We do not have the source for search engines such as Yahoo, Ask, Google and Bing. Nutch is the best we have to see how search engines work. Nutch users can development it and add new algorithms. It being an open source.
3. Transparency: - The important factor to use Nutch is open source. We can see all the algorithms that used to it work. Furthermore so anyone can see how the ranking algorithms work to view the search results.

5.3 What is A Lucene

Is a free/open source in the field of information retrieval and it has more software library, developed and written in Java by Doug Cutting. It is supported by the Apache Software Foundation. Lucene has been ported and can be used with many programming languages like Python, C++, Delphi, Perl, and PHP and C #.

5.3.1 Indexing Use Lucene

Inside the search engines there is the concept of indexing. That means it works on processing the original data, and it works on filtering the "stop word list". This facilitates the search process. Process analyze the "Nutch's documents" by lucene. It free/open source in the field of information retrieval, it written by java and it has more software library, developed by "Apache Software Foundation" [42]. In figure 29 below shows Relationship between Lucene and Applications.

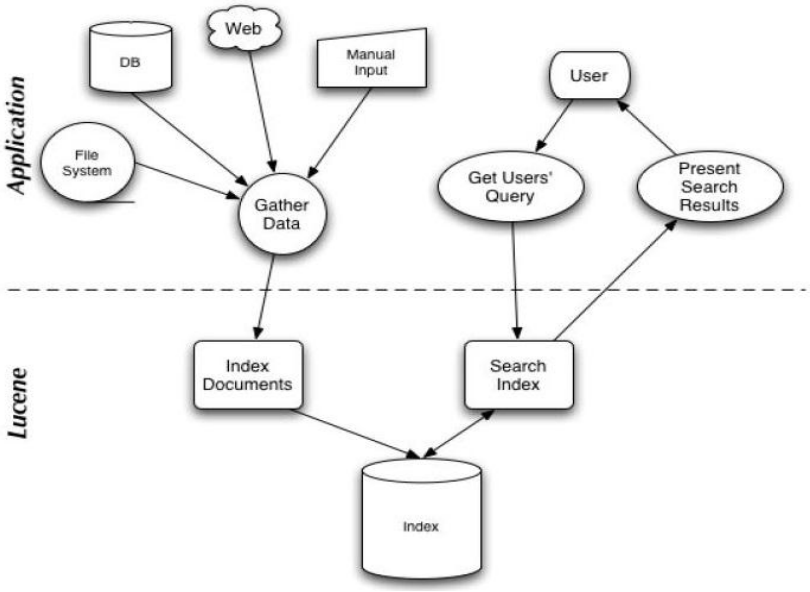


Figure 29 Relationship between lucene and applications

5.3.2 Creating the Index

Three steps to complete the indexing process using lucene are:-

The first step: Document Converting: - The good thing about using lucene doesn't care about data format, types and their languages, as long as we can convert the data to text. This means we can use lucene to search and index any type of data, like PDF, HTML, Microsoft word documents or any other formats we can be analyzed and extract textual information [42], as shown in figure 30 below.

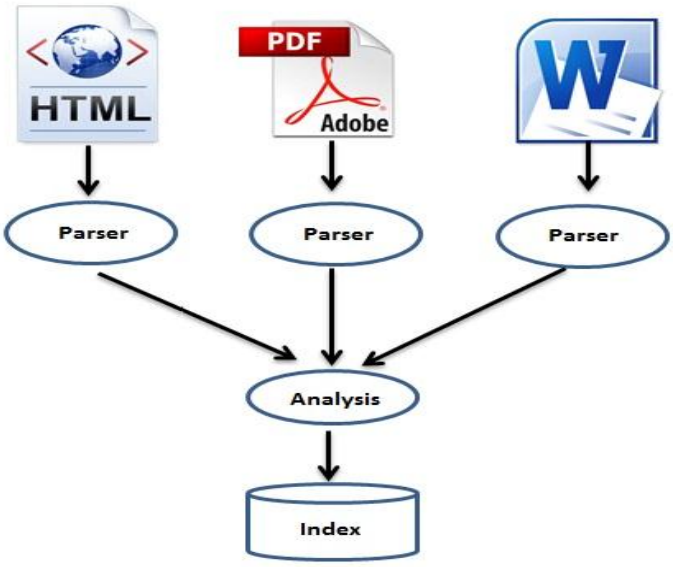


Figure 30 Document converting

The second step: Analysis: - After the completion of the process of converting data, indexing and have created Lucene Documents. Then comes the process of preparing and sort the data indexed, Lucene makes the data more convenient for indexing. To do this, Lucene works on the segmentation of textual data into parts. It works on remove all frequent but have no meaning tokens from the input, such as stop words (a, an, the, in, on, and soon) in English text. There is an important point about documents that contain metadata such as the author, the title, the last modified date, and potentially much more. Metadata refers to "data about data". We must separate these information or metadata and indexed as separate section [42].

The third step: Storing the Index: - Lucene works to sort documents, such as words or numbers, to access to them quickly. Lucene works to break documents indexed into terms, like this example "This is a red car", Lucene separates it into the symbols or words: This, is, a, red, car, and then Lucene works filtering stop words, as shown in figure 31 below.

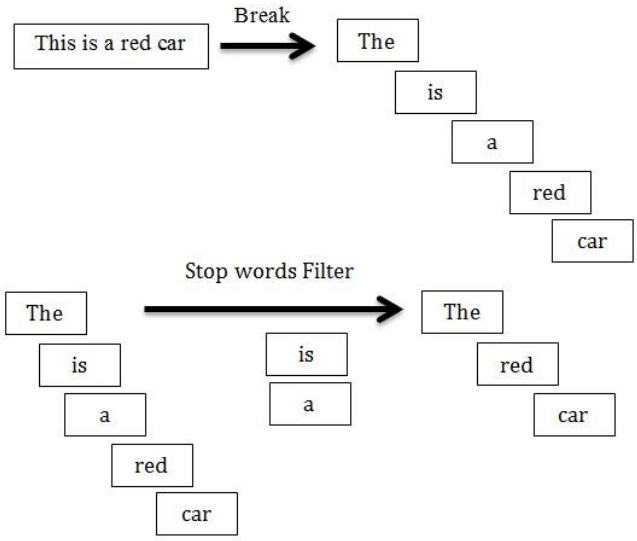


Figure 31 Indexing Process

Furthermore, Lucene uses the (Inverted Index Feature) to sort the documents. Benefits to use inverted index, it is dividing Sentences into Words to speed up and optimize the search process. For example, we have five documents in the Lucene index (Doc1, Doc2, Doc3, Doc 4 and Doc 5) and only (Doc1 and Doc 5) has the University keyword, so Nutch Web Crawler doesn't need to search in the (Doc2, Doc3 and Doc4), when the user search about the University keyword, for this the search process and respond be quick.

CHAPTER 6

IMPLEMENTING A NUTCH WEB CRAWLING

In this chapter we will talk about initialize the "NUTCH and LUCENE" to install and implementation.

Before explain how install the Nutch Web Crawler, we will remind our goals in this study are:

1. Crawl on the Web using Nutch Web Crawler.
2. Indexing via Lucene.
3. Read the Lucene Indexing using the Lucke Lucene Tool.
4. View our results using Tag Cloud Technology.

Figure 32 below shows our system.

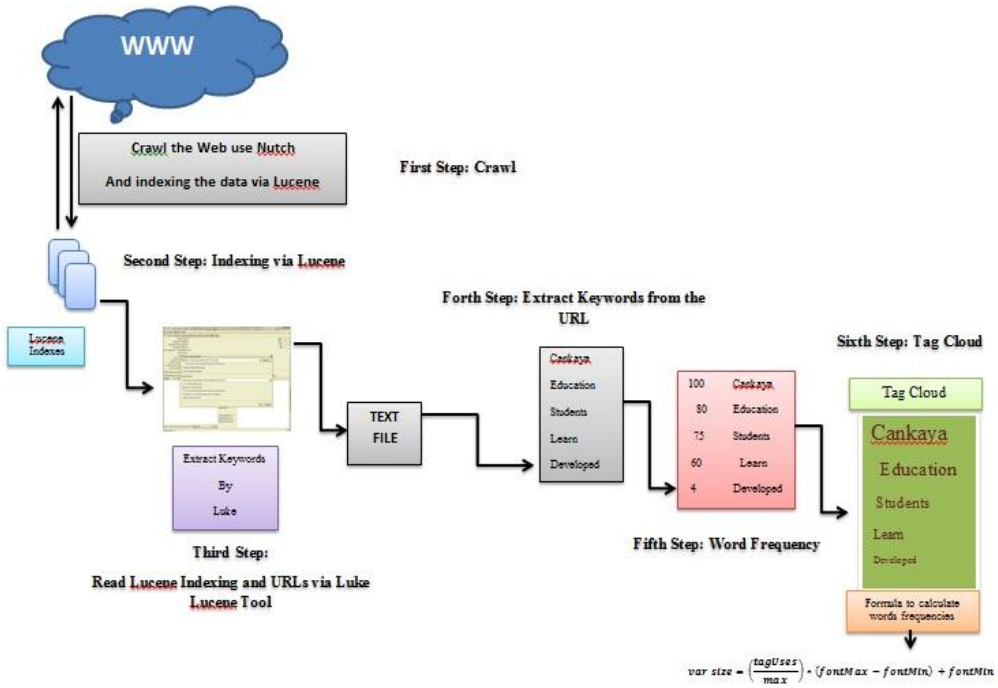


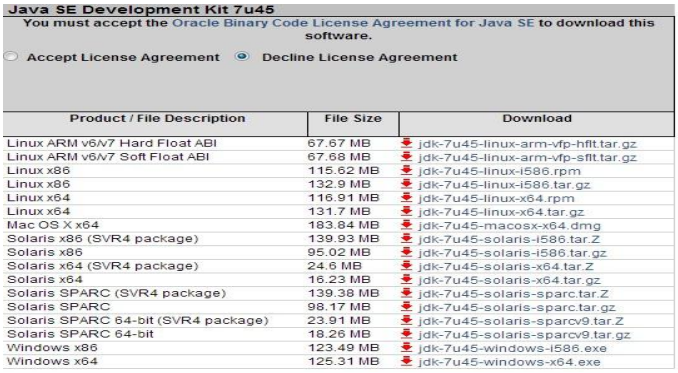
Figure 32 Our system fetch and view result by tag cloud

6.1 Using Nutch

In this section we will explain how to set up the Nutch, we need to download all about it. The operating system that we use in our work is Win 23-bit is a platform to install the Nutch. We downloaded the Nutch from Apache Foundation via the link http://archive.apache.org/dist/nutch/ with all source code, to run it or implemented.

6.2 Installing the Java Environment

Important step that we follow on our project is install the Java Runtime Environment (JRE) on our machine because the Nutch written in Java program language. Finally our machine or PC personal can host the Web Search Engine. This version of the java is good and useful to users of the "Nutch". The Java Development Kit (JDK), also available by Oracle Foundation, it is more useful for software developers. This version of java "jdk1.7.0_17" issued from Oracle foundation was used for the purposes of this study. We can install or run "Nutch" on any version of Java starting with 33 and the newest version of either release is recommended. All versions available on this Web page: "http://www.oracle.com/technetwork/java/javase/downloads/index.html". Figure 5.2 below shows download java from the website.



Java SE Development Kit 7u45

You must accept the Oracle Binary Code License Agreement for Java SE to download this software.

Accept License Agreement Decline License Agreement

Product / File Description	File Size	Download
Linux ARM v6/v7 Hard Float ABI	67.67 MB	jdk-7u45-linux-arm-vfp-hflt.tar.gz
Linux ARM v6/v7 Soft Float ABI	67.68 MB	jdk-7u45-linux-arm-vfp-sflt.tar.gz
Linux x86	115.62 MB	jdk-7u45-linux-i586.rpm
Linux x86	132.9 MB	jdk-7u45-linux-i586.tar.gz
Linux x64	116.91 MB	jdk-7u45-linux-x64.rpm
Linux x64	131.7 MB	jdk-7u45-linux-x64.tar.gz
Mac OS X x64	183.84 MB	jdk-7u45-macosx-x64.dmg
Solaris x86 (SVR4 package)	139.93 MB	jdk-7u45-solaris-i586.tar.Z
Solaris x86	95.02 MB	jdk-7u45-solaris-i586.tar.gz
Solaris x64 (SVR4 package)	24.6 MB	jdk-7u45-solaris-x64.tar.Z
Solaris x64	16.23 MB	jdk-7u45-solaris-x64.tar.gz
Solaris SPARC (SVR4 package)	139.38 MB	jdk-7u45-solaris-sparc.tar.Z
Solaris SPARC	98.17 MB	jdk-7u45-solaris-sparc.tar.gz
Solaris SPARC 64-bit (SVR4 package)	23.91 MB	jdk-7u45-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	18.26 MB	jdk-7u45-solaris-sparcv9.tar.gz
Windows x86	123.49 MB	jdk-7u45-windows-i586.exe
Windows x64	125.31 MB	jdk-7u45-windows-x64.exe

Figure 33 Download java product windows x86

6.3 Selecting a Web Interface

In the previous step has been installed java on our machine which will host the web search engine. The second step is to create a search engine interface that allows users to search through it. The application used in our work, is "Apache Tomcat" ability of filling that role, is one of the applications Apache Software Foundation project, Apache Tomcat is open source. Apache Tomcat is an appropriate environment to run Java code, and it provides a Java HTTP web server. In this study

used Apache Tomcat version 6.0.37 shown in figure 34. Apache Tomcat downloaded from its home page at URL "http://tomcat.apache.org/".

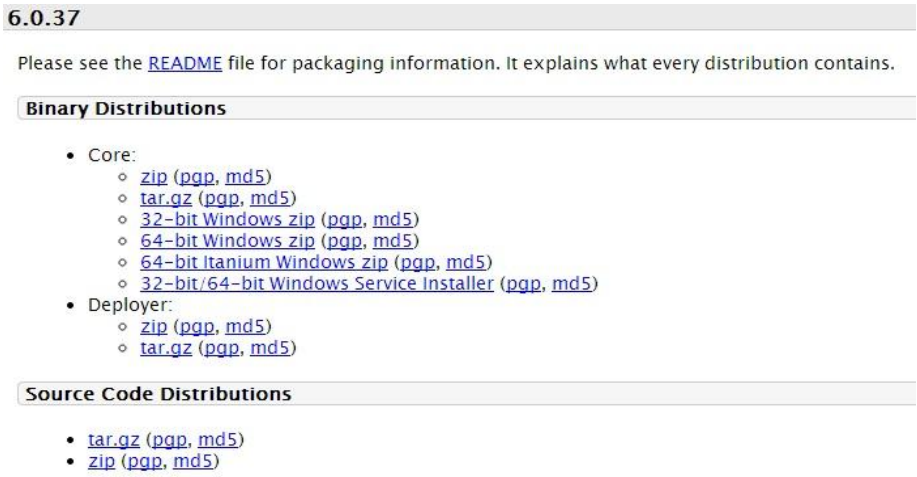


Figure 34 Download "Apache Tomcat" supported the 32-bit windows operating system.

The figure 35 below shows Apache Tomcat through it chooses the search engine interface.

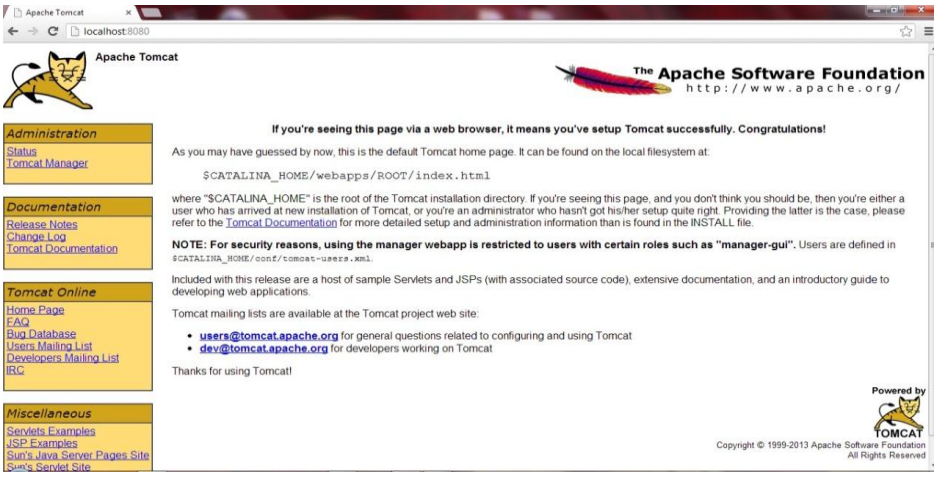


Figure 35 Tomcat interface

6.4 Installing a Shell Environment

The final step to install or setting Nutch a 32-bit Windows environment is the environment which through it sends or writes commands to the web crawler, this shell or environment is "Cygwin". By using Cygwin Can is implemented command-line interface for 32-bit Window. Cygwin is a tools support or provide a Unix-like environment and command-line interface. We used in our thesis this environment. In this thesis used Cygwin new version DLL 1.7.25 shown in figure 36. The new

version of Cygwin shell may be available on http://www.cygwin.com/. Win 2000, win XP, win 2003 Server, win Vista, win 2008 Server, win 7 and 2008 win Server R2 Supported by Red Hat Cygwin (RHG).

Figure 36 Installing and updating cygwin environment or packages

6.5 Web Crawling with Apache Nutch

After completing the necessary requirements, to provide an appropriate environment to "NUTCH and LUCENE", can now use them to crawl in the web. Is an open source with it written in a java programming language, with using it can crawl the Web and find Web pages in an automated manner and reduce lots of maintenance work, the Nutch web crawling can create a copy of all the visited web pages without duplicating and no SQL. The Nutch Owns two parts of the crawl search specified and non-specified in the web. Here we chose seed URL, to Nutch start the crawling from it WWW.Cankaya.edu.tr. The Nutch crawling works to fetch all URLs in the cankaya. The Nutch web search engine meaning works on a specific domain. Example without specific domain is Google search engine. The Apache Software Foundation launched this kind of crawling to bring specific topics with no loss of time to catch non-specific topics. The architecture of apache Nutch is shown in figure 37.

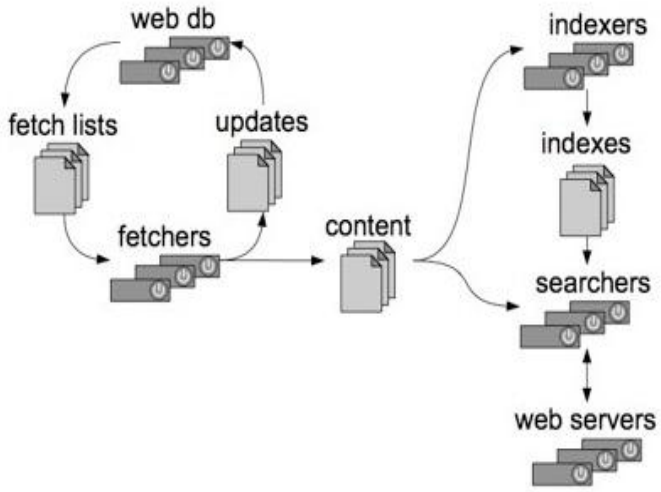


Figure 37 The architecture of nutch crawling

6.5.1 Preparing Nutch for Crawling

Before testing and evaluation of search engine work on the internet, is necessary we do some controls to make a focused crawl on the web and not unfocused large scale web crawler. We used in our thesis a focused crawler on the web. For start crawling on the web must determine target or domain which should be crawled on this domain, this crawler called focused crawl. The Nutch contains set of files, there is a subdirectory called "Conf" inside it contains a search engine's configuration files, under the name "crawl-urlfilter.txt". By the file located in the directory consists of lines that can determine or filter URLs for the web crawler. Each line is a regular expression that preceded by a plus sign "+", which is mean specifies that any URLs that match the expression should be included in the crawl, or a "-". Which precedes an expression who's matching URLs should be reject, these signs represent processes of fetch and rejection the URLs. The line "+^http ://([a-z0-9]*\.)*Cankaya.edu.tr/" was added to the "crawl-urlfilter.txt" for the purposes of testing the "Nutch Web Crawling" and allows the entire "Cankaya.edu.tr" domain to be included in the list crawl. The Structure of the "crawl-urlfilter.txt" file is shown n in figure 38.

```
# See the License for the specific language governing permissions and
# limitations under the License.


# The url filter file used by the crawl command.

# Better for intranet crawling.
# Be sure to change MY.DOMAIN.NAME to your domain name.

# Each non-comment, non-blank line contains a regular expression
# prefixed by '+' or '-'.  The first matching pattern in the file
# determines whether a URL is included or ignored.  If no pattern
# matches, the URL is ignored.

# skip file:, ftp:, & mailto: urls
-^(file|ftp|mailto):

# skip image and other suffixes we can't yet parse
-\.(gif|GIF|jpg|JPG|png|PNG|ico|ICO|css|sit|eps|wmf|zip|ppt|mpg|xls|gz|rpm|tgz|mov|MOV|exe|jpeg|JPEG|bmp|BMP)$

# skip URLs containing certain characters as probable queries, etc.
-[?*!@=]

# skip URLs with slash-delimited segment that repeats 3+ times, to break loops
-.*(/[^/]+)/[^/]+\1/[^/]+\1/

# accept hosts in MY.DOMAIN.NAME
+^http://([a-z0-9]*\.)*apache.org/
+^http://([a-z0-9]*\.)*cankaya.edu.tr/
+^http://([a-z0-9]*\.)*edition.cnn.com/
+^http://([a-z0-9]*\.)*www.en.uobaghdad.edu.iq/

# skip everything else
-.
```

Figure 38 The structure of the "crawl-url filter.txt" file

6.5.2 Configuring Apache Nutch Crawl

Before being crawl in the web, there is file must be edited, this file is "nutch-site.xml". The “nutch-default.xml” this file includes several properties on the identification of Crawler or behavioral crawl. The file "nutch-site.xml" must be added to it "http.agent.name", "http.agent.description", "http.agent.url", and "http.agent.ema -il", these steps are "Internet etiquette". Must give details about this Crawler, for example must write or mention the name of the organization associated with the crawler, description shows the purpose of crawling on the World Wide Web, E-mail should be equipped to communicate the web crawler's handlers and the URL field should point to a URL offering an explanation of the crawler's purpose. After the completion of all of this information then comes a step to create URL list. The configuration "nutch-site.xml" file, shown in figure 39 below.

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>

<!-- Put site-specific property overrides in this file. -->

<configuration>

<property>
  <name>http.agent.name</name>
  <value>Crawl</value>
  <description>
  </description>
</property>

<property>
  <name>http.agent.description</name>
  <value>Ali Nutch Crawl</value>
  <description>Ali Nutch Crawl</description>
</property>

<property>
  <name>http.agent.url</name>
  <value>NUTCH</value>
  <description>http://nutch.apache.org/index.html</description>
</property>

<property>
  <name>http.agent.email</name>
  <value>My E-mail</value>
  <description>alialosh_2004@yahoo.com</description>
</property>

</configuration>
```

Figure 39 Configuration the "nutch-site.xml" file

6.6 Creating a URL List to Fetch

After the completion of the steps to initialize the crawl, now we will create a list of URLs. Put these URLs inside blank text file. Then be crawling on these URLs. The name of this text file is "urls.txt" or "list.txt". This text file is inside the folder under the name "urls". By use the shell command to running the crawl, it crawls on the "urls.txt". In our work we have selected a group of URLs, "http://www.apache.org/", "http://www.cankya.edu.tr " and "http://www.tubitak.gov.tr ", we can use one "url" or more. Figure 40 below shows the list of URLs.

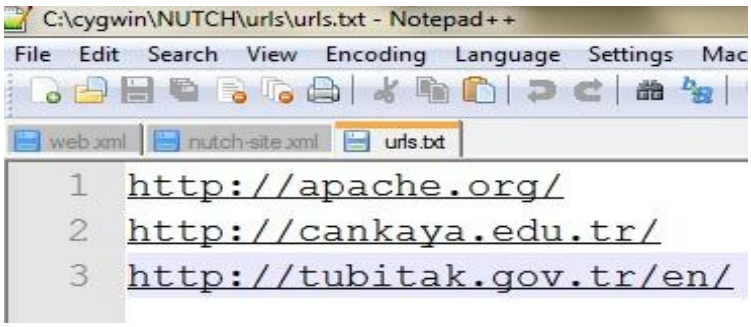
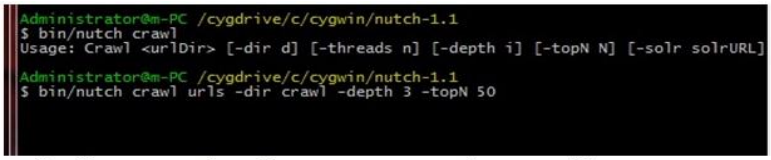


Figure 40 The urls.txt file

6.7 Performing a Nutch Crawl and Using the Crawl Command

After the completed of the main steps to prepare the crawler, then to start crawling on the internet we used in our thesis the "Cygwin" to apply the crawl commands. After we install the "Cygwin" on our machine, now we are ready to run or implementation a crawl, figure 41 below shows the parameters command to crawl:-



•**-dir** *dir* names the directory to put the crawl in.

•**-threads** *threads* determines the number of threads that will fetch in parallel.

•**-depth** *depth* indicates the link depth from the root page that should be crawled.

•**-topN** *N* determines the maximum number of pages that will be retrieved at each level up to the depth.

Figure 41 parameters command bin/nutch crawl

After run the command "bin/nutch crawl urls -dir crawl -depth 3 -topN 50" to crawl the web, figure 42 shows the start crawling command. The results of a crawling must save somewhere on the machine, be crawling on the basis of the location of the URLs.

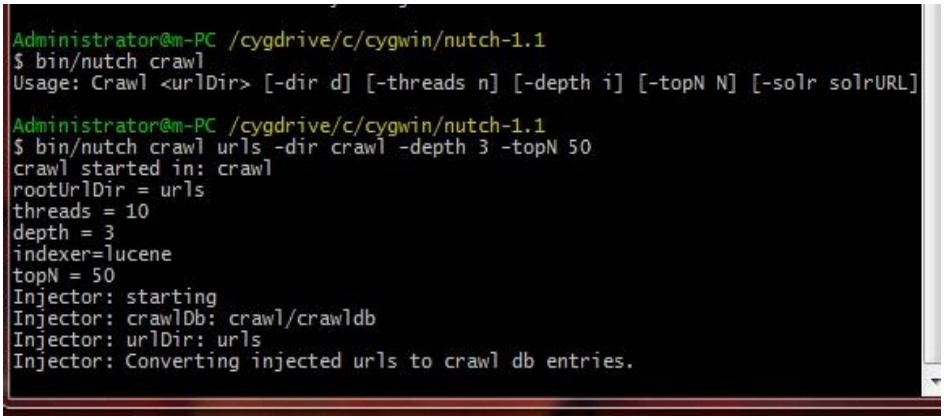


Figure 42 Run crawl command

In our work chose some URLs to search in the web, as shown in figure 43 below. This crawler will choose these URLs to crawl in the web. These URLs stored in the file under name "urls.txt".

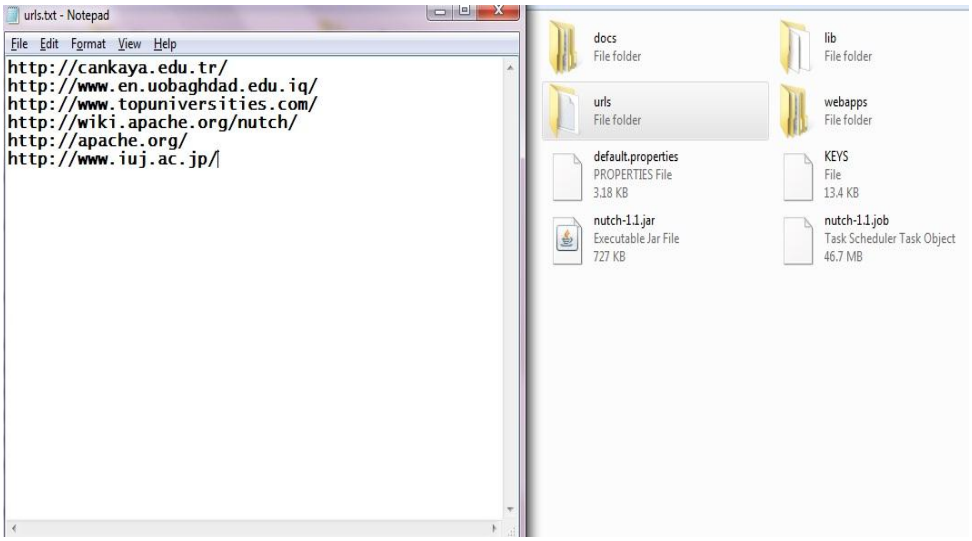


Figure 43 urls.txt

After starting the process of crawling on the web, will generate "crawl folder" shown in the figure 44 below. This folder contains all the information of crawl.

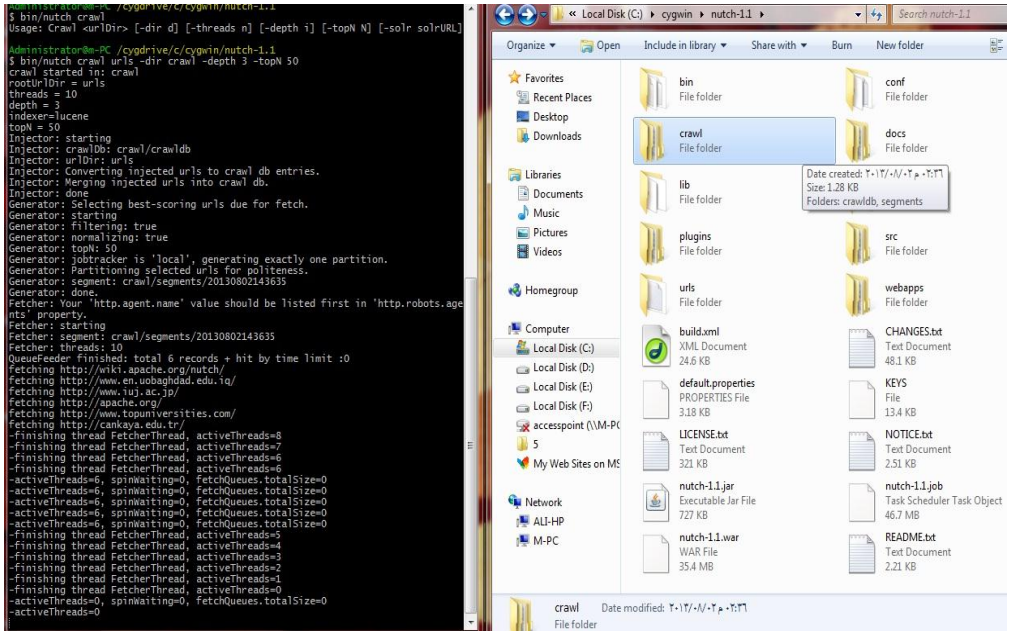


Figure 44 Crawl folder

6.8 Show the Result Crawling (Nutch Data)

After the end of the process of crawling, the "crawl folder" contains the three folders important are "crawldb folder, linkdb folder and segment folder", as shown in figure 45 below.

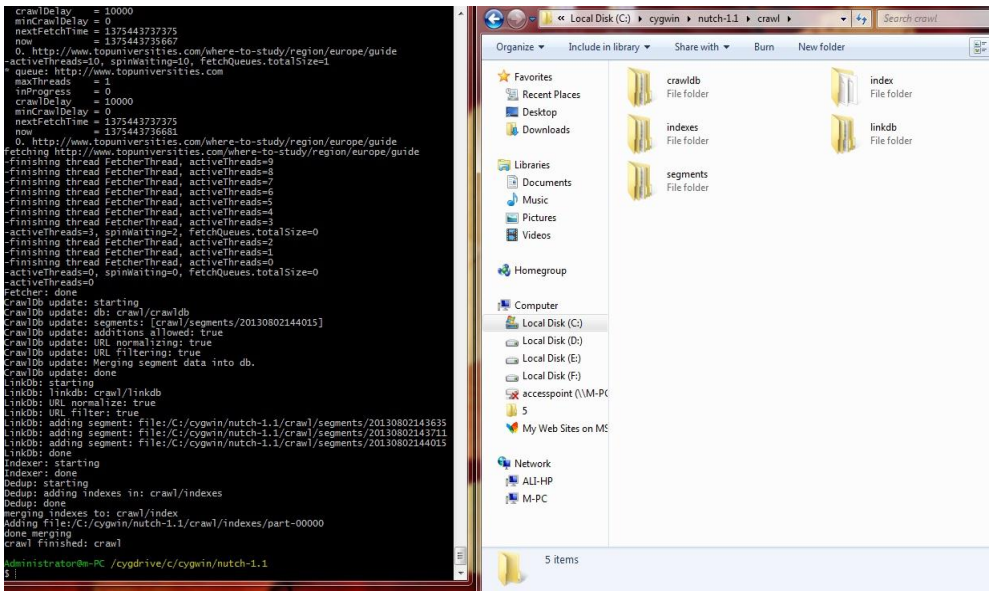


Figure 45 Directories created

The figure 46 below shows "crawl folder" components

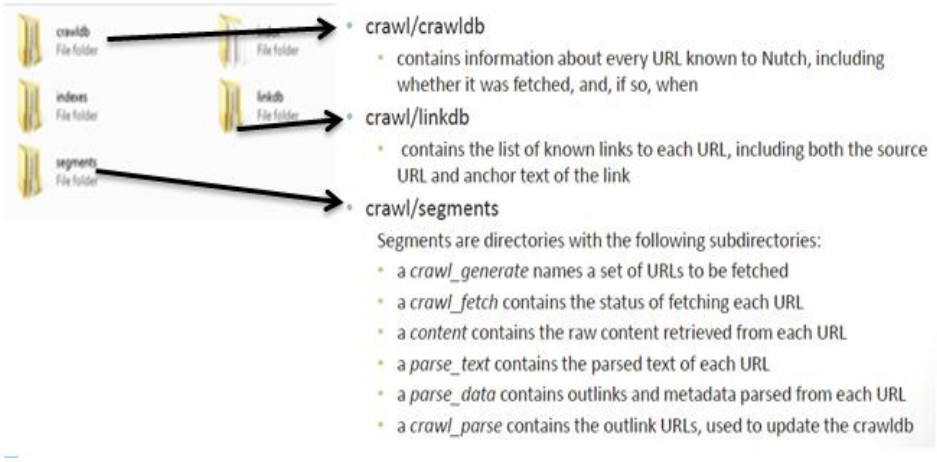


Figure 46 Nutch data

Will we remind the most Nutch commands lines important, after run this command "bin/nutch" we will get commands lines, as shown in the figure 47 below:

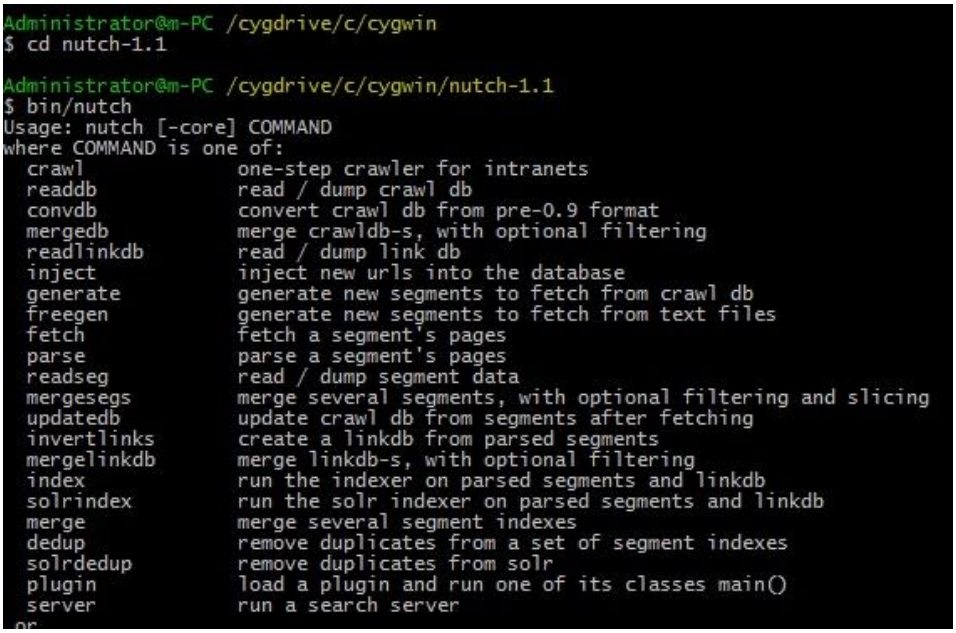


Figure 47 Nutch commands

6.9 Searching the Crawl Results

When finished the crawling process on the web. The second stage is the search in Results crawl. To display these results to the user, we used in our thesis "Apache Tomcat". It is provides this service. Figure 48 below show us directory of crawling results.

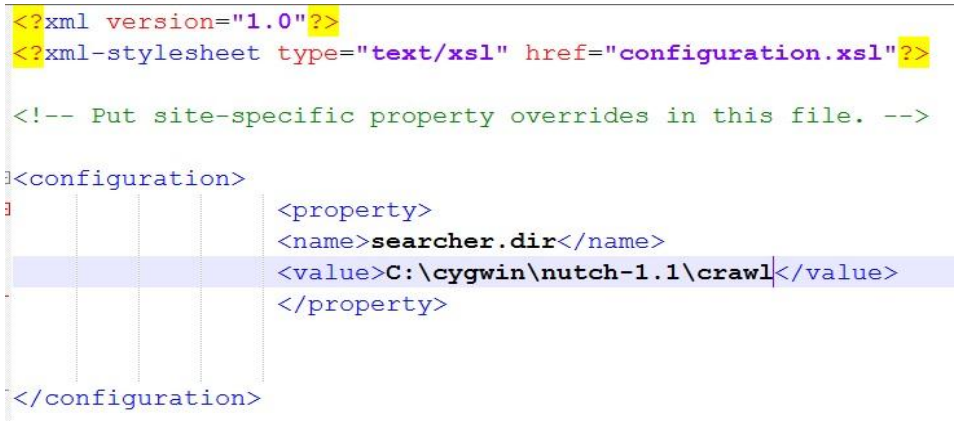


Figure 48 Directory the crawling results

This means that we take this directory (C:\Cygwin\nutch-1.1\crawl) as shown in figure 16 above and we put it in (nutch-site.xml) this file is in (Apache Software Foundation Folder). The directory is "(C:\Program Files\Apache Software Foundation\Tomcat 6.0\webapps\nutch-1.1\WEB-INF\classes)". The "Apache Tomcat Interface" will be displayed by put this links "http://localhost:8080/" in Web browser, as shown figure 49 below.

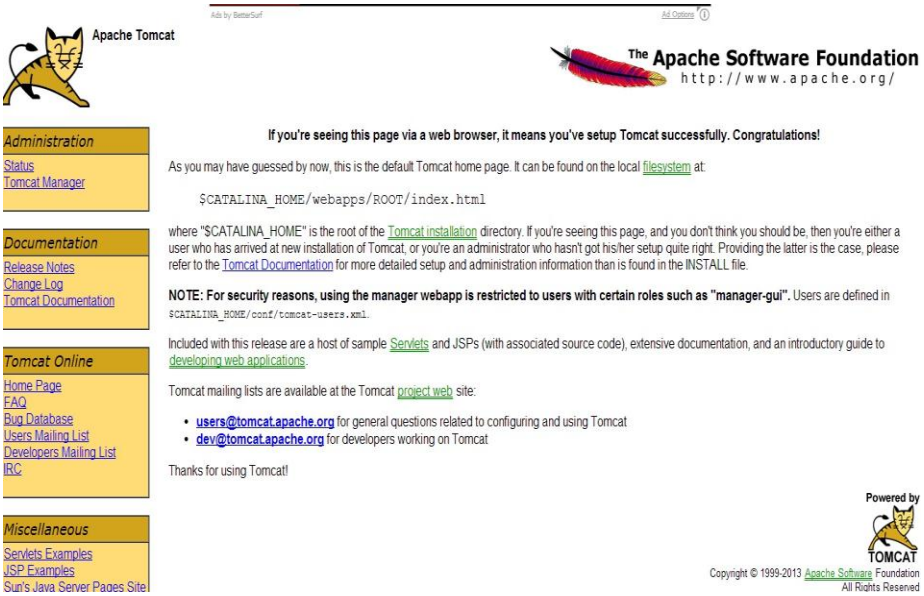


Figure 49 Apache tomcat interface

This interface above contains options in left side, we will choose from these options "Tomcat Manager". By choose this option will Interface appears under the name of "Tomcat Web Application Manager", as shown in figure 50 below.

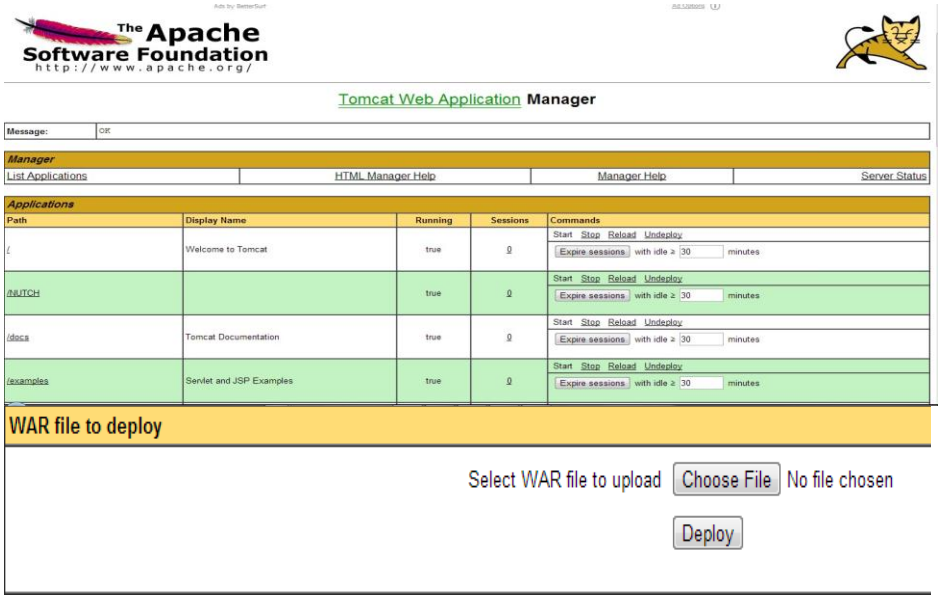


Figure 50 Apache tomcat deploy

To add Nutch to Tomcat Web Application, the choose file button under the "Select WAR file to upload" section should be selected. Then select "nutch-1.1.war" file there will create a new subdirectory Tomcat's "webapps" directory under name "/nutch-1.1 ", as figure 51 below.

Figure 51 nutch-1.1 webapps

After adding "Nutch 1.1 Web" to Tomcat Application correctly, as shown in figure 20 above. Now we can search using "Nutch Web".

6.10 Nutch Web Applications

The search process in the Web comes after the completion of the process of crawling on the web. From Tomcat Application we choose Nutch 1.1 as shown in figure 19 above, will appear us search interface like the figure 52 below.

Figure 52 nutch-1.1

Now user can use Nutch interface through enter of keyword in box search. Then select or press search button to view the results like hits list, as shown in figures 53 and 54 below.

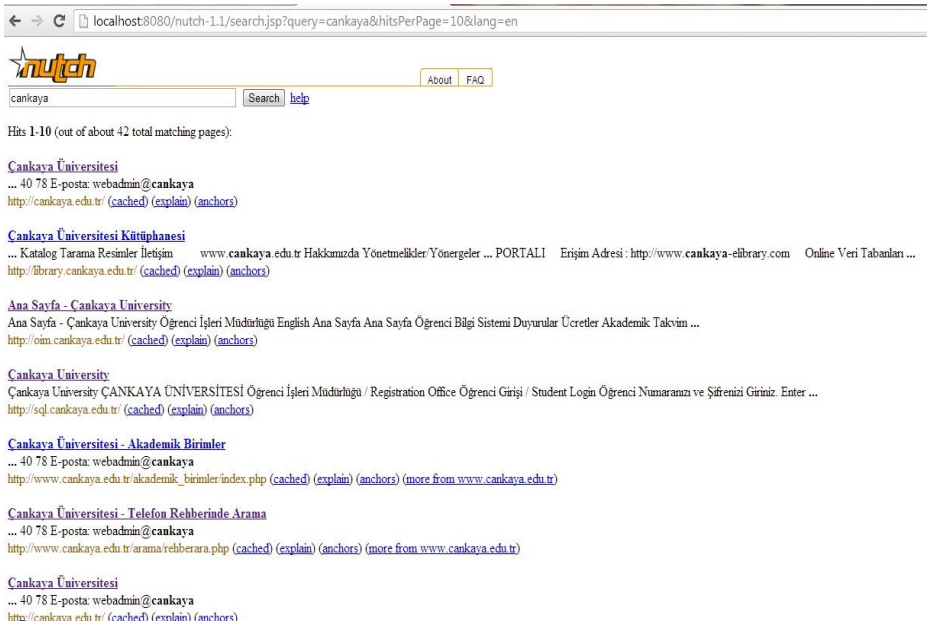


Figure 53 nutch-1.1 web with hits list

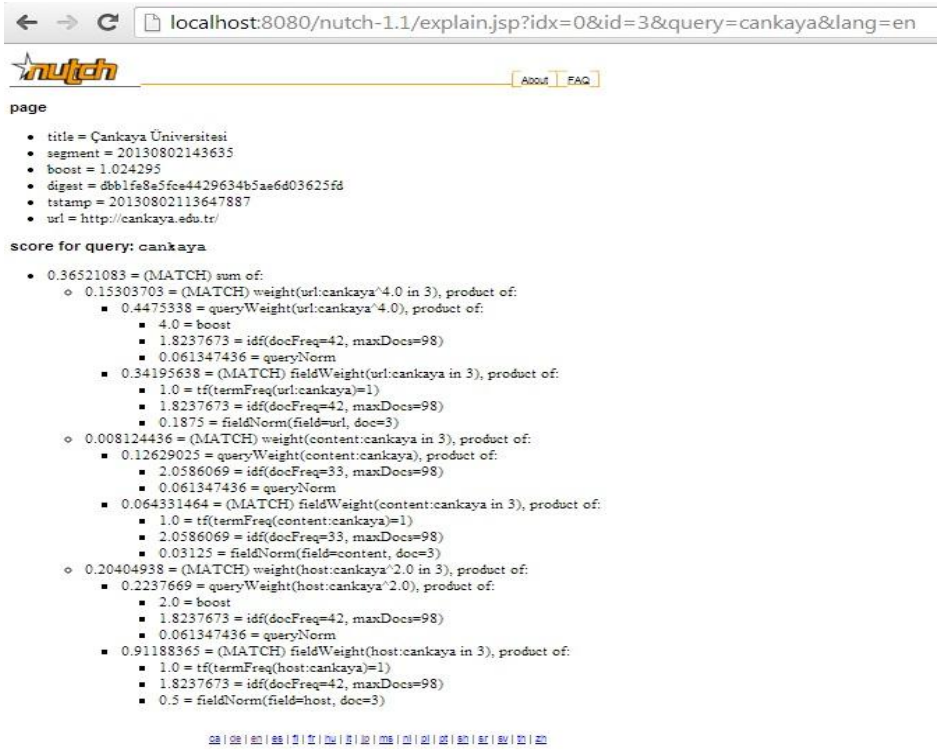


Figure 54 Nutch's score explanation page, matching the query "Cankaya"

Now after we got on the results, we have three important folders, they are crawldb, linkdb and segments, as shown in figure 5.15 above, we want read each folder via use Nutch commands and display their content. These commands as shown in figure 55 below. More details in Appendix C.

```
1- bin/nutch readdb crawl/crawldb -stats
2- bin/nutch readdb crawl/crawldb/ -dump db
3- bin/nutch readlinkdb crawl/linkdb/ -dump link
4- bin/nutch readseg -dump crawl/segments/20131216194731
   crawl/segments/201312161
   9473 _dump-nocontent-nofetch-noparse-noparsedata-noparsetext
```

Figure 55 Command to read nutch folders content

6.11 Use Luke to Analyze the Lucene and Nutch Indexes

To view the contents of Lucene and Nutch indexes, we used the "Luke Lucene Toolbar" to view the results of indexes. "The development and tool is useful to accesses already existing and view the results the Lucene indexes, and allows you to display and modify their content easily", in figure 56 below Showing results by use Luke.

.

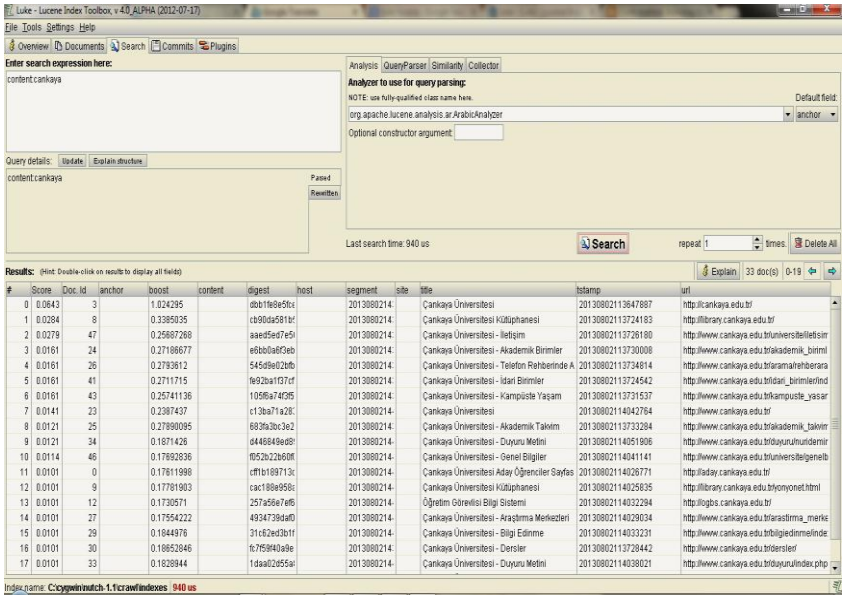


Figure 56 Luke result

6.11.1 Search about Keywords in Lucene and Nutch indexes use Luke

We used "Luke Toolbar" to search about keywords, they are in Nutch indexes. As example we used "Üniversite Hakkında" after search about it we found more URLs Contains it. These URLs are stored in Nutch indexes. In figure 57 below shows "Cankaya Üniversite and Üniversite Hakkında keyword".

Figure 57 Cankaya University web page

After search about this "Üniversite Hakkında" keyword we got result as shown in figure 58 below.

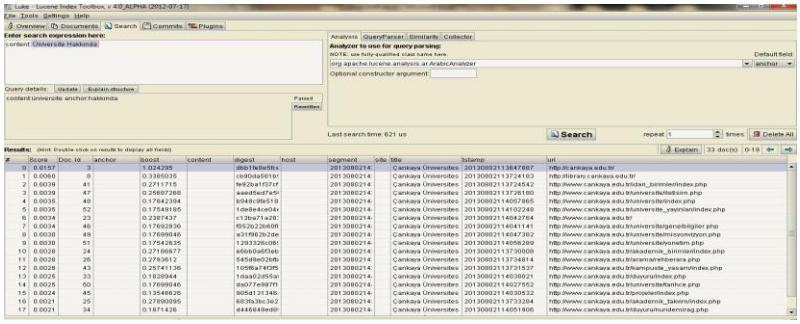


Figure 58 Search result about "Üniversite Hakkında" keyword use Luke Lucene

6.12 Analysis the Lucene Indexing Using Tag Cloud Technology

Tag Cloud is used to identify the topics easily through the expression of words color and different font sizes makes easy to recognize on topic by the most prominent words. We have two important parts to view Tag Cloud, are highest frequency and lower frequency [61]. We can read the Lucene content using the Luke Lucene tools, we mentioned this earlier in the part 5.11, and these tools provide us details about what we fetched from the Web.

We can view our results using Tag Cloud two ways, first way the URL and second way the TEXT.

In the first method, must analysis the URL and take and taking their content (paragraph), we can found the paragraph between Tag this tag is , these details are detailed in paragraph 5.12.1.

In the second method, just we take the text and put it in the Tag Cloud and display the results, these details are detailed in paragraph 5.12.2.

6.12.1 Analysis the URL and Word Frequency

In this part, we took a sample example to explain the word frequency that means how many times each word appears in the Web page. We use python Programming Language to count the frequencies to each word in the Web page [59]. Our work is divided into two parts first part, read the Web page and take all the words appears in the Web page, second part, filter the words into two parts (Meaningful words and stop words). The meaningful word like University and the stop word like the, to, an and etc. The meaningful words are Important [58]. That words which will calculate, then removing all stop words. In this study we took Cankaya University Web page to calculate all words. Figure 59 below shows Cankaya history Web page with words.

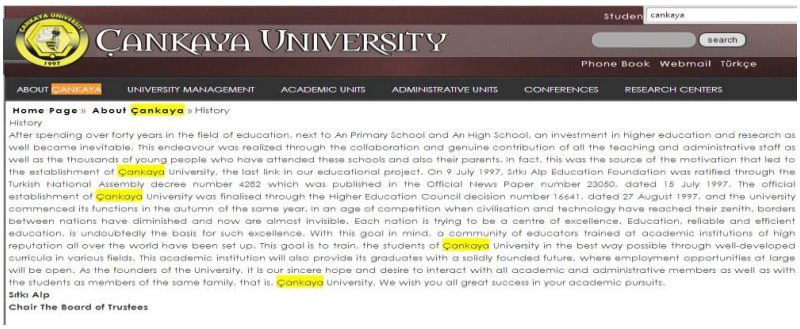


Figure 59 Cankaya history web page

We took sample example to test our code to read some sentences, figure 60 shows list of words**.**

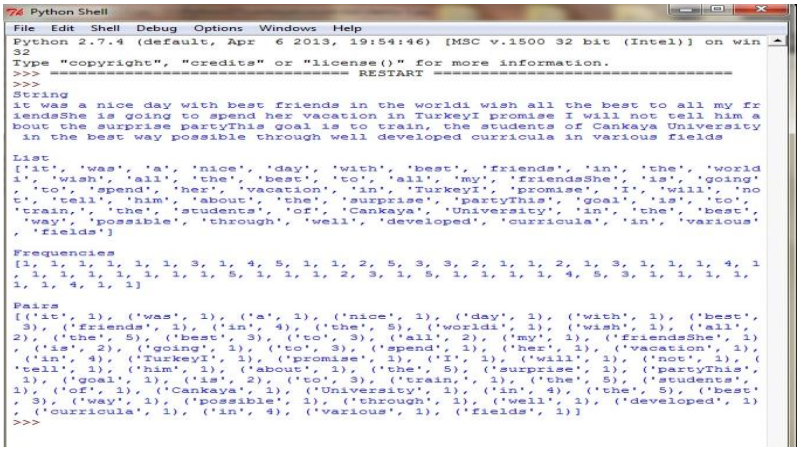


Figure 60 List of words

So, our code is working correctly to read list sentences. We start to read the Cankaya history Web page and index all words in the Web page. The words of Cankaya history exist between this tag   [60], figure 61 shows source code of Cankaya history Web page.

```
<p align=justify>After
      spending over forty years in the field of education, next to Arı Primary
      School and Arı High School, an investment in higher education and research
      as well became inevitable. This endeavour was realized through the collaboration
      and genuine contribution of all the teaching and administrative staff as
      well as the thousands of young people who have attended these schools and
      also their parents. In fact, this was the source of the motivation that
      led to the establishment of Çankaya University, the last link in our educational
      project. On 9 July 1997, Sıtkı Alp Education Foundation was ratified through
      the Turkish National Assembly decree number 4282 which was published in
      the Official News Paper number 23050, dated 15 July 1997. The official establishment
      of Çankaya University was finalised through the Higher Education Council
      decision number 16641, dated 27 August 1997, and the university commenced
      its functions in the autumn of the same year. In an age of competition when
      civilisation and technology have reached their zenith, borders between nations
      have diminished and now are almost invisible. Each nation is trying to be
      a centre of excellence. Education, reliable and efficient education, is
      undoubtedly the basis for such excellence. With this goal in mind, a community
      of educators trained at academic institutions of high reputation all over
      the world have been set up. This goal is to train, the students of Çankaya
      University in the best way possible through well-developed curricula in
      various fields. This academic institution will also provide its graduates
      with a solidly founded future, where employment opportunities at large will
      be open. As the founders of the University, it is our sincere hope and desire
      to interact with all academic and administrative members as well as with
      the students as members of the same family, that is, Çankaya University.
      We wish you all great success in your academic pursuits.
    </p>
```

Figure 61 Python code to indexing words

Now we can call or put the URL (http://www.cankaya.edu.tr/universite/tarihce_en.php) in our code and return all the words with frequency for the Web page, sorted in order of descending frequency with print the list of words. Figure 62 below shows the Words- frequency.

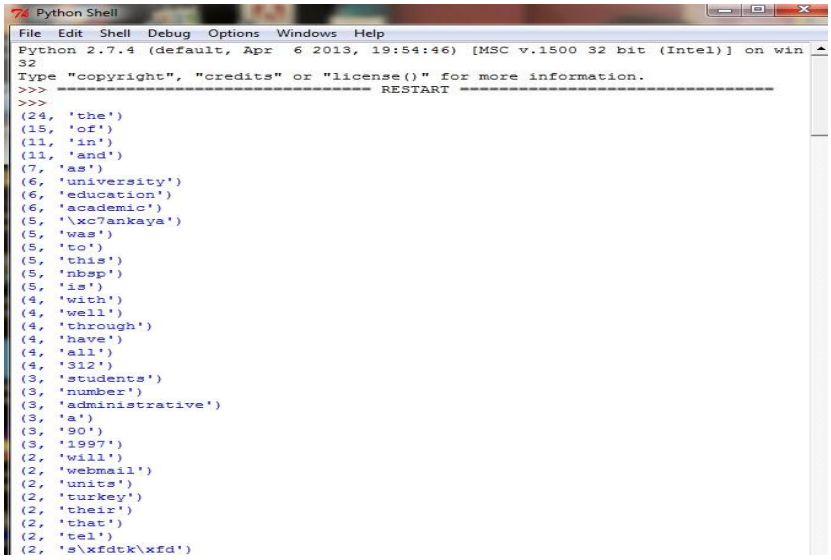


Figure 62 Words- frequency

After run the code above, now we will be removing all stop words and keep only the meaningful words, figure 63 shows the stop words from the list.

```
stopwords = ['a', 'about', 'above', 'across', 'after', 'afterwards']
stopwords += ['again', 'against', 'all', 'almost', 'alone', 'along']
stopwords += ['already', 'also', 'although', 'always', 'am', 'among']
stopwords += ['amongst', 'amoungst', 'amount', 'an', 'and', 'another']
stopwords += ['any', 'anyhow', 'anyone', 'anything', 'anyway', 'anywher
stopwords += ['are', 'around', 'as', 'at', 'back', 'be', 'became']
stopwords += ['because', 'become', 'becomes', 'becoming', 'been']
stopwords += ['before', 'beforehand', 'behind', 'being', 'below']
stopwords += ['beside', 'besides', 'between', 'beyond', 'bill', 'both']
stopwords += ['bottom', 'but', 'by', 'call', 'can', 'cannot', 'cant']
stopwords += ['co', 'computer', 'con', 'could', 'couldnt', 'cry', 'de']
stopwords += ['describe', 'detail', 'did', 'do', 'done', 'down', 'due']
stopwords += ['during', 'each', 'eg', 'eight', 'either', 'eleven', 'els
stopwords += ['elsewhere', 'empty', 'enough', 'etc', 'even', 'ever']
stopwords += ['every', 'everyone', 'everything', 'everywhere', 'except
stopwords += ['few', 'fifteen', 'fifty', 'fill', 'find', 'fire', 'first
stopwords += ['five', 'for', 'former', 'formerly', 'forty', 'found']
stopwords += ['four', 'from', 'front', 'full', 'further', 'get', 'give
stopwords += ['go', 'had', 'has', 'hasnt', 'have', 'he', 'hence', 'her
stopwords += ['here', 'hereafter', 'hereby', 'herein', 'hereupon', 'her
stopwords += ['herself', 'him', 'himself', 'his', 'how', 'however']
stopwords += ['hundred', 'i', 'ie', 'if', 'in', 'inc', 'indeed']
stopwords += ['interest', 'into', 'is', 'it', 'its', 'itself', 'keep']
stopwords += ['last', 'latter', 'latterly', 'least', 'less', 'ltd', 'ma
stopwords += ['many', 'may', 'me', 'meanwhile', 'might', 'mill', 'mine
stopwords += ['more', 'moreover', 'most', 'mostly', 'move', 'much']
stopwords += ['must', 'my', 'myself', 'name', 'namely', 'neither', 'nev
stopwords += ['nevertheless', 'next', 'nine', 'no', 'nobody', 'none']
stopwords += ['noone', 'nor', 'not', 'nothing', 'now', 'nowhere', 'of']
stopwords += ['off', 'often', 'on', 'once', 'one', 'only', 'onto', 'or']
stopwords += ['other', 'others', 'otherwise', 'our', 'ours', 'ourselves
stopwords += ['out', 'over', 'own', 'part', 'per', 'perhaps', 'please']
stopwords += ['put', 'rather', 're', 's', 'same', 'see', 'seem', 'seeme
stopwords += ['seeming', 'seems', 'serious', 'several', 'she', 'should'
stopwords += ['show', 'side', 'since', 'sincere', 'six', 'sixty', 'so']
stopwords += ['some', 'somehow', 'someone', 'something', 'sometime']
stopwords += ['sometimes', 'somewhere', 'still', 'such', 'system', 'tak
stopwords += ['ten', 'than', 'that', 'the', 'their', 'them', 'themselve
stopwords += ['then', 'thence', 'there', 'thereafter', 'thereby']
stopwords += ['therefore', 'therein', 'thereupon', 'these', 'they']
```

Figure 63 stop words list

Figure 64 and table 1 below shows the list of meaningful words after removing all stop words after this step we will draw the chart of meaningful words with each repetition of the word in the Web page.

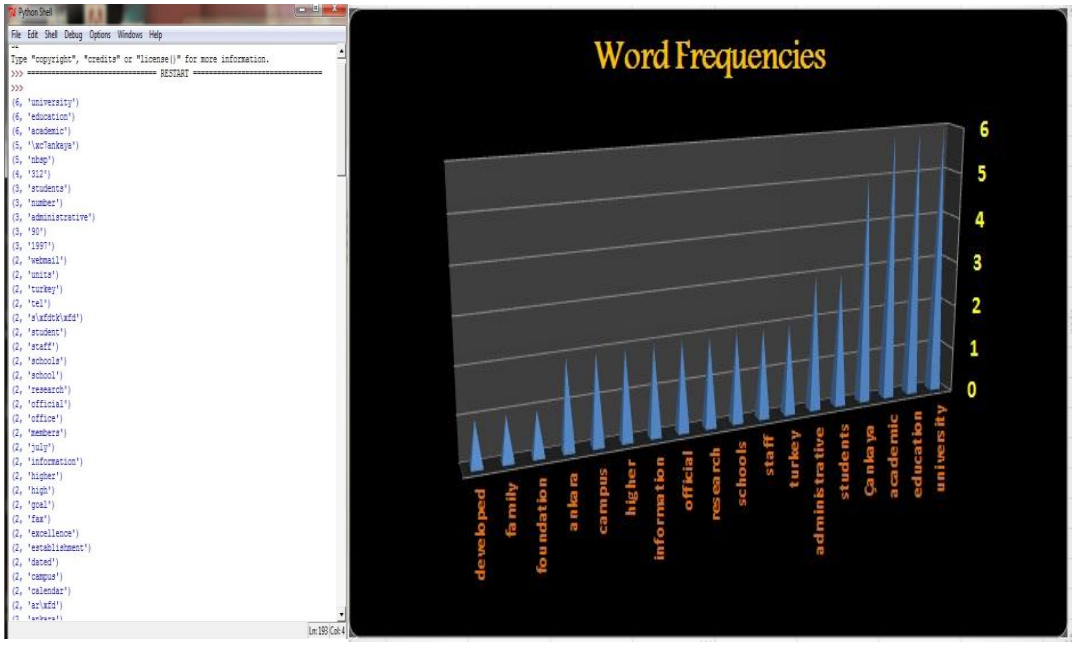


Figure 64 Meaningful words and chart

Table 1 List of word-frequencies

Seq.	Word	Frequencies
1	university	6
2	education	6
3	academic	6
4	Çankaya	5
5	students	3
6	administrative	3
7	turkey	2
8	staff	2
9	schools	2
10	research	2
11	official	2
12	information	2
13	higher	2
14	campus	2
15	Ankara	2
16	foundation	1
17	family	1
18	developed	1

We view the result the table above using the www.wordle.net, it open source to display the keywords in the form of Tag Cloud. Figure 65 below shows the result.

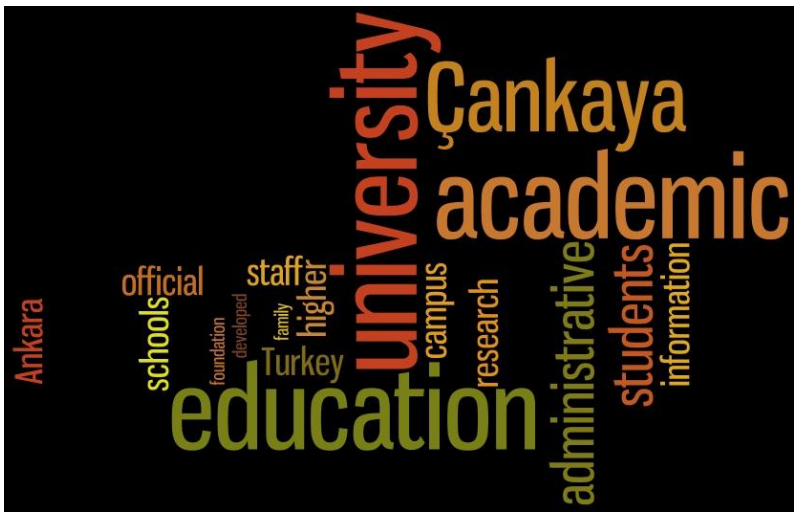


Figure 65 Tag cloud using www.wordle.net

6.12.2 Analysis the Text and Word Frequency

In this part we took the text from the Luke Lucene Tools and analysis it, we put filter to check all words in the text and this filter works to keep all meaningful words like (Education, Cankaya and University) and removing all stops words or common words like (a, an and etc.). In this part we used formula to calculate how many times each word appears in the Web page, where the word that is repeated over (Word Frequency) will show a large font size.

$$\boldsymbol{var\ size} = \left(\frac{\boldsymbol{tagUses}}{\boldsymbol{max}}\right) * (\boldsymbol{fontMax} - \boldsymbol{fontMin}) + \boldsymbol{fontMin}$$

- Var size: The result of font size of the word.
- Tag uses: Word frequency in the Web page.
- Max: The proportion of word frequency expected in Web page.
- Min: Lowest rate.
- Font max: Larger font size.
- Font min: Smaller font size.

The next example will explain the work of this formula.

var max = 100;

var min = 1;

var fontMin = 10;

var fontMax = 20;

We took some words and we apply this formula on them, and then got these results below:

$$var\ size\ (Applcation) = \left(\frac{100}{100}\right) * (20 - 10) + 10 = 20$$

The result of Cankaya Word || 100 || font-size: 20

$$var\ size\ (Resarch) = \left(\frac{75}{100}\right) * (20 - 10) + 10 = 17.5$$

The result of Education Word || 75 || font-size: 17.5

$$var\ size\ (Cankaya) = \left(\frac{30}{100}\right) * (20 - 10) + 10 = 13$$

The result of Academic Word || 75 || font-size: 13

After applying this formula, we'll show our results via Tag Cloud, first we will view the result with meaningful and common words, and figure 66 below shows the result.

Figure 66 Tag cloud before removing common words

Figure 67 below shows the result after removed the common words.

Figure 67 Tag cloud after removing common words

And we have another option is a black list, some time we want remove words like (tto,7u60) and Figure 68 below shows the result after removed the black words.

Figure 68 Tag cloud after removing black words

Beauties to use Tag Cloud:

In use Tag Cloud has many benefits, it can analysis of text responses and display that text in various forms (font types, font size, color and different positions), another benefits Tag Cloud has option to remove the account the word frequency and remove the common words.

CHAPTER 7

CONCLUSION AND FUTURE WORK

7.1 Conclusion

The search engines open source available on the Web we can use them to build own search engine to crawl the Web with reduced cost.

The aim of our thesis was to build search engine open source and display result. In this study we used Apache Nutch is an open source search engine. It is contains and rich many libraries that can be used to information retrieval from the Web and it support multiple languages. The important reason to use Nutch is works over Lucene. It open source written by java. It doesn't care about types of data like PDF, MS Word and HTML. After Nutch fetch web pages, Lucene works to index and analysis these data, then convert them to text. Biggest advantages to use Nutch search engine, it contains high transparency we can be developed it because it open source.

Furthermore, in our new study we made the Nutch works in focused to fetch the webpages. We trained the web crawler to fetch webpages from the web according keywords that we identify. Our web crawling work to fetch related topics, this leads to shorten the time to bring in information from the web and store the all-important information on the hard disk. And our study we used "Luke Lucene Toolbar" to display the results of indexing by Lucene. This tool is useful to accesses already existing and views the results the Lucene indexes, and allows you to display and modify their content easily. It displays all URLs that have been fetching from the web according specific keywords. This means the web crawler read the content the webpage if it finds these keywords, it will fetch this web page and send to Lucene to indexed, if not finds these keywords, it will ignore this webpage and try to find another webpage. This type of web crawlers is considered good for Information

Retrieval, it has its advantages reduce lost time with indexing related topics according the user query.

7.2 Future Work

Work on the development of Nutch and making the Web Nutch Crawler works in focus to fetch the web pages. We will work on building filter to check of each keyword in the web page that we want fetch it from the web. We are building an algorithm operates similar work for the scanner, that means this algorithm work to check all words in the web page. General Nutch Crawler work to fetch the URLs from the Web and send them to Lucene to indexing. In future study after Nutch fetch the URLs and send them to Lucene to indexing, Lucene work to check all words in the Web page according what we want. For example we will using this URL amazon.com and we want just indexes the Web pages include (ipad), so when the Nutch start to fetch all URLs appears in the amazon.com, then Nutch send them to Lucene to indexing, if Lucene found the (ipad) in the webpage, it will indexing that page, otherwise it will ignore it, and check another webpage. Figure 69 shows Nutch focused web crawling.

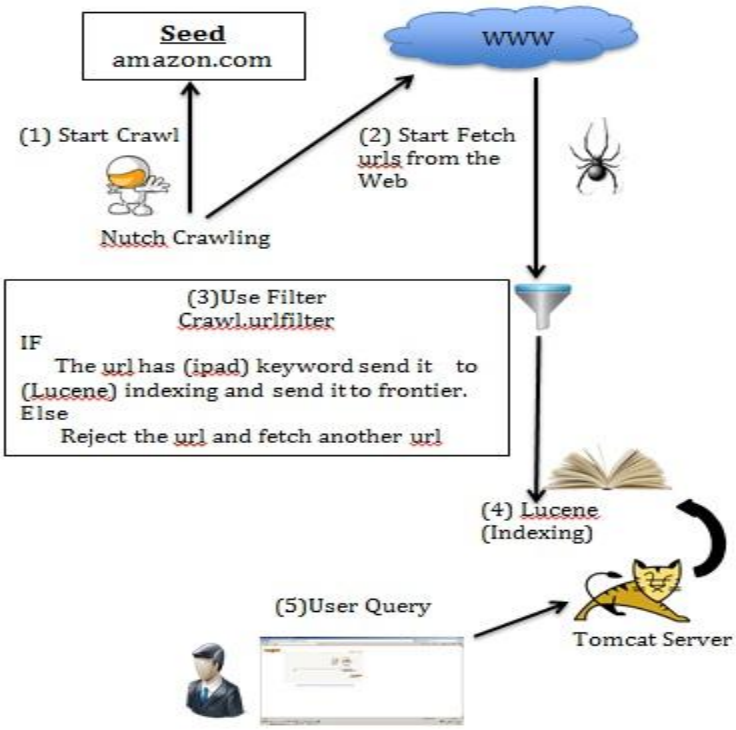


Figure 69 Nutch focused web crawling

In our study we cover the general Nutch Web Crawler, in future work we will make the Nutch Web Crawler, follow the Focused Algorithm. We have two ways to do that, the first way read the URLs, after Nutch injector the URLs and fetch them one by one, befor Lucene index the URLs documents, and we will put condition or filter to check each URL. For example we want fetch and index the URLs that have the (IRAQ keyword) in figure 70 below shows the URL with keyword.

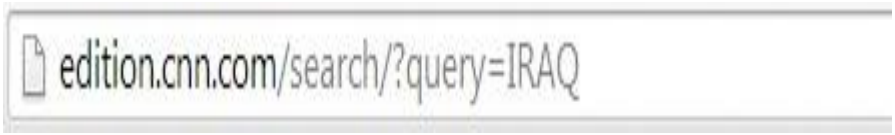


Figure 70 URL with query

So, if lucene found the IRAQ keyword in the URL, Lucene works to parse the Web Page and index, else Lucene ignore or reject that URL and check another URL.

The Second way, we will check the Webpage content, that means the Nutch Web Crawler scan the content of Webpage if found the IRAQ keyword, Lucene works to index the Webpage, Lucene can found the Webpage content between this tag . So, before Lucene knows if the URL has that keyword (IRAQ), must Nutch Web Crawler download all URLs, after that analysis each URL. So, we can fetch the only the related topics.

REFERENCES

1. **Barry M. L., David D. C., Robert E. K., Leonard K., Daniel C. L., Jon P., Larry G. R., Stephen W., (2009)**, *"A Brief History of the Internet "*, ACM SIGCOMM Computer, vol. 39, pp. 22-31.

2. **Bernard J. J., Spink, A., (2006),** *"How are we searching the world wide web? A comparison of nine search engine transaction logs"*. Information Processing and Management 42(1):pp. 248-263.

3. **Handley M., (2006),** *"Why the Internet only just works"*, BT Technology Journal, vol. 24 No 3, pp. 119-129.

4. **Esmail A., (2010)**, *"Internet"*, http://www.slideshare.net/areznamo/internet-4812281. (Data Download Date: 10 March 2013).

5. http://www.google.com/goodtoknow/web/101/. (Data Download Date: 22 March 2013).

6. **Bedi P., Thukral A., Banati H.**, **(2012),** *"A Multi-Threaded Semantic Focused Crawler"*, Journal of Computer Science and Technology, vol 27, Issue 6, pp. 1233-1242.

7. **Rouse M., (2007),** *"Basic Internet"*, (Data Download Date: 10 March 2013) http://searchwindevelopment.techtarget.com/definition/browser.

8. **Jamali M., Sayyadi H., Bagheri H., Abolhassani H., (2006),** *"A Method for Focused Crawling Using Combination of Link Structure and Content Similarity"*, ISBN: 0-7695-2747-7, pp. 753-756.

9. **Umbrich J., Karnstedt M., Harth A., (2007),** *"Fast and Scalable Pattern Mining for Media-Type Focused Crawling"*, Digital Enterprise Research Institute.

10. **Dan C., (2000),** *"A Little History of the World Wide Web"*, http://www.w3.org/History.html. (Data Download Date: 22 March 2013).

11. **Gabriela A., (2013),** *"What Can the World Wide Web Offer ESL/EFL Teachers"*, http://herecomesgab.blogspot.com/2013/05/the-world-wide-web-consists-of-allthe.html. (Data Download Date: 22 March 2013).

12. http://www.google.com/about/. (Data Download Date: 10 March 2013).

13. **Semančík R., (2009),** *"Deficiencies of World Wide Web Architecture",* http://15926.org/references/deficiencies-of-wwwarchitecture.pdf. (Data Download Date: 22 March 2013).

14. **Paul G., (2014),** *"What Is the Difference Between the Internet and Web",*http://netforbeginners.about.com/od/internet101/f/the_difference_between_internet_and_web.htm. (Data Download Date: 10 March 2013).

15. **Aaron W., (2006 - 2013),** *"History of Search Engines: From 1945 to Google Today",* http://www.searchenginehistory.com/. (Data Download Date: 22 March 2013).

16. **KOBAYASHI M., TAKEDA K., (2000),** *"Information Retrieval on the Web"*, ACM Computing Surveys, Vol. 32, No. 2, pp. 145-173.

17. **Leydesdorff L., Hellsten I., Wouters P., (2005),** *"How Search Engines Re-write the Past",* New Media & Society (forthcoming). pp. 1.

18. **Gasser U., (2006),** *"Regulating search engines: Taking Stock and Looking Ahead"*. Yale Journal of Law and Technology. Volume 8. pp. 201.

19. **Grehan M., (2002)**, *"How Search Engines Work"*, New York.

20. **Bing L., (2007, 2011),** *"Web Data Mining"*. Book, 2nd ed. USA. Chapter 6. pp. 211.

21. http://www.webopedia.com/TERM/S/search_engine.html. (Data Download Date: 10 March 2013).

22. **Brin S., Page L., (1998),** *"The Anatomy of a Large-Scale Hypertextual Web Search Engine"*. http://ilpubs.stanford.edu:8090/361/1/1998-8.pdf. (Data Download Date: 23 July 2013).

23. **Seymour T., Frantsvog D., Kumar S. (2011),***"History Of Search Engines",* International Journal of Management & Information Systems. Volume 15, Number 4, pp.47.

24. **Peshave M., (2005),** *"How Search Engines Work and a Web Crawler Application"*, Department of Computer Science University of Illinois at Springfield, IL 62703. pp.89.

25. **O'Connor D., (2003),** *"Search Engines Review Page"*. http://21cif.com/tutorials/micro/mm/searchengines/. (Data Download Date: 10 March 2013).

26. **Patel P.**, *"Search Engine",* Lecturer School of Library and Information Science DAVV, Indore. http://www.clib.dauniv.ac.in/E-Lecture/SEARCH%20ENGINE%20PP.pdf. (Data Download Date: 10 March 2013).

27. **Ratha B.**, *"Search Engine",* Lecturer School of Library and Information Science Devi Ahilya University, Indore, http://www.clib.dauniv.ac.in/E-Lecture/Search%20Engine.pdf. (Data Download Date: 10 March 2013).

28. **Berstein A., (2011),** *"Using Google: Strategies for Searching the Web"*, http://www.vinu.edu/sites/vinu.edu/files/Google_Guide.pdf. (Data Download Date: 10 March 2013).

29. **Kausar A., Dhaka V., Singh S., (2013),** *"Web Crawler: A Review",* International Journal of Computer Applications, vol. 63, no.2, pp. 63.

30. **Brandman O., Cho J., Garcia-molina H., Shivakumar N., (2000),***"Crawler-Friendly Web Servers",* Computer Science Stanford . pp. 1-24.

31. **Olston C., Najork M., (2010),** *"Web Crawling"*, Information Retrieval, vol. 4, no. 3, pp. 175–246

32. **Stamatakis K., Karkaletsis V., PaliouraS G., Horlock J., Grover C., James R., Dingare S.**, **(2003),** *"Domain-SpecificWeb Site Identification: The CROSSMARC Focused Web Crawler"*. Institute of Informatics and Telecommunications., pp. 75.

33. **Babaria R., (2007),** *"Focused Crawling"*, a Project Report. (Data Download Date: 10 July 2013)

34. **Raghavan S., Garcia-Molina H., (2001),** *"Crawling the Hidden Web"*, Proceedings of the 27th VLDB Conference, Roma.

35. **Trujillo R., (2006),** *"Simulation Tool to Study Focused Web Crawling Strategies"*, Department of Information Technology Lund University. pp. 1-50.

36. **Thomas H., Charles E., Ronald L., Clifford S., (2009).** *"Introduction to Algorithms"*, 3rd ed. United States of America. Chapter 6, pp. 594.

37. **Benjamin T., (2010)**, *"Design and Analysis of a Nondeterministic Parallel Breadth-First Search Algorithm"*. Department of Electrical Engineering and Computer Science. pp.3.

38. **Baezayates R., Marin M., Castillo C., Rodriguez A.**, **(2005),** *"Crawling a Country: Better Strategies than Breadth First for Web Page Ordering"*. pp. 864.

39. **Ajwani D., Dementiev R., Meyer U., Osipov V, (2007),** *"Breadth First Search on Massive Graphs "*. pp. 1-15.

40. **Najork M. and Janet L., (2001),** *"Breadth First Search Crawling Yields High Quality Pages"*. Venue Palo Alto, pp. 114-118.

41. **Barfourosh A., Anderson M.**, **(2002),** *"Information Retrieval on the World Wide Web and Active Logic: A Survey and Problem Definition"*. pp. 1-45.

42. **Guojun Y., Xiaoyao X., Zhijie L, (2010).** *"The Design and Realization of Open-Source Search Engine Based on Nutch"*, IEEE. pp.176 – 180.

43. **Cutting D., (2005),** "*Nutch: an Open-Source Platform for Web Search"*, workshop on Open Source Web Information Retrieval.pp. 31.

44. **Signorini A., (2005),** *"A Survey of Ranking Algorithms"* http://alessiosignorini.com/articles/ranking-algorithms-survery/paper.pdf. (Data Download Date: 10 March 2013).

45. **Coppin B., (2004),** *"Artificial Intelligence Illuminated"* Jones and Barlett Publishers. Chapter 4. pp. 75.

46. **Alexander S., (2010),** *"Algorithms and Programming: Problems and Solutions"*, pp. 135.

47. **Sivanandam S., Deepa S., (2008)**, *"Introduction to Genetic Algorithms"*, chapter 1, pp. 2-6.

48. **Shian-Hua L., Jan-Ming H., Yueh-Ming H., (2002),** *"ACRID ,Intelligent Internet Document Organization and Retrieval"*, IEEE, pp. 599 – 614.

49. **José R., (2007),** *"Using Genetic Algorithms for Query Reformulation "*, pp. 16. http://www.bcs.org/upload/pdf/ewic_fd07_paper16.pdf , (Data Download Date: 10 March 2013).

50. **Sanchez E., Miyano H., Brachet J., (1995),** "*Optimization of Fuzzy Queries with Genetic Algorithms"*, Sao- Paulo, Brazil, pp. 293– 296.

51. **Zacharis Z., Themis P., (2001)**, *"Web Search Using a Genetic Algorithm"*, IEEE Internet Computing, ISSN. 1089-780. pp. 18-26.

52. **Varadarajan R., Hristidis V., Li T., (2008),** *"Beyond Single-Page Web Search Results",* IEEE, VOL. 20, NO. 3, pp.1-14.

53. **Caruana R., Niculescu A., (2006),** *"An Empirical Comparison of Supervised Learning Algorithms"* Proc 23rd International Conference on Machine Learning, Pittsburgh, PA.

54. **Wang W., Chen X., Zou Y., Wang H., Dai Z., (2010),** *"A Focused Crawler Based on Naive Bayes Classifier"*, Third International Symposium on Intelligent Information Technology and Security Informatics. IEEE. pp. 517 – 521.

55. **Flach P, Lachiche N.**, **(2004),** "*Naive Bayesian Classification of Structured Data"*, Machine Learning, Kluwer Academic Publishers. pp. 1-36.

56. **John K., (1999)**, *"Hubs, Authorities, and Communities"*, ACM Computing Survey, vol.31, no. 5, pp.1-4.

57. **Miller J., Rae G., Schaefer F., (2001),** *"Modifications of Kleinberg's HITS Algorithm Using Matrix Exponentiation and Web Log Records"*, ISBN: 1-58113-331-6, pp. 444-445.

58. **Brahaj A., (2009),** *"List of English Stop Words"* http://norm.al/2009/04/14/list-of-english-stop-words/. (Data Download Date: 22 March 2013).

59. **Kuhlman D., (2012),** a Python Book: *"Beginning Python, Advanced Python, and Python Exercises"*. 1st Ed, part 1, pp. 10.

60. **Thompson D., (2006)**, "*BASIC HTML".* EBook. 2nd ed. Chapter 1. pp. 1. (Data Download Date 22 March 2013).

61. **Heimerl F., Lohmann S., Lange S., Ertl, T., (2014),** *"Word Cloud Explorer: Text Analytics Based on Word Clouds"*, IEEE Conference Publications, pp. 1833-1842.

62. **Khare R., Cutting D., Sitaker K., Rifkin A., (2005),** *"Nutch: A Flexible and Scalable Open-Source Web Search Engine"*, Chiba, Japan. pp.62.

63. **Mccandles S., Hatcher E., Gospodnetić O., (2010)**, E-book *"Lucene in Action"*. 2nd ed, Part one pp. 6. http://dl.e-book-free.com/2013/07/lucene_in_action_2nd_edition.pdf (data Download Date: 22 March 2013).

64. **Babaria J., Saketha N., Krishnan S., Bhattacharyya M., Murty N., (2007)**, *"Focused Crawling with Scalable Ordinal Regression Solvers"*, In Proceedings of the ICML-2007 Conference.

APPENDICES

APPENDIX A - THEN INDEX THIS WEB PAGE

Home Page » **About Çankaya** » History

History

After spending over forty years in the field of education, next to An Primary School and An High School, an investment in higher education and research as well became inevitable. This endeavour was realized through the collaboration and genuine contribution of all the teaching and administrative staff as well as the thousands of young people who have attended these schools and also their parents. In fact, this was the source of the motivation that led to the establishment of Çankaya University, the last link in our educational project. On 9 July 1997, Sıtkı Alp Education Foundation was ratified through the Turkish National Assembly decree number 4282 which was published in the Official News Paper number 23050, dated 15 July 1997. The official establishment of Çankaya University was finalised through the Higher Education Council decision number 16641, dated 27 August 1997, and the university commenced its functions in the autumn of the same year. In an age of competition when civilisation and technology have reached their zenith, borders between nations have diminished and now are almost invisible. Each nation is trying to be a centre of excellence. Education, reliable and efficient education, is undoubtedly the basis for such excellence. With this goal in mind, a community of educators trained at academic institutions of high reputation all over the world have been set up. This goal is to train, the students of Çankaya University in the best way possible through well-developed curricula in various fields. This academic institution will also provide its graduates with a solidly founded future, where employment opportunities at large will be open. As the founders of the University, it is our sincere hope and desire to interact with all academic and administrative members as well as with the students as members of the same family, that is, Çankaya University. We wish you all great success in your academic pursuits.

Sıtkı Alp

Chair The Board of Trustees

Figure A- 1: Cankaya University Web Page History

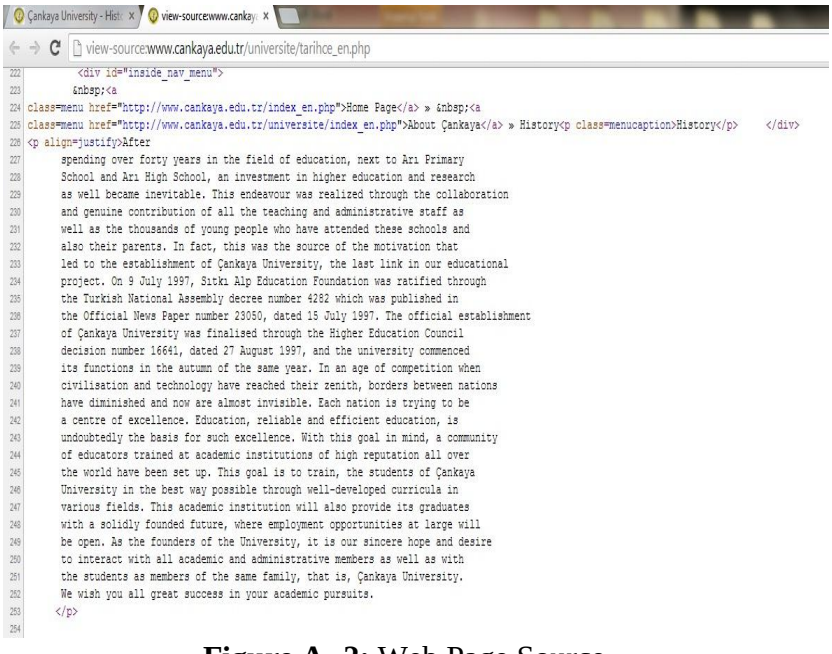


Figure A- 2: Web Page Source

APPENDIX B - TAG CLOUD WITH OPEN SOURCE WEB SITES WORDLE (http://www.wordle.net/)

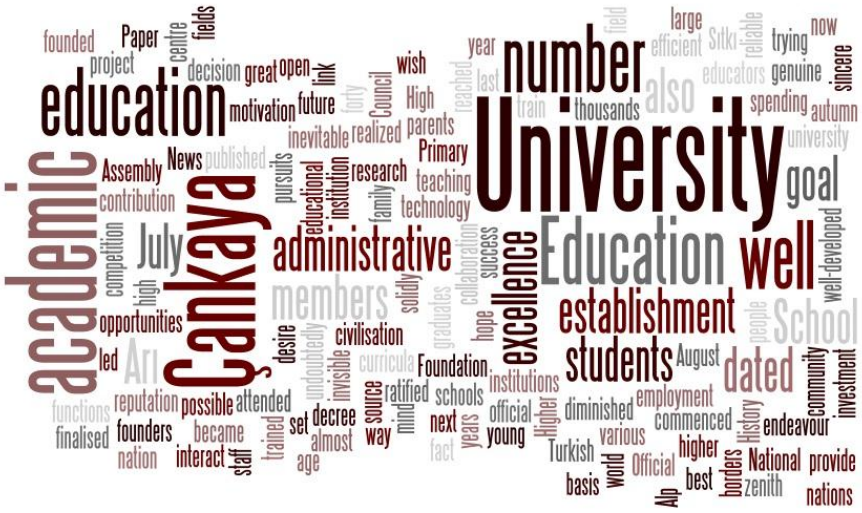


Figure B- 1: Tag CloudEnglish

Figure B- 2: Tag CloudArabic

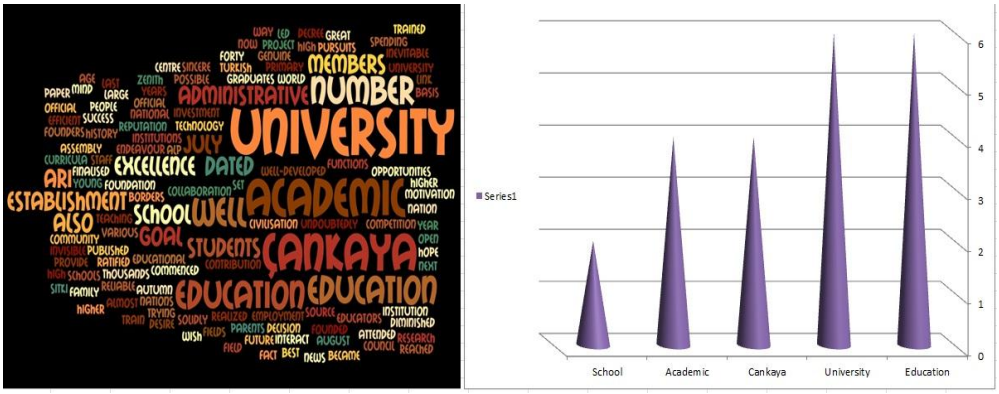


Figure B- 3: Tag Cloud With Chart for the Repeated Keywords (Cankaya University Web Page)

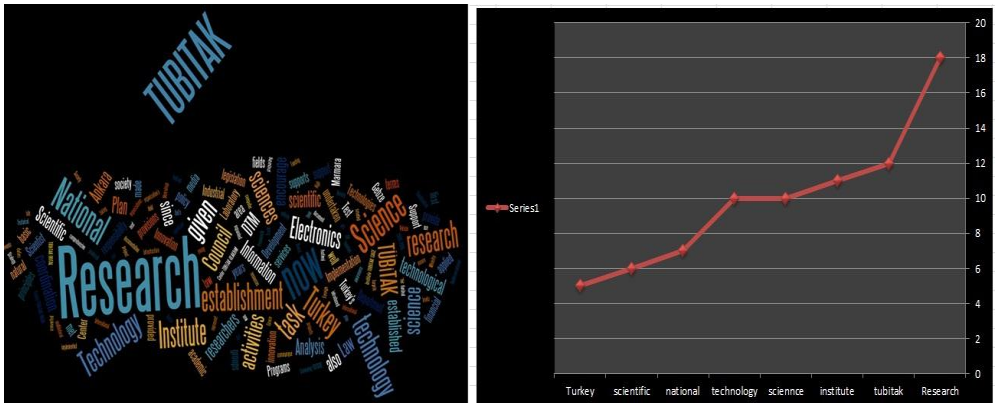


Figure B- 4: Tag Cloud With Chart for the Repeated Keywords (TubitakWeb Page)

B- 5: More examples with (http://www.tagxedo.com/)

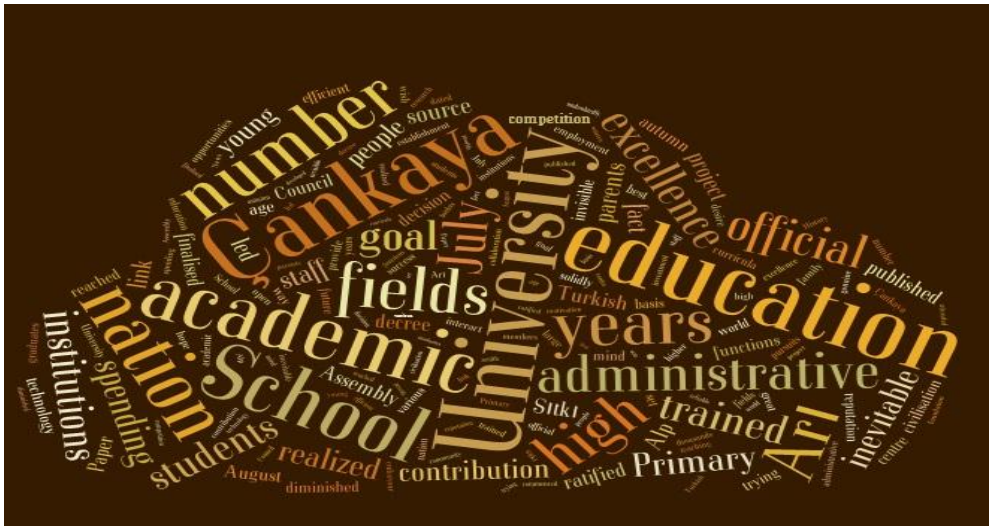


Figure B- 5

B- 6: More example with **(http://www.wordle.net/)**

Figure B- 6

APPENDIX C - DISPLAY NUMBER OF URLS THAT RETRIEVED FROM THE WEB AND THE ALGORITHM OF FETCH

```
Administrator@m-PC /cygdrive/c/cygwin/nutch-0.9
$ bin/nutch readdb crawl/crawldb -stats
CrawlDb statistics start: crawl/crawldb
Statistics for CrawlDb: crawl/crawldb
TOTAL urls:     1773
retry 0:        1773
min score:      0.0
avg score:      0.007
max score:      2.0
status 1 (db_unfetched):        1670
status 2 (db_fetched):  84
status 3 (db_gone):     13
status 4 (db_redir_temp):       2
status 5 (db_redir_perm):       4
CrawlDb statistics: done
```

Figure C- 1: Command to view the content of crawl

```
Administrator@m-PC /cygdrive/c/cygwin/crawl
$ bin/nutch readdb crawl/crawldb/ -dump db
CrawlDb dump: starting
CrawlDb db: crawl/crawldb/
CrawlDb dump: done
```

part-00000

```
1  http://acs105.cankaya.edu.tr/    Version: 7
2  Status: 1 (db_unfetched)
3  Fetch time: Mon Dec 16 19:48:34 EET 2013
4  Modified time: Thu Jan 01 02:00:00 EET 1970
5  Retries since fetch: 0
6  Retry interval: 2592000 seconds (30 days)
7  Score: 0.0
8  Signature: null
9  Metadata:
10
11 http://acs106.cankaya.edu.tr/    Version: 7
12 Status: 1 (db_unfetched)
13 Fetch time: Mon Dec 16 19:48:34 EET 2013
14 Modified time: Thu Jan 01 02:00:00 EET 1970
15 Retries since fetch: 0
16 Retry interval: 2592000 seconds (30 days)
17 Score: 0.0
18 Signature: null
19 Metadata:
20
21 http://acs205.cankaya.edu.tr/    Version: 7
22 Status: 1 (db_unfetched)
23 Fetch time: Mon Dec 16 19:48:34 EET 2013
```

Figure C- 2: Command to read crawldb

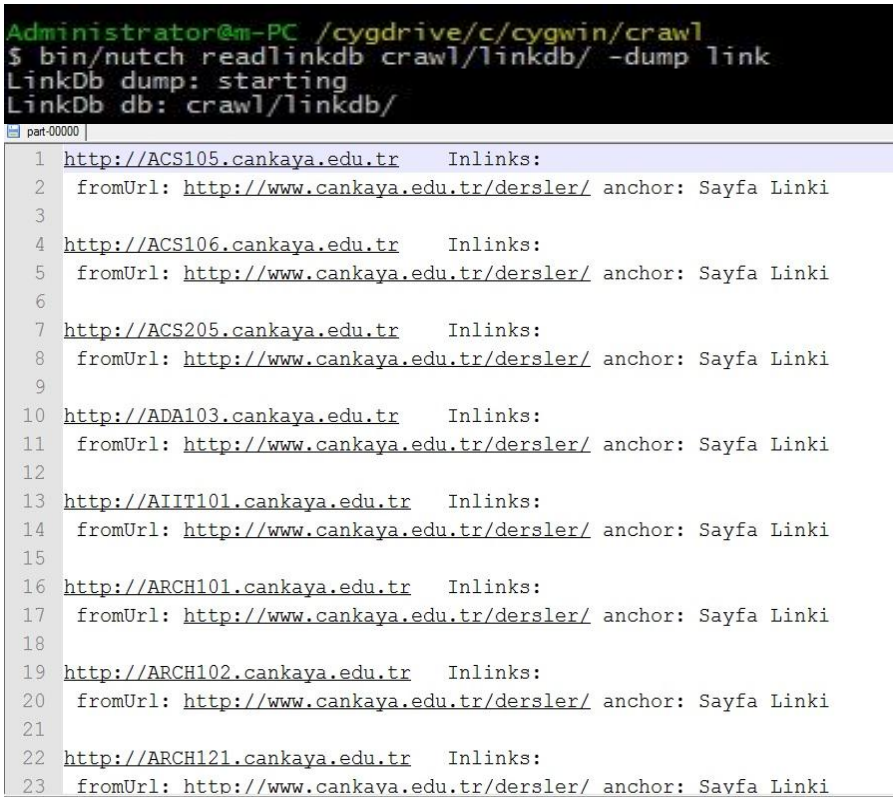


Figure C- 3: Command to read linkdb

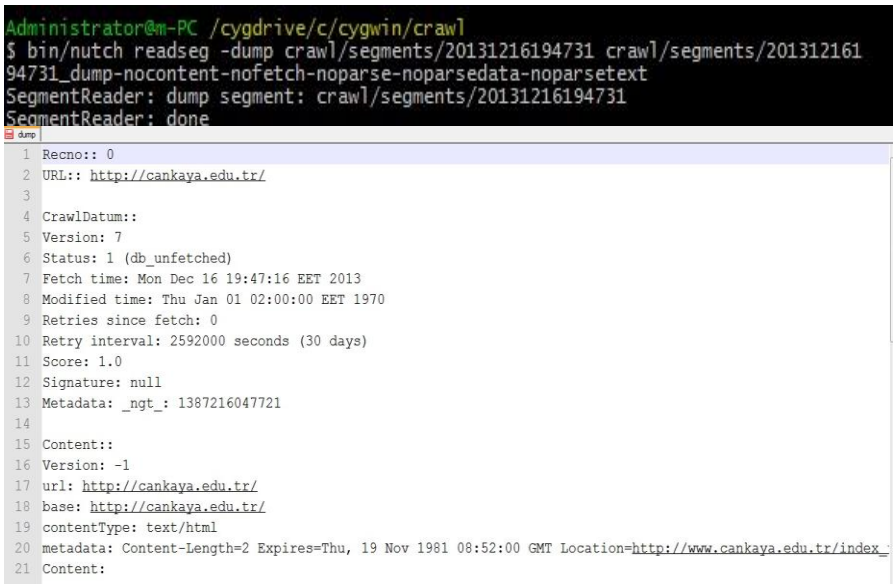


Figure C- 4: Command to read segments

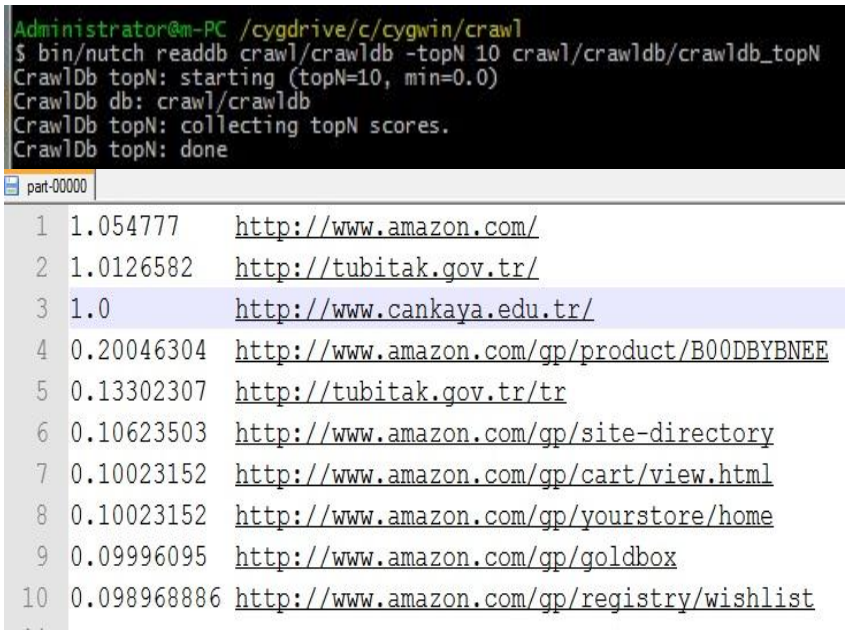


Figure C- 5: View the first ten URLs

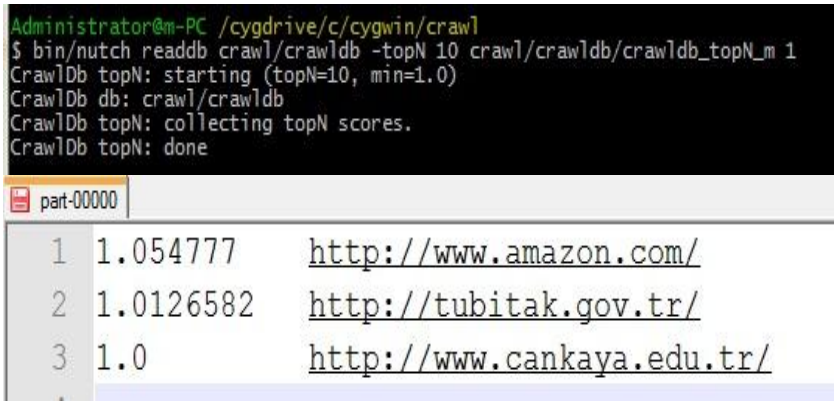


Figure C- 6: View the first three URLs

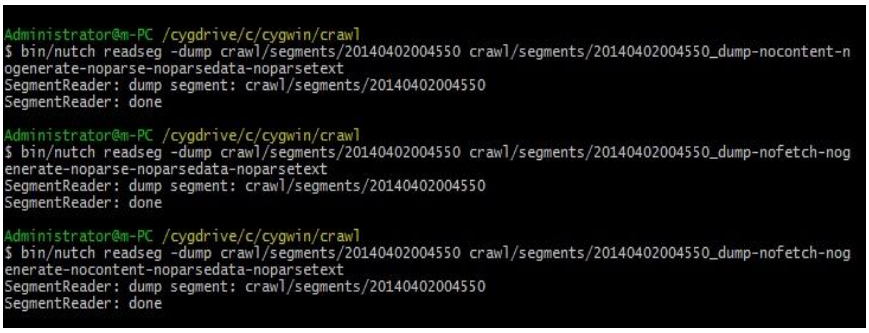


Figure C- 7: Read the segments

Figure C- 8: incoming anchor text

Nutch Web Crawler follow the (Breadth First Search Algorithm), to fetch the URLs from the Web, like Queue First in First Out (FIFO), Figure C-9 shows the Breadth First Search Algorithm.

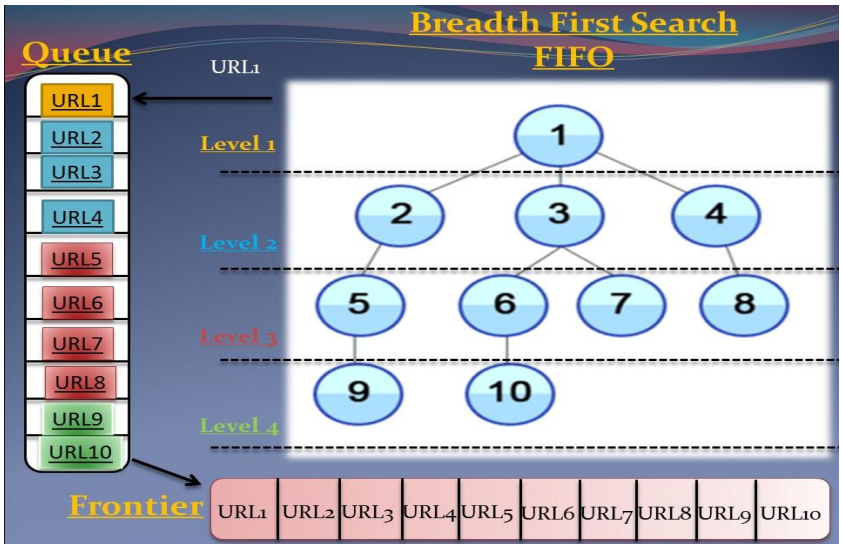


Figure C- 9: Breadth First Search Algorithm

APPENDIX D - QUESTIONNAIRE FOR PARTICIPANTS OF THE SURVEY

Before, I start to write my thesis i took the views of my colleagues (Survey Questionnaire), about my idea, it is how Search Engine and Web Crawler work does. Because more people or Internet users use a Search Engine to search about information, but they don't know how a Search Engine catches the information from the Web and displayed to users. Number of Survey Questionnaire is 50, who use the Search Engine .Figure D-1 below shows the rate the Search Engine users, who know how Search Engine works and who don't know, how does it work. And i took their opinion about work study, how search Engine, then i got good rate my idea, it is (Use Open-Source Search Engine). Figure D-2 below shows this rate.

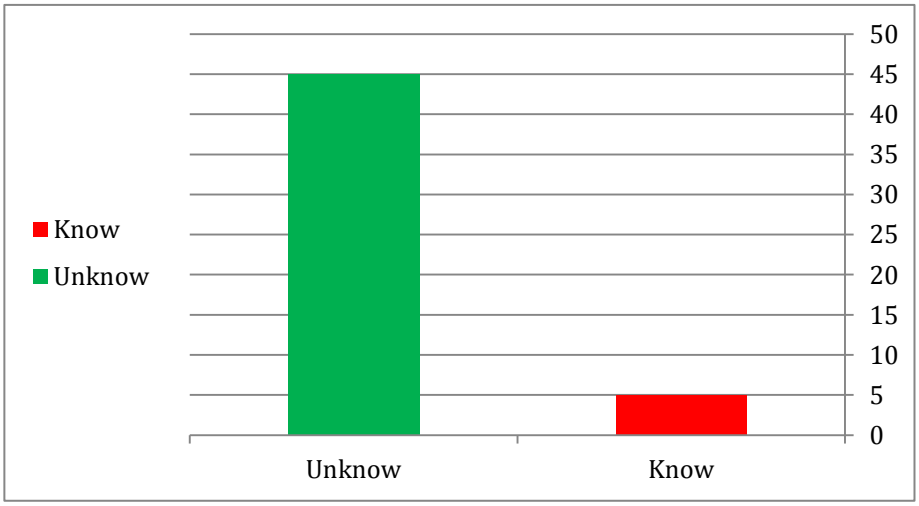


Figure D 1: Who know and don't know how a Search Engine works

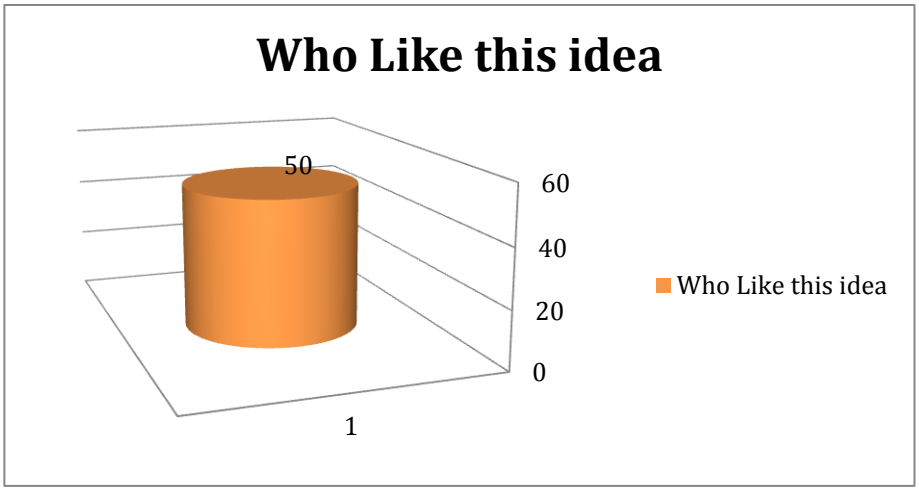


Figure D 2: Rate of who like this idea

The Table below shows the rate the people, who know and don't know how Search Engine works.

Table D 1 Survey Questionnaire

NO.	Name	Affiliation	Do you known, how Search Engine does work?	Like the idea (use Search Engine open-source to clarify mechanism of action Search Engine & Web Crawler)
1	Ali Adil Ali	Ministry of Education/ IRAQ	NO	YES
2	MAHER IBRAHIM	UPM-Remote Sensing & GIS /Malaysia	NO	YES
3	Ahmed Abdulrahman	Erciyes University/Computer Engineering	NO	YES
4	Sarah A. Al Doori	Florida Institute of Technology/ Electrical Engineering	NO	YES
5	Wedean AL Hadban	Florida institute of technology/ biology	NO	YES
6	Yaser Issam Hamodi	Ministry of Education/ IRAQ	NO	YES
7	Farah Qasim	Cankaya Uni/ English Literature	NO	YES
8	Omer Nizer	Cankaya University	NO	YES
9	Hussain Mahdi	UKM University	NO	YES
10	Sarmad AL-Saleahei	Diyala University/IRQ	NO	YES
11	Ali Nihad	Ministry of Education/ IRAQ	NO	YES
12	Ali Alrukaby	Mustansiriya University	NO	YES
13	Emad Alhadithi	Ministry of Finance	NO	YES
14	Omer Ahmed	Ministry of Education/ IRAQ	NO	YES
15	Emad Elhashimi	Cankaya University	NO	YES
16	Omar Saad	Ministry of Education/ IRAQ	NO	YES
17	Wesam Raad	Al-Mansour University	NO	YES
18	Ali Majid	Cankaya University	NO	YES
19	Abdulrahman S. Alkateb	Çankaya Üniversitesi	NO	YES
20	Ahmed Rada	Ministry of Education/ IRAQ	NO	YES
21	Hussam Faisal	Al-Rafidain University/ College	NO	YES
22	Duraid Yehya Mohamm	Al-Masour University/ College	NO	YES
23	Mohamed Adil Ali	College of Medicine/IRQ	NO	YES
24	Sarah Othman	Baghdad University	NO	YES
25	Ali Raad	Technology University	NO	YES
26	Abdul Rahman Alazawee	Mnistry of Health	NO	YES
27	Firas Al-tememi	Cankaya University	NO	YES

28	Haider Bander	Cankaya University	NO	YES
29	Dr-Mohammed Fauzi	AL-Nahrain University/IRQ	NO	YES
30	Asaad Qasim	Ministry of Education/ IRAQ	NO	YES
31	Janan Farjo	Çankaya University	NO	YES
32	Emad Majed Al Zebari	Çankaya University	NO	YES
33	Omer Subhi	Çankaya University	NO	YES
34	Hamadalla Alsultany	Al-Masour University/ College	NO	YES
35	Ziyad Abbas	Al-Rafidain University	NO	YES
36	Yahya Arslan	Çankaya University	NO	YES
37	Hayder Kubba	Ministry of Education/ IRAQ	NO	YES
38	Ammar Abotabek	Çankaya University	NO	YES
39	Dr. Mazin AL-Leheibe	Al-Mustansiriya College of Medicine	NO	YES
40	Othman Subhi	Çankaya University	NO	YES
41	Raad Ali Ameen	Çankaya Üniversitesi	NO	YES
42	Ali Muneer	Al-Masour University/ College	NO	YES
43	Sara Mohammed	Technology University	NO	YES
44	Hajer Othman	Baghdad University	NO	YES
45	Maysaa Adil Ali	Florida Institute of Technology/ biology	NO	YES
46	Ahmed faris al-kaisi	Baghdad College of Economic Sciences University \ Computer Science	YES	YES
47	Alaa najah	Baghdad mass media/IRQ	YES	YES
48	Ahmed Al-Azzawi	Cankaya Uni – IT	YES	YES
49	Mohammed K. Hussein	IT /Cankaya University	YES	YES
50	Harith Mahdi	Cankaya University/ Turkey	YES	YES

APPENDIX E – LUCENE SUPPORTS PAGERANK

The important part in Search Engine is ranking the results to the users. After storage the documents by Lucene on local disk or hard disk, Lucene works to removing the stop words and stemming words. Then Lucene works to give ID and location to each document were stored. Lucene has the IndexSearcher. It is created using an analyze. By it Lucene take the user's query and analyze it, to find out where the user is seen. The IndexSearcher can access to the documents and ranking the results according user's query. Figure E-1 below shows the user's query [63].

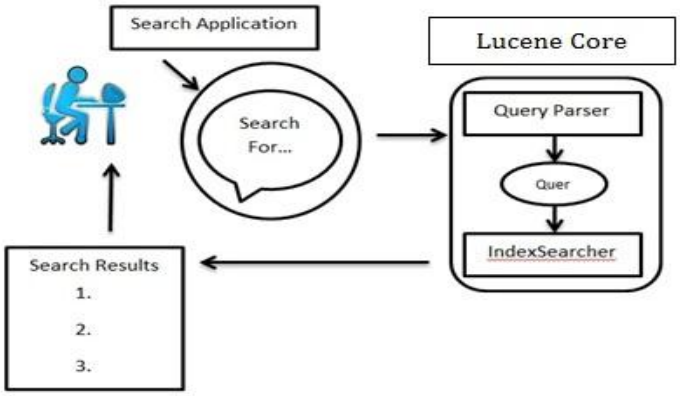


Figure E-1: Process of Search Results

We used this URL (http://www.edmunds.com/) this Website about car selling, and we chose three different keywords are (Car, Ford and BMW), and we used logical operation like (AND, OR and NOT), and we got on results in figures below. In Figure E-2 below shows the Lucene made ranking results only about "Car Keyword"

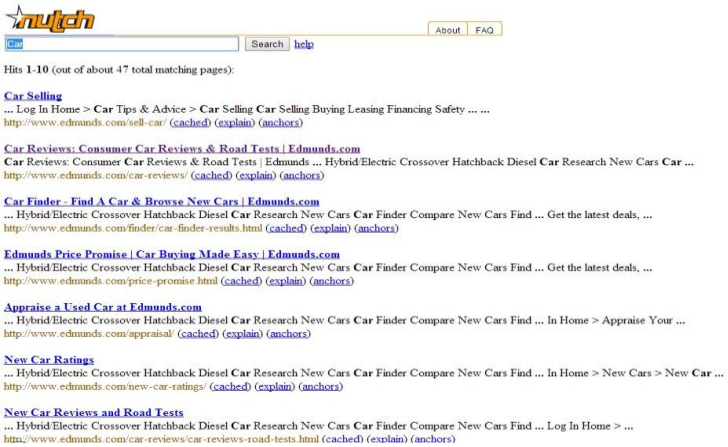


Figure E-2: Search Result about "Car Keyword"

In Figure E-3 below shows the Lucene made ranking results only about "BMW Keyword"

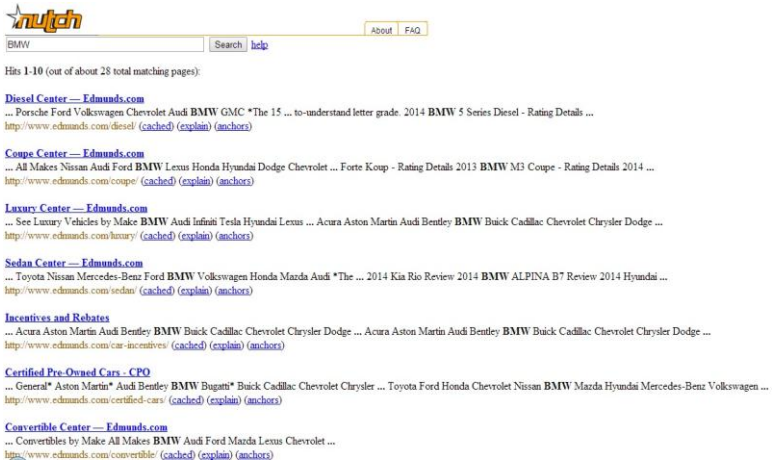


Figure E-3: Search Result about "BMW Keyword"

In Figure E-4 below shows the Lucene made ranking results only about "Ford Keyword"

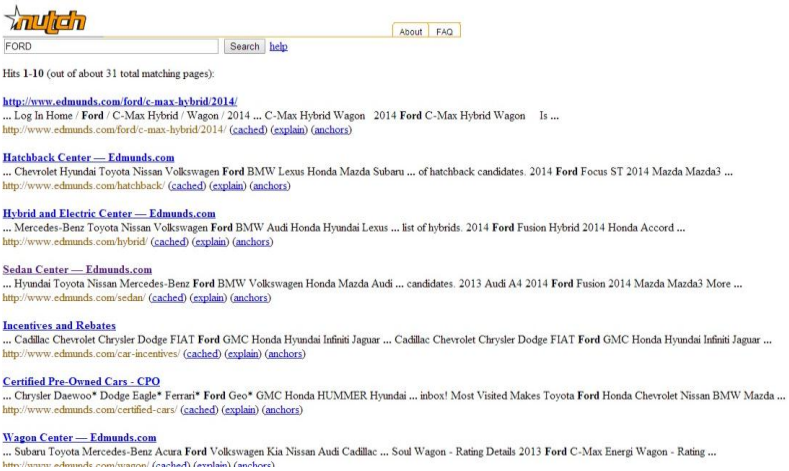


Figure E-4: Search Result about "Ford Keyword"

In Figure E-5 below shows the Lucene made ranking results only about "Ford and BMW Keywords" with use logical operation is (AND).

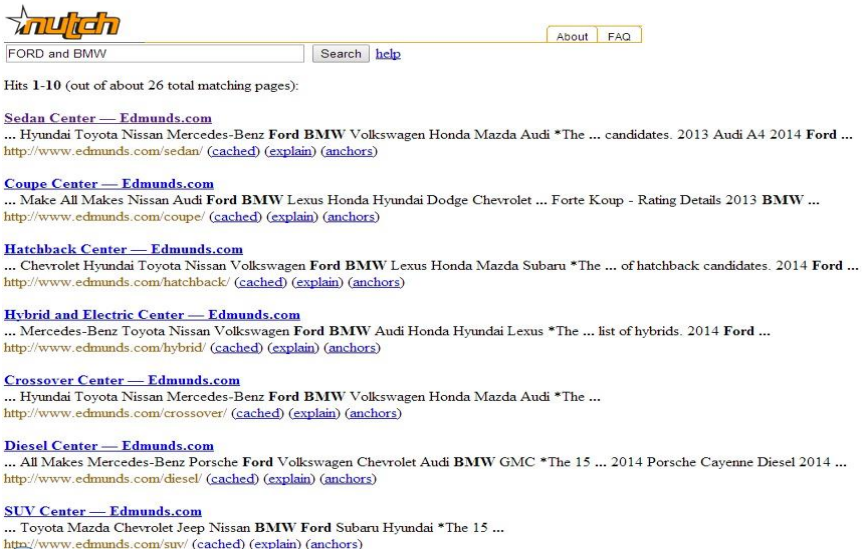


Figure E-5: Search Result about "FORD and BMW Keywords"

Web Crawler jumps or moving from Webpage to another one to catch more Webpages and it works to index them. Figure E-5 below shows the Web Crawler path.

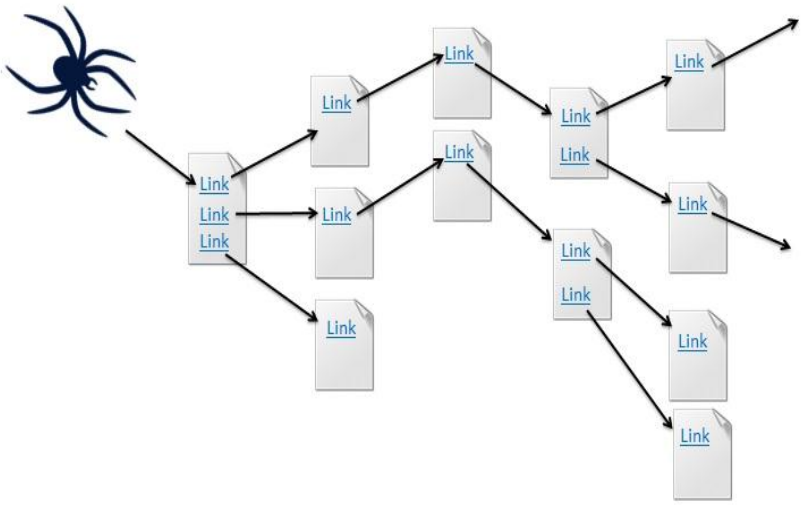


Figure E-6: Web Crawler Path

APPENDIX F – SYSTEM REQUIREMENTS

To run this Search Engine and Web Crawler in this study, we used software and Hardware requirements, they are:

Software Requirements:-

1- Nutch 1.1: http://www.apache.org/dyn/closer.cgi/lucene/nutch/.
2- JAVA JDK 6 update 3: http://java.sun.com/javase/downloads/index.jsp/.
3- Apache web server 6: http://tomcat.apache.org/download-60.cgi.
4- Cygwin: http://www.cygwin.com/.
5- Operating System (Windows 7).

Hardware Requirements:-

1- Processor 2.20 GHz.
2- RAM 2.00 GB.
3- Hard Disk 298 GB.

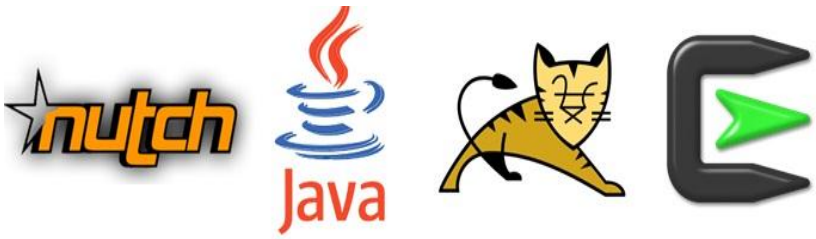


Figure F-1: System Software Requirements

APPENDICES G

CURRICULUM VITAE (CV)

PERSONAL INFORMATION

Surname, Name: Abdulwahid, Nibras
Date and Place of Birth: 10 March 1984, Baghdad
Marital Status: Married
Phone: +90 5370344044
Email: an_angel2011@yahoo.com

EDUCATION

Degree	Institution	Year of Graduation
M.Sc.	Cankaya Univ. Computer Sciences	2014
B.Sc.	Al-Rafidain Univ. Computer Sciences	2006
High School	Arab Revolt	2001

WORK EXPERIENCE

Year	Place	Enrollment
From 2007 to now	Ministry Of Education	Specialist

FOREIGN LANGUAGES

Advanced English, Beginner Turkish.

PUBLICATION

- **ABDULWAHID N (2014)** "*Improve the Performance of the Work of the Restaurant Using PC Touch Screen*". J Comput Sci Syst Biol 7: PP.103-107.

HOBBIES

Travel, Reading, Swimming, Fashion and cooking, Planning and Design, drawing.