A Proposed Method for Documents Indexing

Alia Karim Abdul Hassan and Duaa Enteesha mhawi
Computer science Department - University of Technology - Baghdad.

Received 30 January 2016 ; Accepted 10 August 2016

Abstract

In this paper, a new method is proposed for documents indexing based on constructing two tables, namely, words-information table and pages-information table. These two tables used to represent the first step in information retrieval (which prepare the documents set (preprocessing)). In Information retrieval systems, tokenization is an integrals part whose prime objective is to identifying the tokens and their count. In this paper, can be proposed an effective tokenization approach, which is based on proposed new method called documents indexing and results shows that efficiency of proposed algorithm. Tokenization on documents helps to satisfy user's information need more precisely and reduced search sharply. Preprocessing of input document is an integral part of Tokenization, which involves preprocessing of documents and generates its respective tokens, which is the basis of these tokens. Probabilistic IR generate its scoring and gives reduced search space. Comparative analysis based on the two parameters; reduce the time of search space, Pre-processing time, and reduce the size of memory.

Keywords: information retrieval (IR), Dataset (DS)

طريقة مقترحة لفهرسة الوثائق

علياء كريم عبد الحسن و دعاء نتيشة مهاوي

قسم علوم الحاسوب الجامعه التكنولوجيه\بغداد

الملخص

في هذه الورقة البحثية تم اقتراح طريقة جديده لفهرسة الوثائق والتي تستخدم لتمثيل الخطوة الاولى في استرجاع المعلومات (التي تحضر مجموعه الوثائق (عمليات مسبقة))، في أنظمة استرجاع المعلومات، التقطيع هو جزء تكاملي الذي يحدد الهدف من الرمز وحسابه، في هذه الورقة البحثية نستطيع ان نقترح خطوات تقطيع كفؤه مبنيه على أساس اقتراح طريقه جديده تسمى فهرسة الوثائق والنتائج تبين تلك الكفاءة للطريقة المطورة. تقطيع الوثائق يساعد ليلبي حاجه مستخدم المعلومات ويقلل فضاء البحث. اعاده معالجة الوثائق الداخلة هي جزء تكاملي لعمليه التقطيع، التي تتطلب اعاده معالجه للوثائق وتوليد رمز مطلوب الذي يعتبر أساس لهذا التقطيع. احتمالية استرجاع المعلومات تولد بدرجه وتعطي تقليل لفضاء البحث. مقارنة التحليل مبنيه على عاملين: تقليل فضاء البحث والوقت في اعاده المعالجة وتقليل من حجم الذاكرة.

الكلمات المفتاحية: استرجاع المعلومات(IR)، والداتا سيت (DS).

Introduction

Amount of operational data has been increasing exponentially from past few decades, the expectations of data-user is changing proportionally as well. The data-user expects more deep, exact, and detailed results. Retrieval of relevant results is always affected by the pattern, how they are stored indexed. Various techniques designed to index the documents, which done on the tokens identified with in documents, new techniques by using inverted index. Information retrieval (IR) handles the representation, storage, organization, and access to information items [1]. In IR, one of the main problems is to determine which documents are relevant and which are not to the user's needs. In practice, this problem usually mentioned as a ranking problem, which aims to solve according to the degree of relevance (matching) between all documents and the query of user [1] [2].Which deals with information retrieval [3]. General structure of information retrieval as shown in figure 1.

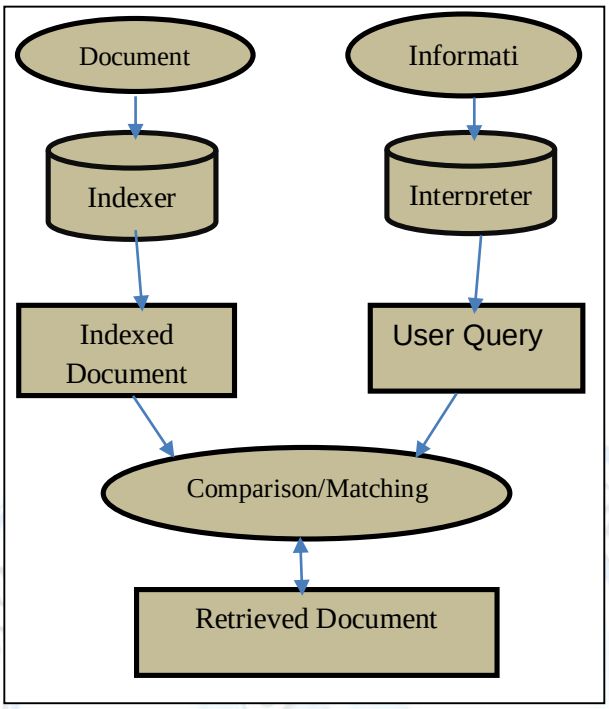

Figure 1: general structure of information retrieval

This paper proposed an algorithm for indexing webpage documents IR, which enhance the time processing and storage space.

Information Retrieval Process Model

The proposed system consists of two stages: the first stage is the preprocessing (prepare the dataset and store in database that will use as input to the second stage to retrieve relevant documents to the user query), in this stage can be proposed new method to index documents called proposed documents indexing algorithm. Illustrated in details in this paper.

Related work

Uematsu researcher used the inverted index in 2008, which used to store position of word, and document ID. Word position data is a list of offsets or positions in which the words occur in the document. Such occurrence information (i.e. Document ID and word position data) for each word is expressed as a list, called the "inverted list", and all the inverted lists taken together are referred to as the inverted index. In addition, A. Dallal in 2014 used EII, by using

for each word weight, frequency, number of unique word, total weight etc. but this method need large space of memory this size 100MB, and need time to store the results. So that, these models cannot be used.

Documents Indexing

This stage consists of two processes: (**Dataset reading** and **proposed documents indexing algorithm**). Figure 2 shows a block diagram describing the main processes of this stage.

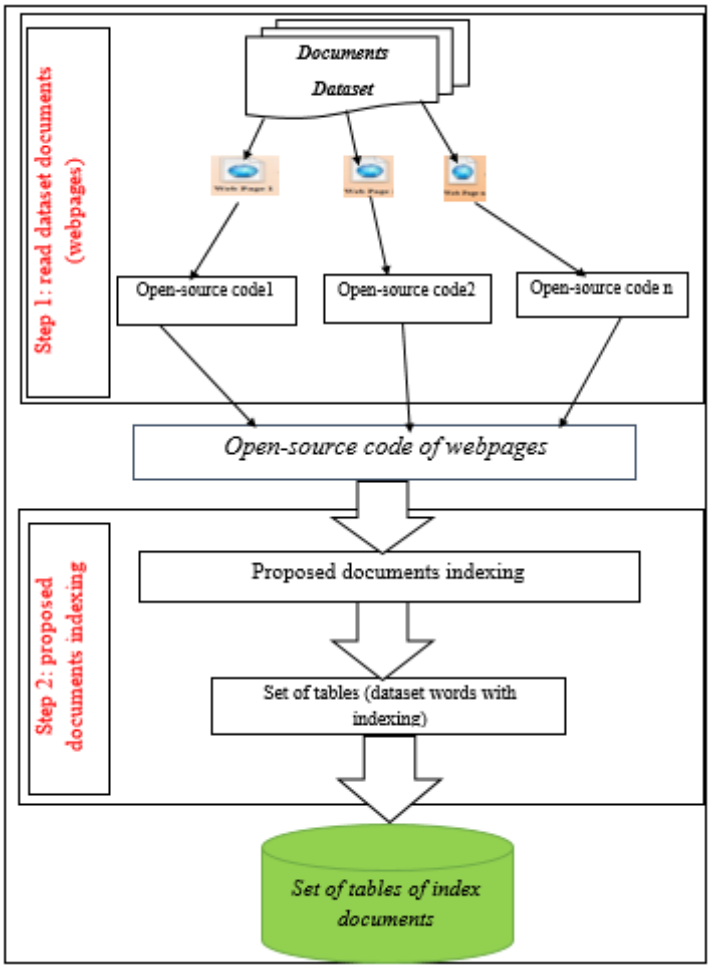

Figure 2: Block-Diagram of the proposed IRS

Dataset Reading

The proposed system implemented using a free dataset for simulation purpose. This dataset containing World Wide webpages gathered from computer science departments of various universities. Dataset consists of 8280 semi-structured documents, written in Hypertext Markup Language (HTML), webpages documents, which were manually classified into seven directories these directories, are department, students, staff, faculty, projects, courses and others see Table 1. Inside each directory five classes, each of which represents universities names, see Table 2. Table 3 shows documents tags. Each of these tags reflects a specific level of importance within the document, as well as these tags contains essential information near to the term of the user query, Figure 3 shows document before and after reading.

Table 1: Dataset containing set of directories

Directory name	No. of documents
Departments	181
Students	1641
Faculty	1124
Courses	930
Projects	504
Others	3763
Staffs	137
Total	**8280**

Table 2: Dataset directories contains

University name	No. of documents
Texas	**827**
Cornell	**867**
Washington	**1205**
Wisconsin	**1263**
Misc.	**4120**

Table 3: Dataset documents tags

Tag-name	*Weight*	*How many tag occur in page*	*Description*
title	6	appears only one time and not repeated	It is most important tags, because it contains terms near to the request of user query
header & sub-header (h1,h2,3)	5	appears n times	Words found in these tags provide information about the structure
A anchor	4	appears n times	This tag contains word this word points to another word or link
Italic (I)& bold (B)	3	appears n times	gives the descriptions of the document score
body	1	appears only one time and is not repeated	less important tag that contains the plain text

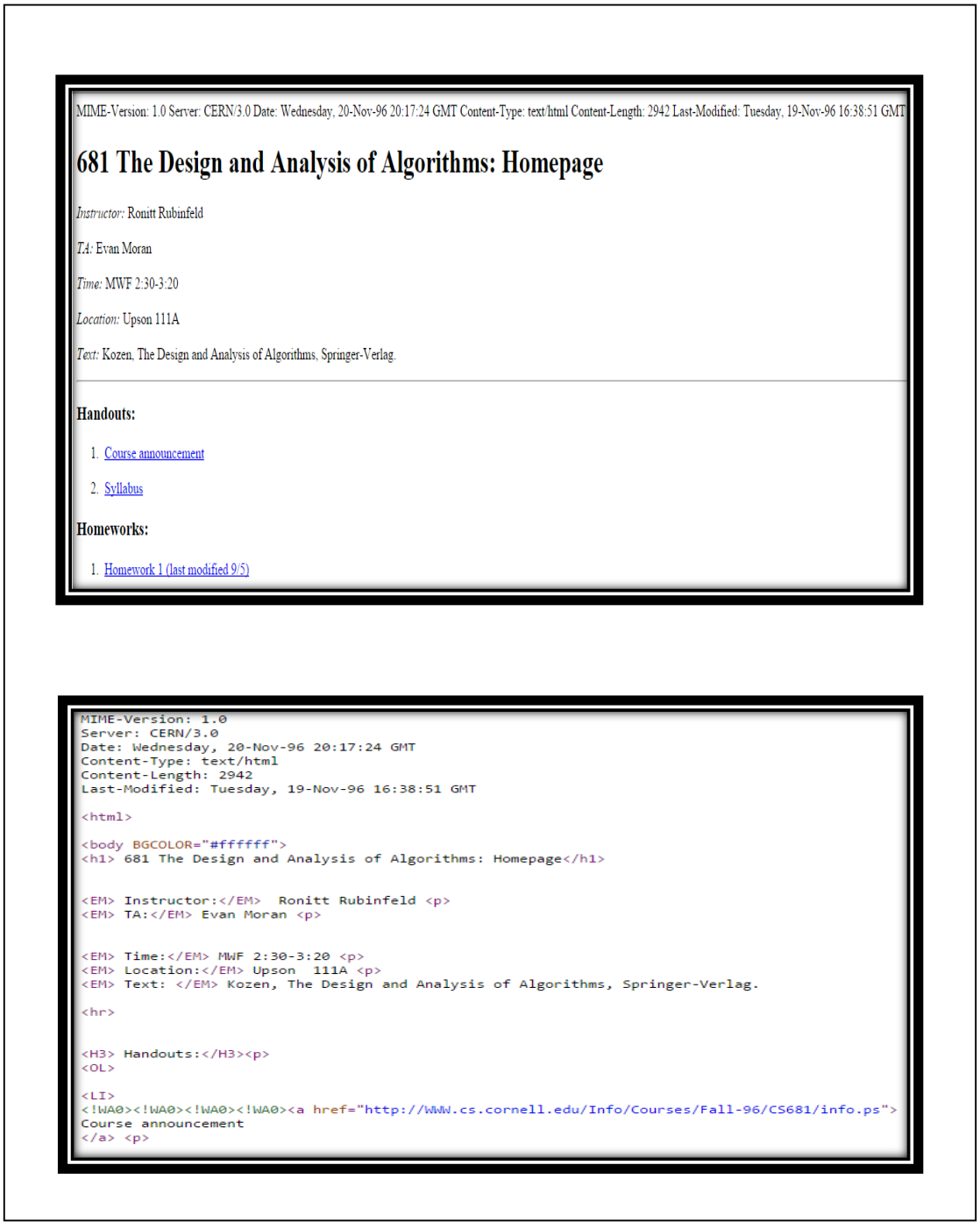

Figure 3:Document (webpage) befor and after reading

The Proposed Documents Indexing Algorithm

Search engines keep index of all available documents, by the use of documents index. In this paper, a proposed algorithm is used to index the documents called Proposed Documents Indexing. Algorithm 1 describes this proposed algorithm of indexing. The proposed algorithm reduces the storage space of memory required during the query processing. Documents indexing receives the source-code of each document, and does preprocessing to generate two tables. First table is called pages-information table, which contains information about each page in term (id, p-name, words-count, Total-Tags-Weight). The second table is called words-information table, which contains information about each word in term (word, pages-list).

Proposed Algorithm 1: Proposed documents indexing

Inputs: Open-source code of each document.

Outputs: Two tables: (pages-information table and words-information table)

Step 1: (preprocessing)

For each open-doc. (i) do

Begin

- Pages-information table.id=id (doc.(i)) \\ id represent (directory code , university code , page code.)
- Pages-information table. Page name= page name. (doc.(i))

While not EOF (open doc.(i)) do

Begin

Tokenization process **(Open doc.(i)) // extract the words (W)**

Get W

If W is stop-word or Special-character or sentence-Delimiter then

- remove (W)

Else Begin

- Pages-information table. Total-weight = total-weight (doc.(i)) \\ compute total weight from equation 3.1.
- Pages-information table. Total-count-word= summation (W)
- Words-information table. Word= W
- Words-information table. Pages-list= each page contain the same W

End if End while

-Store in pages-information table (id, p-name, total-weight, and total-count-word).

-Store in words-information table (word, pages-list)

End for

End

Figure 4, shows a block-diagram of the proposed documents indexing main steps.

The proposed documents indexing process begins with the following process:

1. **Special-word table construction by doing the following process:**

   a. Stop-word removing
   b. Special-characters removing
   c. Sentence-delimiters removing

2. Construction of Html tag information table by Remove unpreferred tags (html, head, sub-headers (h1, h2, h3), body), and generate weight of these tags remove.

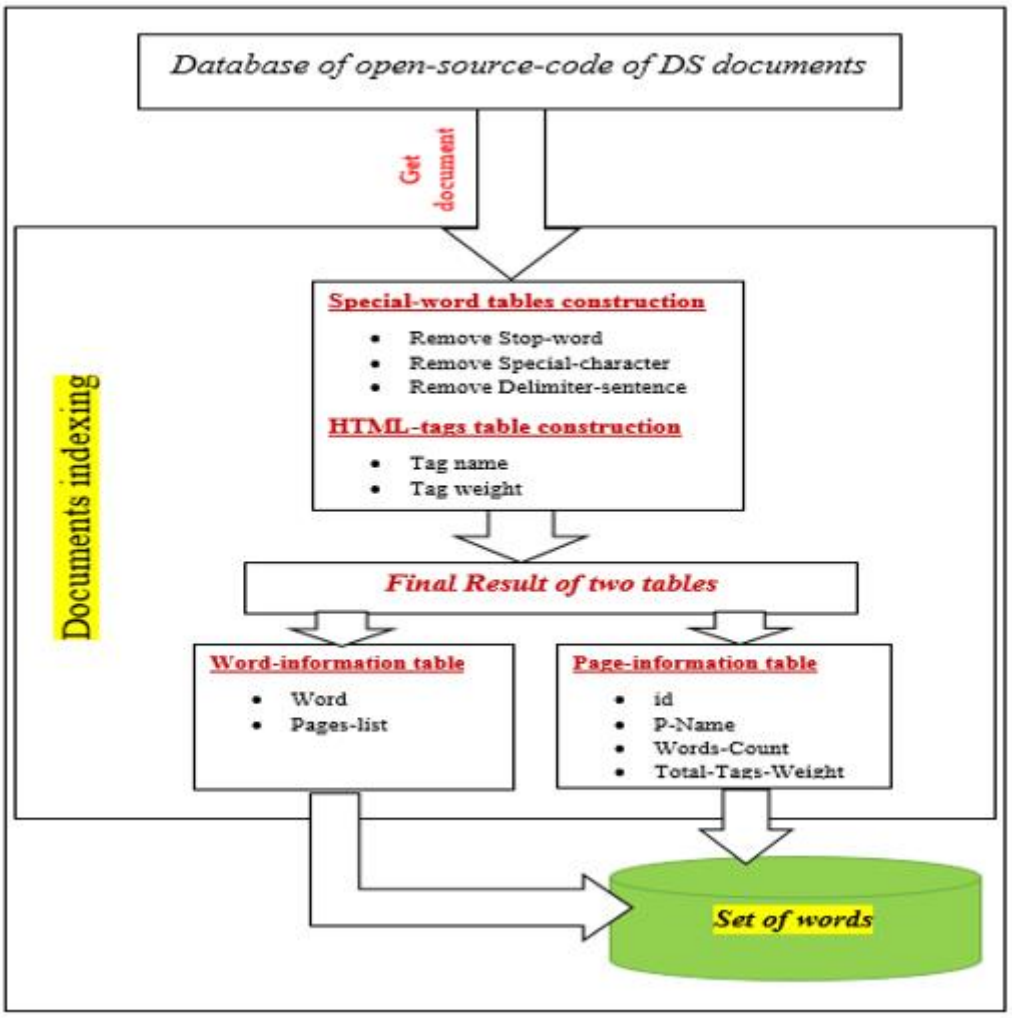

Figure 4: Block-diagram of documents indexing algorithm

After these preprocessing the indexing process, begins. Pages-information table generation is made by finding the total-tags-weight, using equation 1.

$$Total\text{-}tag\text{-}weight = Weight\ (W)\ tag + \sum Weight\ (W)\ tag \ \ldots\ 1$$

Where:

Weight (W) tag: weight of word by using weight of tag

Table 4 shows an example of pages-information table. The words-information table is constructed by first storing the word then finding the pages-list for this word, table 5 is an example of words-information table.

Table 4: Details and contents of pages-information table

Word	PageList
abac	3-1-52
abacom	3-1-52
abadi	0-1-549
abandah	2-1-885
abandon	2-1-273
abandoned	0-4-13 | 0-4-14 | 3-0-357 | 3-0-603 | 3-0-605
abb	3-1-52
abbadi	0-1-502 | 2-0-28 | 2-1-615 | 3-1-24
abbervations	0-1-6
abbey	2-1-273
abbotsford	3-1-52
abbott	0-2-31 | 2-1-483
abbreviate	3-0-258
abbreviated	2-1-69 | 2-1-786 | 2-1-810 | 2-1-912 | 2-3-24
abbreviations	3-0-243 | 3-1-23 | 3-1-41
abc	0-1-23 | 0-1-285 | 2-1-229 | 2-1-273 | 3-0-141
abcd	3-1-72 | 3-1-73
abcs	3-1-52
abdelsalam	0-1-260 | 0-1-261 | 0-1-262 | 0-1-263 | 0-1-264 | 0-1-265 | 2-1-31
abderrahim	2-1-287
abdol	2-1-150
abducted	2-3-2
abduction	2-1-754 | 3-0-147
abductive	2-1-709 | 2-1-754
abdul	2-1-179 | 3-0-497

Table 5: Details and contents of words-information table

ID	PName	WordsCount	TotalTagsWei
1-1-157	http_^^www.ida.liu.se^	118	128
0-1-100	http_^^pine.cs.yale.edu_4201^cs112^	289	129
2-2-1	http_^^www.cs.utexas.edu^users^boyer^	255	129
2-2-18	http_^^www.cs.utexas.edu^users^mooney^	205	129
0-1-575	http_^^www.cs.uoregon.edu^classes^cis210^	134	130
0-1-621	http_^^www.csc.ncsu.edu^eos^info^csc518_info^www^index.html	79	130
1-1-149	http_^^www.dcs.warwick.ac.uk^pub^	178	131
1-1-25	http_^^math.uwaterloo.ca^CS_Dept^homepage.html	127	131
2-1-807	http_^^www.cs.vassar.edu^faculty^welty^	248	131
2-2-20	http_^^www.cs.utexas.edu^users^novak^	175	132
2-0-14	http_^^www.cs.cornell.edu^Info^People^dean^dean.html	130	132
0-1-237	http_^^www.cis.upenn.edu^~lee^cse380.html	181	133
0-1-574	http_^^www.cs.uoregon.edu^classes^cis199www^	165	134
2-1-759	http_^^www.cs.unm.edu^~bap^	227	134
2-1-313	http_^^www.cs.arizona.edu^japan^www^rick.html	626	134
2-1-760	http_^^www.cs.unm.edu^~bederson^	119	134
0-1-511	http_^^www.cs.uidaho.edu^~foster^495s96^	335	135
0-1-616	http_^^www.csc.ncsu.edu^eos^info^csc417_info^www^index.html	206	135
0-1-2	http_^^ai.eecs.umich.edu^classes^498-2^	282	135
2-1-647	http_^^www.cs.uiuc.edu^CS_INFO_SERVER^DEPT_INFO^CS_FACULTY^FA	102	135
2-1-715	http_^^www.cs.umd.edu^~stewart^	118	136
2-1-614	http_^^www.cs.ucsb.edu^~agrawal^	243	136
0-1-303	http_^^www.cs.byu.edu^courses^cs345.ivie^345syll.html	261	137
2-1-126	http_^^netweb.usc.edu^danzig^	354	137
0-3-15	http_^^www.cs.washington.edu^education^courses^373^95a^index.html.95a	75	137
0-1-362	http_^^www.cs.duke.edu^~mlittman^courses^cps370^	1495	138

The proposed method ability to index all terms that is meaning, and then add information need for each term, these information stored in two tables, first table store words and pages-list of each word. Each page list consists of ID-list, also this ID consists of three parts (directory number, university number, and page number), this ID make the process of retrieve related document faster than the traditional. The second table especial of pages information, and this table store the name of page and total word count finally total weight of each page. This help of made up the proposed fitness function. So that the memory space of the developing smaller than traditional and ref [6] In figure 5 traditional need large space of memory, for each entry read document need 2-byte (2*8), and dataset used is 8280 documents. Can be required to multiple total words in dataset (67,672 words) with all document in dataset (8280 documents) with 16, the result is 8965,186,560 MB. While memory size need to store, the indexing in Ref. [6] is 100MB. While the proposed documents indexing in the first stage of this proposed system, only need 19.9MB to store data in memory. This reduce of memory size in proposed system done through remove each word outside tags (html, head, sub-header (h1, h2, and h3), and body).in addition remove each word that contains number together such as: operating565 or any word contains any of special character that is came together with this word can be removed. By using the proposed documents indexing in this paper the memory space became smaller than traditional and ref. [6]. Because this method work as follows: open the source code of each

webpage then determine, where the start tag of html begins or tag of title, head or body because some webpage cant begins with html tag, so that take this reason in account. Therefore each word out of this tag can be removed also can be remove special word and delimiter sentence but cannot applied the stemming process. Then entered to each tag and put the weight for each word, specific to the degree of a tag contains these words. To give this webpage value when the keywords of query found in this page to spent on two problems of information retrieval. Then can be account the total words weight and account, finally for each word applied the principle of pages-list, used in the evaluation function (fitness function). The principle of pages list consider the essential in the IR systems.

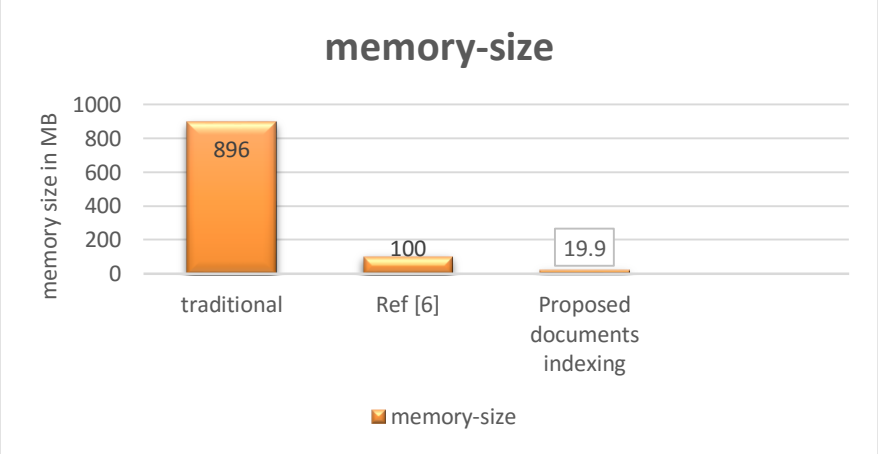

Figure 5: Memory space for document indexing

Conclusion

In this paper, proposed method of indexing documents can be used for indexing webpage documents information retrieval. Simulation results of the proposed algorithm then compared with same traditional algorithm and ref. [6]. Approved the efficiency of the proposed algorithm in term of storage space and processing time.

Implementation

Proposed algorithm executed with 8280 documents (dataset).

References

1. Christopher D. ,Prabhakar R. and Hinrich S. , "An Introduction To Information Retrieval", Book , Cambridge University Press , February 16, 2008.
2. Cristina L., Vicente P., Felix d., "Genetic algorithms in relevance feedback: a second test and new contributions", Proceedings in Information Processing and Management 39, 2003.
3. Al Siva K., Dr. P. Premchand, Dr. A. Govardhan, "Query-Based Summarizer Based on Similarity of Sentences and Word Frequency", International Journal of Data Mining & Knowledge Management Process (IJDKP) Vol.1, No.3, May 2011.
4. Kim, S., and Zhang, B-T. (2003). Genetic mining of html structures for effective web document retrieval. *Applied Intelligence*, vol.18, no.3, pp.243-256.
5. The 4 Universities Data Set. (1998). [online]. Available at: http://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-20/www/data/[Accessed 12/11/2009].
6. Amar. Al-D., apply, ***"enhancing recall and precision of web search by using genetic algorithm"***, 2013, pp. 343 - 348.