International Journal of Web Engineering 2012, 1(3): 15-23
DOI: 10.5923/j.web.20120103.01

Design and Evaluation of Semantic Guided Search Engine

Salah Sleibi Al-Rawi[1,*], Ahmed Tariq Sadiq[2], Sumaya Abdulla Hamad[3]

[1]Information Systems Department, College of Computer , Anbar University, Ramadi-Anbar, Iraq
[2]Computer Science Department, University of Technology, Baghdad, Iraq
[3]Computer Science Department, College of Computer, Anbar University, Ramadi-Anbar, Iraq

Abstract Search engines provide a gateway through which people can find relevant information in large collections of heterogeneous data. Search engines efficiently service the information needs of people that require access to the data therein. Web search engines service millions of queries per day, and search collections that contain billions of documents. As the growth in the number of documents that are available in such collections continues, the task of finding documents that are relevant to user queries becomes increasingly costly. In this work, Semantic Guided Internet Search Engine is built to present an efficient search engine – crawl, index and rank the web pages – by applying two approaches. The first one, implementing Semantic principles through the searching stage, which depends on morphology concept – applying stemming concept – and synonyms dictionary, and the second, implementing guided concept during input the query stage which assist the user to find the suitable and corrected words. The advantage of guided concept is to reduce the probability of inserting wrong words in query. The concluded issues in this research that the returned web pages are semantic pages yielding with synonyms depending on the query terms which achieved the concept of semantic search and as compared with Google, good results are appeared depending on the Recall and Precision measurements reaching 95% - 100% for some queries in spite of the differential of environment between the two systems. Also, the performance of the search is improved by using guided search and by using the improved PageRank, which reduces the retrieved time. Finally removing stop words from a document minimizes the storage space, which enhanced the proposed system.

Keywords World Wide Web, Information Retrieval, Semantic Search, Guided Concept

1. Introduction

Searching for information on the Web is, for most people, a daily activity. Search and communication are by far the most popular uses of the computer. Not surprisingly, many people in companies and universities are trying to improve search by coming up with easier and faster ways to find the right information. These people, whether they call themselves computer scientists, software engineers, information scientists, search engine optimizers, or something else, are working in the field of information retrieval. Information retrieval is a field concerned with the structure, analysis, organization, storage, searching, and retrieval of information, which involves a range of tasks and applications[1].

A search engine is a practical application of information retrieval techniques to large-scale text collections[2]. The usual search scenario involves someone typing in a query to a search engine and receiving answers in the form of a list of documents in a ranked order. The big issues in the design of search engines include the ones identified for information retrieval: effective ranking algorithms, evaluation, and user interaction. There are a number of additional critical features of search engines performance of the search engine in the terms of measures such as response time, query throughput, and indexing speed, where response time is the delay between submitting a query and receiving the result list, throughput measures the number of queries that can be processed in a given time, while the indexing speed is the rate at which text document can be transformed into index for searching. An index is a data structure that improves the speed of search. Coverage measures how much of the existing information, in a corporate information environment, that has been indexed and stored in the search engine, while recency or freshness which measures the age of the stored information[1, 2].

In this paper, two different issues will be discussed; these issues are Semantic principles within searching and Guided concept, within search engine query (user interaction).

The aim of this study is to design and implement an improved semantic guided Internet search engine, which can help end users to extract query terms by using a guided list that is obtained from a dictionary database and to prevent the wrong input (spelling mistakes), save the time and keystrokes. On the other hand, it shows how to combine the improved PageRank algorithm with the proposed system to reduce the number of iterations, also how the semantic search returned web pages yielding with synonyms de-

* Corresponding author:
dr_salah_rawi@yahoo.com (Salah Sleibi Al-Rawi)

pending on the query terms to make searching process near to the natural language.

The rest of this paper is organized as follows. Section 2 presents information retrieval and search engine. We present the principle of search engine, and its components, semantic search and synonyms, and a guided concept. Section 3 presents proposed a system design, its architecture and algorithms. Section 4 presents the implementation of this system. Section 5 presents the evaluation of the proposed system. and finally section 6 concludes the paper and presents future works.

2. Information Retrieval and Search Engine

Baeza-Yates and Ribeiro-Neto define Information Retrieval (IR) as the “part of computer science which studies the retrieval of information from a collection of written documents[3].

The retrieved documents aim at satisfying a user information and usually need to be expressed in natural language.” Salton and McGill note that “information retrieval is concerned with the representation, storage, organization, and accessing of information items”[3].

Clearly, this full description of the user information need cannot be used directly to request information using the current interfaces of Web search engines. Instead, the user must first translate this information need into a query which can be processed by the search engine (or IR system). In its most common form, this translation yields a set of keywords (or index terms) which summarizes the description of the user information need. Given the user query, the key goal of an IR system is to retrieve information which might be useful or relevant to the user. The emphasis is on the retrieval of information as opposed to the retrieval of data[2]. In data retrieval, the result of a query must be accurate: it should return the exact match tuples of the query, no more and no less. If there is no change to the database, the result of a query executed at different times should be the same. Data retrieval deals with data that has a well-defined structure and semantics (e.g. a relational database). The goal of an IR system is to retrieve all the documents, which are relevant to a query while retrieving as few non-relevant documents as possible. To achieve this goal, IR needs users to provide a set of words which convey the semantics of the information need. Also, a document is represented by a set of keywords or index terms for IR to extract. These keywords or index terms can be derived from information experts or a computer through eliminating articles and connectives, the use of stemming (which reduces distinct words to their common grammatical root), and identifying nouns (which eliminates adjectives, adverbs, and verbs).

There is a difference between IR and searching the Web. IR allows access to whole documents, whereas, search engines do not. The reason is that it is too expensive to store all the Web pages locally and too slow to access remotely on other Web servers[4].

Thus a search engine is the practical application of information retrieval techniques to large scale text collections [1, 16,19].

2.1. Search Engines Components

Basically, a search engine is a software program that searches for sites based on the words that you designate as search terms. Search engines look through their own databases of information in order to find what it is that you are looking for[5].

Search engines are the key to finding specific information on the vast expanse of the WWW. Without sophisticated search engines, it would be virtually impossible to locate anything on the Web without knowing a specific URL[6].

Search engines are not simple. They include incredibly detailed processes and methodologies, and are updated all the time. All search engines go by this basic process when conducting search processes, but because there are differences in search engines, there are bound to be different results depending on which engine you use.

1. The searcher types a query into a search engine.
2. Search engine software quickly sorts through literally millions of pages in its database to find matches to this query.
3. The search engine's results are ranked in order of relevancy[5].

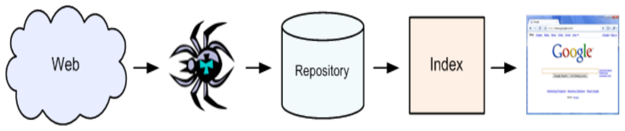

Figure 1. Crawling the Web, indexing, and retrieved when using a search engine[8]

Same search on different search engines produces different results because not all indices are going to be exactly the same. It depends on what the spiders find or what the humans submitted. But more important, not every search engine uses the same algorithm to search through the indices. The algorithm is what the search engines use to determine the relevance of the information in the index to what the user is searching for[6]. Search engines have three major elements. First is the spider, also called the crawler. The spider visits a web page, reads it, and then follows links to other pages within the site. This is what it means when someone refers to a site being "spidered" or "crawled." The spider returns to the site on a regular basis, such as every month or two, to look for changes. Everything the spider finds goes into the second part of the search engine, the index. The index, sometimes called the catalog, is like a giant book containing a copy of every web page that the spider finds. If a web page changes, then this book is updated with new information. Sometimes it can take a while for new pages or changes that the spider finds to be added to the index. Thus, a web page may have been "spidered" but not yet "indexed." Until it is indexed -- added to the index -- it is not available to those searching with the search engine. Search engine software is the third part of a search engine. This is the program that sifts through the millions of pages recorded in the index to find matches to a search and rank them in order of what it believes is most relevant[7]. The fig. 1 shows how the Web is crawled and placed into a local repository where it is indexed and retrieved when using a search engine[8].

So, in the third part there is a query processor which has several parts, including the user interface (search box), the "engine" that evaluates queries and matches them to relevant documents, and the results formatter. We can see that in the following figure[9]:

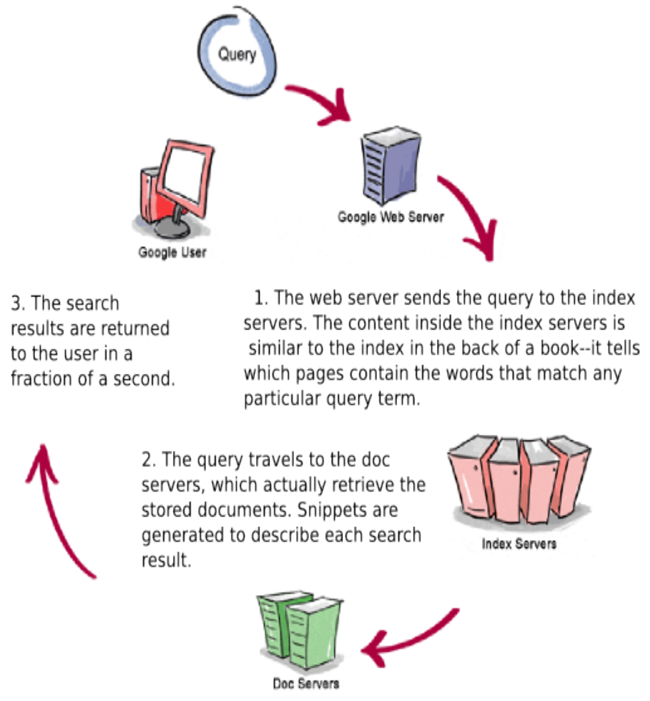
Query
Google Web Server
Google User
3. The search results are returned to the user in a fraction of a second.
1. The web server sends the query to the index servers. The content inside the index servers is similar to the index in the back of a book--it tells which pages contain the words that match any particular query term.
2. The query travels to the doc servers, which actually retrieve the stored documents. Snippets are generated to describe each search result.
Index Servers
Doc Servers

Figure 2. Query processing[9]

2.2. Semantic Search

Sometimes it is difficult to find information that exactly matches the users' question, when we don't have to provide "keywords" or, "Boolean operators" Instead, we type our question in our "natural language", the language we might use when asking a question of a knowledgeable adult.

Natural language would be much easier for users. They could ask search engines like they ask normal people. To make this possible the search engine needs to "understand" the natural language query. Approaches are made to provide this feature to the user.

The problem with the search engines of today is that they lack of intelligence. The search engine can only find pages that have the chosen key/search/content word in the text. If we, for instance, are in need of information on "*Learn English language*" the search engines will only find pages which have the words *Learn* and *language* on them. Pages regarding "*Study English lingo*", "*Educate English linguistics*", or "*Learn English tongue*" will not appear in the search engine result pages even though they are or could be very relevant. If we know very little about *Learn English language*, you will, perhaps, never consider searching for these words; therefore, never find this relevant information[10, 17].

Semantics search engine helps end user's on World Wide Web to use several words which have the same meaning to make searching process near to the natural language.

Thus we develop a new search engine to process the end user query semantically, using morphology concepts and synonyms dictionary.

2.2.1. Synonyms

Synonyms are different words with similar or identical meanings. Antonyms are words with opposite meanings.

Examples of synonyms are the words cat and feline. Each describes any member of the family feline. Similarly, if we talk about a long time or an extended time, long and extended become synonyms[11, 18].

Examples of other English synonyms are:

Car and automobile
baby and infant
student and pupil
pretty and attractive
smart and intelligent
sick and ill
funny and humorous
died and expired

Synonyms can be nouns, adverbs or adjectives, as long as both members of the pair are the same part of speech.

Some lexicographers claim that there are no synonyms that have exactly the same meaning (in all contexts or social levels of language); however, most speakers of languages sense that some synonym pairs are identical in meaning, for all practical purposes. Different words similar in meaning usually differ for a reason; feline is more formal than cat; long and extended are only synonyms in one usage and not in others, such as a long arm and an extended arm.

The purpose of a thesaurus is to offer the user a listing of similar or related words; these are often, but not always, synonyms. See hyponym for a closely related phenomenon, "words included in other words", as tulip is included in flower, but not vice-versa.

2.3. Guided Concept

Guided means the suggestion words that appear when the user of search engine typing the query in the search box using drop down list. Google amuses us with its features and we have already observed this at 'Google Suggest', which analysis your web history and other Zeitgeist data to display you relevant words [12].

Google has now officially launched 'Google Suggest', a feature for search box that will suggest some relevant queries to user. The suggestions are based on popularity of terms and Zeitgeist data. You can call it a real time alternative of "Did you mean?" feature.

According to the Google blog, suggest feature will help user in following ways:

1. To reduce spelling mistakes.
2. To save time and keystrokes.

3. To help in formulating user search i.e. automatically make search queries for user[13].

Google suggest is a feature that recommends you keywords based on what you type in Search box. For example, when you type "y" in search box, Google assumes that you are searching for YouTube or Yahoo. As you type more alphabets, Google will suggest you some more queries in an effort to save users time, as we do not have to type whole search words. Suggestions are very helpful because of several reasons like you skip spelling mistakes and minimize effort in typing long sentences. Google suggestions are based on particular user's history and other user's behavior all over the world and popular searched made during that time. However, some people do not prefer Google suggestions because sometimes it is annoying, it shows explicit results and many times a user loses his focus from the original search to some more appealing content. For example: While searching for "windows seven", Google suggests "Windows seven transformation pack" which is not legitimate, as shown by below figure[13, 14].

3. A Proposed System Design

In this section several improvements are proposed by constructing a semantic guided search engine. The goal from these propositions is to retrieve a well ranked result. This will be achieved by implementing the following:

1. Improving indexing process by extracting several factors to provide more information about content of the page and retrieve a well and more ranked result.
2. Implementing a ranking system which computes the rank score depending – by first – on hyperlink structure and features among web pages and secondly depending on the relevancy between the given query and web pages.
3. Improving searching process by applying semantic principles – that means searching not only for the query terms but for their synonyms –
4. Using guided concept to improve query and help end user when they input query in search box.

3.1. Proposed System Architecture

Basic architecture of proposed system is shown in fig.3 starting with the user query when user posts it to the search engine interface. The guided phase was implemented at the same time of inputting the query by drop down list to suggest the query terms. Next the valid query is posted to the searcher which performs several operations to retrieve the required documents from the repository to be ranked by the ranker and browsing the result in a form of sorted list of pages according to their ranking score.

This will be accomplished by two basic parts depending on user relativity with the system. First one is off-line part, and the second is on-line part. Both of them are described in details in the next two sections.

3.2. Off-line Part

This part of the system consists of several sub-systems and databases, which will be run independently from the end user. These can be described by the following:

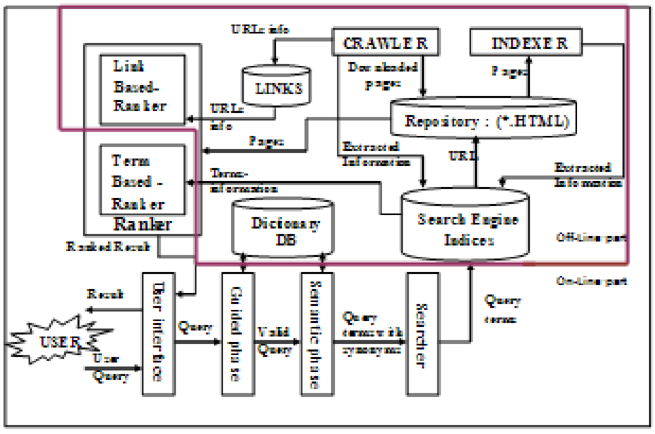

Figure 3. Functional Diagram for the Proposed Search Engine

3.2.1. Crawler

Crawler is the application that is interacting with web servers which are all beyond the control of the system. Using proposed search engine more than (350) web pages have been downloaded as the main data set and by implementing the crawling algorithm (algorithm(1)) more than (6600) web pages have been crawled. Each web page, that crawler starts with; has a unique identification number (page-id). The crawler begins to find all hyperlinks ending with .html or .htm, then the crawler checks each one of these links whether it is retrieved or not. If it is not, its URL will be extracted and assigned a unique page-id and stored into the URL-list.

The crawler has another function that extracts important information about links –link attributes will be used in the pagerank computation – these attributes extracted depend on links between source and destination pages by parsing these source pages.

It is worth mentioning that those numbers were chosen arbitrarily by the author.

There are three important attributes; the first one is the visibility of the link which is determined by the specific tags of HTML page. These tags are <B> which means the bold style of the link's text and <I> which means the italic style of the link's text. If one of these tags appears in the text the visibility value equal to 3, when they appear at the same time the visibility value is equal to 1. The second attribute is the position of the link within the source page which can be determined by finding position of first word in the link's text within the source page which is partitioned into three parts. If the link appeared in the first part, the link's position value will be equal to 3, if it appears in the second part, the value will be equal to 2, finally if the link appears in the third part, the value 1 will be equal to the link position.

The distance between the source and the destination web pages is the third attribute which can be computed by finding the degree of differences between two host-names in the URL for sources and destinations web pages. If the host-names are different the value will be equal to 5, and it will be equal to 1 if the host-names are equal.

If these operations have been accomplished for the web page, it will be stored into the search engine indices. Then the crawler repeats the process over another web page which pulls it from URL-list until un-crawled web pages exist.

Below the algorithm that can be used to implement a web crawler

Algorithm(1): Crawler Algorithm

Input: Set of URLs

Output: URLs table, Link-Information table, Hyperlink table

```
Begin
Add the URL to the empty list of URLs to search.
While not empty (the list of URLs to search)
begin
Take the first URL in from the list of URLs
Mark this URL as already searched URL.
If the URL protocol is not HTTP then
Break;
Go back to while
Open the URL
If the opened URL is not HTML file then
Break;
Go back to while
Extract information for link-information
table and hyperlink table
Iterate the HTML file
While the html text contains another link
begin
If the opened URL is HTML file then
If the URL isn't marked as searched then
Mark this URL as already searched URL.
Else
If type of file is user requested
Add to list of files found.
End While
End While
End of Algorithm
```

3.2.2. Indexer

Indexer is the most complicated process in constructing a search engine that can be used for extracting and keeping information about each word that occurs in web page and useful information that describes the content of the indexed web page. This process performs many steps after reading the web page as a normal text file.

The first step is page parsing which deals with fetching each page from repository and parses it into a text file that contains the most important HTML tags generating lexical tokens from this text file.

The next step is extracting the keywords and their attributes by checking each token whether it is HTML tag or real word. If token is a real word then three another steps are implementing:

- Stemming words to improve search engine performance by reducing the size of lexicon table.
- Removing the noisy words by using stop-word dictionary.

Note that: Each one of these tables (lexicon, stop-word) exists in the search engine indices.

On the other hand, if the token is HTML tag then it will be processed in the last step at which the inverted index table and other important tables are created. This can be done by extracting the attributes from HTML tags (of its word(s)) which are processed by the tag-process (token) procedure. Depending on each tag type there are different steps of processing.

- Also if a token is a real word then extracting word descriptor attributes (word position, word importance, word style, word size, word font color, word font face) will be performed and stored in the inverted index with the word-id which will be fetched from lexicon table. The inverted index maps between word-id and page-id where the word occurred. These tables (lexicon, inverted index) are found in the search engine indices.

During index processing other two types of attributes are extracted, the first type is concerned with the information of the page (Page Title, Author Name, Keywords used in the web page, Descriptor Terms, Publishing Date, Modification Date). These attributes are stored in page information table which exists in search engine indices.

The second type of attributes is concerned with the description of the page (number of words in the web page, the general font face, the general font color, the general of font size). These attributes are stored in the page form information table; also it exists in the search engine indices.

All of these attributes will help in the Term-Based Ranking phase. The following is a simple description of indexer algorithm

Algorithm(2): Indexer Algorithm

Input: P set of web pages

Output: Inverted Index table, Page-Information table, Page-Form-Information table

```
Begin
For all webpage p Є P do
Open p
Parse p into tokens T
Extract information for each t Є T
Check if t is tag then
Set the attributes of p into Page-Information table and Page-form information table
Else
Process t as a real word
Remove stop-words
If   not (t Є Inverted Index table) then
Add t into Inverted Index table
Set attributes of t into Inverted Index table
End for
End of Algorithm
```

3.2.3. Repository

The repository is used to store all the web pages that are fetched by the crawler (with type of .html and .htm) and processed by Indexer.

3.2.4. Search Engine Indices

Each search engine has a database which consists of many tables that can be used during different phases in its processing. The proposed system has a database which is created by SQL server and consists of the following tables: URL table, Lexicon table, Stop-word table, Synonyms table, Inverted Index table, Page-Form-Information table, Guided table, Link-Information table, and Hyperlink table.

3.2.5. Link-Based Ranker

In this study, this subsystem is implemented using Improved PageRank algorithm which is described as follows, for more details the interested readers are referred to [15],

Algorithm(3): Link-Based Ranking Algorithm
Input: webpage, and hyperlink
Output: PageRank for each webpages

```
Begin
while no_error
for all Pi in web retrieve_factors(Pi)
for all Pj points to pi
$PR(Pi)=(1-d)+d*[\sum_{J<i}(\frac{PR(Pj)l+1}{C(Pj)}*L(Pj,Pi))+\sum_{j\geq i}(\frac{PR(Pj)l}{C(Pj)}*L(Pj,Pi))]$
        err=abs[PR(Pi)l-PR(Pi)l+1]
        if err>0.000001 then
            no_error=false
        else no_error =true
    end [while]
```

End of Algorithm

Where :
PR(pi): pagerank of the page pi
C(pj): the number of outbound link on page pj
d: is a damping factor which can be set between 0 and 1
L(pj,pi): consist several factors(visibility of link, position of link within a web page, and distance between the web pages – section 3.2.1-

3.3. On-line Part

This part runs along with the end-user (query-based process). The following processes describe how this part runs:

3.3.1. User Interface

At first, In this phase guided concept is applied, when the user entered the beginning of query, query suggestions are displayed in a list, the right query can be selected directly from this list by clicking on it. This process optimizes query process by reducing spelling mistakes, saving time and keystrokes, and automatically makes search queries for user. The following is the guided algorithm.

Algorithm(4): Guided Algorithm
Input: the first characters of the first term of the query
Output: terms of the query

```
Begin
If no character entered then
All terms or string begining with a..z is appeared in dropdown list
Else
While characters are entered
Dropdown list terms are reduced to only terms or strings begining with these characters
Check dropdown list
If the query is found then
Choose the query from dropdown list and begin search
Else Return empty query
```

End of Algorithm

The above algorithm represents the query interface; the other part of the main interface is the answer interface in which the returned ordered relevance pages appear with brief properties such as a title of page, an URL, a date, and the author name. These results appear after search processing (searcher) upon the query is applied.

3.3.2. Searcher

This is the main process in the on-line part which selects the relevant pages. It begins when the query entered and search button is clicked. Each word in the query is matched with the Lexicon table in Search Engine Indices to retrieve word_id and some information from Inverted index table depends on word_id such as page_id, position, importance, style, color, face, and size. For any page_id which has been retrieved another information such as number of words, page font face, page font color, and page font size are retrieved from Page-Form-Information table. Then each word stored in a temporary buffers with its information which is treated as factors used by the Term-Based Ranker in the next phase. Algorithm(6) describes this procedure. The Proposed Search Engine supports Semantic Search which means that when the word_id is retrieved from Lexicon table, the Synonyms related to this word are retrieved from Synonyms table and stored with the related word in temporary buffers (synonyms table is described in table (1) below). This process improves search engine by retrieving more relevant pages that depends not only on query, but also on the meaning of the query. Semantic procedure is described in algorithm(5) below.

Table 1. Synonyms Table

words	Synonym1	Synonym2	Synonym3	Synonym4
Where : Words » the set of words Synonym1, Synonym2, Synonym3, Synonym4 » the different meaning of word				

Algorithm(5): Semantic Procedure
Input: terms of the query
Output: terms of the query with synonyms of each term

```
Begin
Call SQL procedure
While term = word in the synonyms table do
Select from synonyms table syno1, syno2, syno3, and syno4
```

```
Return each term with its synonyms
Begin search
End of Algorithm
```

```
Algorithm(6): Search Procedure
Input: terms of query with their synonyms
Output: array of attributes
Begin
Call SQL procedure
While term = word in the lexicon table do
Select from Inverted Index table p-id, pos, loc, font, color, style, and size
Select from page-form-information table page color, page font, and no-of-words
Return
End of Algorithm
```

The searcher algorithm can be illustrated in the following:

```
Algorithm(7): Searcher Algorithm
Input: user query after guided phase
Output: Ranked WebPages
Begin
Check the query
If empy then Return empty webpage
Else
Call Semantic Procedure
Call Search Procedure
Call Ranking system
Order webpages depending on Ranking value
Show webpages
End of Algorithm
```

3.3.3. Term-Based Ranker

Ordered relevant pages will be retrieved to the user interface. This can be implemented by using improved Term-Based Ranking algorithm which is illustrated in the following (for more details look at [15]):

```
Algorithm(8): Term-Based Ranking Algorithm
Input : Set of attribute
Output: ranked webpages
Begin
[1]for all pages in Retrieved_pages_list
for all terms in Query
for each occurance of termj in pagei
attributej = term_attribute(factors)
    $S(i,q) = \sum_{Termj \in q} ((0.5 + 0.5 . \frac{TF(i,j)}{Tfmaxi}) . IDFj . attributej)$
    $R(i,q) = S(i,q) + \sum_{J=1,j\neq i}^{N} \alpha . Li(i,j) . S(i,q)$
go to [1]
End of Algorithm
```

Where:

S(i,q): TFxIDF of page i with respect to query q

TF(i,j): the term frequency of query j in page i

TFmaxi: the maximum term frequency of a keyword in page i

$$IDFj: \log(N/\sum_{i=1}^{N} Ci,j)$$

C(i,j): occurrence of j-th query in the i-th page

R(i,q): Vector Spreading Activation

Attributej: summation of factor (importance, position, style, font face, and color of the term in query j).

4. Implementation

The Proposed Search Engine involves two main parts, Off-line part (server part) and On-line part (client part).

At the server part a database is created by running several software, Crawler, Indexer, and Link-Based Ranker (PageRank) to extract several important information using to retrieve more relevant pages. The Crawler crawls WebPages based on hyperlink, and then each webpage indexes and all the words in this page, except stop words, are extracted with title, keywords, and other information from HTML document and indexed in the database. This increases the probability of getting the relevant pages to a query.

On the other hand, at the client part the query entered and the interface between the client and server retrieved the matched pages by matching the query with the database (Search Engine Indices). But there is one point to note here; the synonyms of the query words are obtained before they are searched in the database to provide more related results.

By using SQL statements the searcher can obtain the matching entries of database in the server side with the query and related information such as title, URL, author name, and the matching portion from the content of the corresponding entry. The Ranking algorithm retrieved more ordered relevant pages with the user query on the client interface.

Table 2. Searching Result, Recall and Precision Measures, and Retrieved Time

Query	Semantic Results	Recall	Precision	Retrieved Time/sec
Network protection	57 of 60	95 %	100 %	00:00:41.5468750
Feigned intelligent	15 of 19	79 %	100 %	00:00:11.1718750
Image action	7 of 9	78 %	100 %	00:00:13.2968750
Image processing	19 of 25	76 %	100 %	00:00:27.7500000
Figure performance	7 of 7	100 %	100 %	00:00:11.4218750
Edge detection	1 of 1	100 %	100 %	00:00:02.7500000

5. Evaluation

Effectiveness is the important performance measure to evaluate the proposed search engine. Several metrics to measure the effectiveness of search systems have been proposed, with each metric varying in the emphasis of the qualities measured.

Two intuitive measures for retrieval systems are the notions of recall and precision.

In this work, six queries are tested and evaluated by Recall and Precision of the semantic results then retrieved time is calculated in proposed system and its values are accepted as shown in the table (2). After testing we note that the proposed system retrieved semantically WebPages and as compared with Google search we note that when *network protection* query used in both proposed system and Google, the proposed system returns WebPages titled *network security, network confidence* in addition to *network protection,* but Google returns only WebPages titled *network protection* and *network security* only. Also if another query is used such as *image action in* Google only WebPages titled *image action* are returned, but the proposed system returns WebPages titled *image processing, image performance*, and *image action*. The same thing occurs with the other query. After all we must remember that both systems have their different environment.

6. Conclusions

By designing and implementing the proposed system, several concluding remarks have been drawn:

1. Improving user interface by applying a tool called guided search which helps the user to extract his query directly to prevent the wrong input (spelling mistakes) and save the time and keystrokes.
2. The result of query contains web pages yielding with synonyms of the query terms depending on database building in SQL server (semantic search).
3. The results are more accurate depending on the Recall and Precision measurements reaching 95% - 100% for some queries in a reasonable retrieved time.
4. Using full text indexing is better than partial indexing for search by increasing the probability of finding the query terms in web pages. But because of the processing of full text indexing of the web pages, longer time was taken.
5. The combination of the improved PageRank algorithm (Link-Based Rank) together with the proposed system reduces the number of iterations that are needed to reach the convergence; therefore, the time is reduced too.
6. Reducing the storage space particularly indices space by removing stop words from document when the index process is applied. This improved system performance.
7. Building many indices tables such as inverted index table, page information table, page form information table, link information table and hyperlink table depending on information which extracted not only from title and Meta tag but from all document contents. This increased the probability of finding the query terms in web pages.

7. Future Works

Several suggestions that could be implemented in the future to make this work more optimal are made:

1. This work is used to retrieve only .html and .htm files, so that retrieving other kinds of textual documents such as .pdf files, image text files such as .bmp and .jpg. And media files such as .wav would be expanded for proposed search engine.
2. Using neural networks to make guided search more efficient.
3. Building an advanced stemming algorithm to remove prefixes such as (rearrange) will become (arrange).
4. Building an intelligent spelling checker to correct the wrong words in the user query.
5. The index data can be encoded (compressed) to save on storage space.
6. Building a search engine system dealing with Arabic keywords.

ACKNOWLEDGMENTS

We would like to express our thanks to Dr. Isra'a T. Al-Attar for her useful and profound guidance to complete this research.

REFERENCES

[1] Croft, W. Bruce, Metzler, D., Strohman, T. "Search Engines Information Retrieval in Practice". Pearson Addision-Wesley. United States of America. 2010.

[2] Baeza-Yates, Ribeiro-Neto. "Modern Information Retrieval". Addison-Wesley-Longman Publishing co. 1999.

[3] Steven Garcia B.App.Sci. (Hons.). "Search Engine Optimisation Using Past Queries". School of Computer Science and Information Technology, Melbourne, Victoria, Australia. March 30, 2007.

[4] Lam, S. "The Overview of Web Search Engines". February 9, 2001. Available at: http://db.uwaterloo.cal~tozsu/courses/cs748t/surveys/sunny.pdf

[5] Boswell, W. "How Does a Search Engine Work? The Basic Inner Workings of Search Engines". Guide About.com. Available at: http://websearch.about.com/bio/ Wendy_Boswell_13134.htm, 2010.

[6] "How Web Search Engines Work". Last updated: February 17, 2006. QuinStreet Inc. 2010. Available at: http://www.webopedia.com/TERMS/S/How_Web_Search_Engines_Work.htm

[7] "How Search Engines Work". Incisive Interactive Marketing LLC. 2010. Available at: http://searchenginewatch.com/How_Search_Engines_Work-Search_Engine_Watch_(SEW).htm .

[8] McCown, F. "Introduction to Web Search Engines". 2009. Available at: http://knol.google.com/k/ introduction-to-web-search-engines#web_crawling

[9] Nancy Blachman, and Jerry Peek. "How Google Works". Last modified on: Friday February 2, 2007. Available at:

http://www.googleguide.com/How_Google_Works-Google Guide.htm .

[10] Fadhil, M. Natiq. “A proposed system for Semantic-Based Internet Search Engine”. M.sc. Thesis. University of Technology. Computer sciences. September 2004.

[11] Wordiq. ”synonyms”. Available at: http://www.wordiq.com/definition/Synonym

[12] Abhishek. “Top 10 Weird Suggestions From Google That Are Really Amusing”. July 19, 2009. Available at: http://www.gtricks.com .

[13] Abhishek. “Let Google Suggests what you Search”. Gtricks Sitemap. 2011. Available at: http://www.gtricks.com .

[14] Abhishek. “How to Disable Google Query Dropdown Suggestions”. Gtricks Sitemap. September 7, 2009. Available at: http://www.gtricks.com/google-tricks/let-google-suggests-what-you-search .

[15] Al-Attar, Isra’a Tahseen. “Internet Search Engine Design”. M.sc. Thesis. University of Technology. Department of Computer sciences. January 2005.

[16] Al-Rawi, Salah S. and Al-Khateeb, Belal. “Comparison the Efficiency of Some Search Engine on Arabic Keywords and Roots”. International Journal of Computational Linguistics Research, Volume 1, Number 1, March 2010.

[17] Saruladha K., Dr. Aghila G., Raj Sajina. “A New Semantic Similarity Metric for Solving Sparse Data Problem in Ontology based Information Retrieval System”. IJCSI International Journal of Computer Science Issues, Vol. 7, Issue 3, No 11, May 2010.

[18] De Swart, Henriette. “Introduction to Natural Language Se mantics”. Powells, 2010.

[19] Al-Rawi Salah. and Al-Khateeb Belal., Specialized Search Engines for Arabic Language, Journal of Information Technology Review, Vol. 2, February 2011.