Enhance Inverted Index Using in Information Retrieval

Dr. Alia Karim Hassan
Computer science Department, University of Technology/Baghdad.
Duaa Enteesha mhawi
Computer science Department, University of Technology/Baghdad.
Email: dododuaaenteesha@yahoo.com

Received on:26/5/2015 & Accepted on: 20/1/2016

ABSTRACT:

This paper proposes a method to represent the first step in information retrieval (IR) (that prepare the document set (preprocessing), In Information retrieval systems, tokenization is an integral part whose prime objective is to identify the token and their count. In this paper, an effective tokenization approach which is based on proposed new method called enhance inverted index (EII). The result shows that efficiency/ effectiveness of the proposed algorithm. Tokenization on documents helps to satisfy user's information need more precisely and reduced search sharply, believed to be a part of information retrieval. Pre-processing 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 generates its scoring and gives reduced search space. The comparative analysis based on the two parameters; reduce the time of search space, Pre-processing time.

Keywords: information retrieval (IR), enhance inverted index (EII).

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. There are various techniques are designed to index the documents, which is done on the token's identified with in documents, new techniques by using inverted index.[1]

Information retrieval (IR) handles the representation, storage, organization, and access to information items. 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] [3].Which deals with information retrieval. General structure of information retrieval is as shown in Figure (1).

https://doi.org/10.30684/etj.34.2B.13
2412-0758/University of Technology-Iraq, Baghdad, Iraq

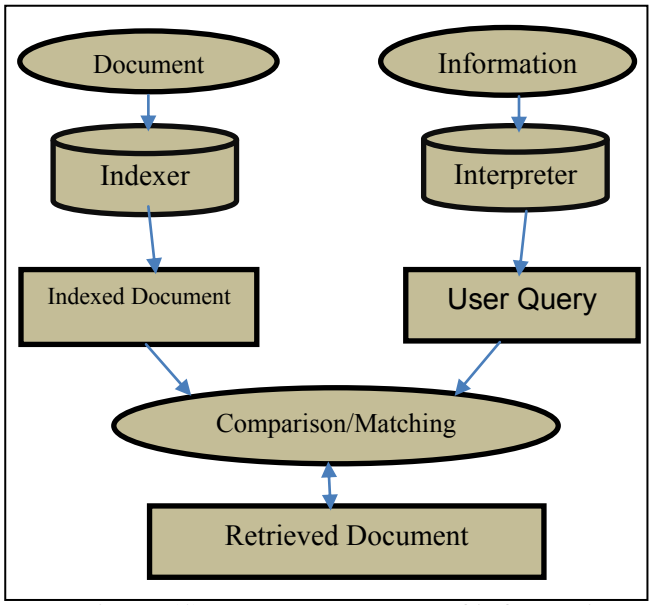


Figure (1): general structure of information retrieval

In this paper, an algorithm for web document IR was proposed that enhances the time processing and storage space.

Information retrieval process model

The proposed system consist of two stages: the first stage is the preprocessing (that prepared the data set and store it in a database that will be used in the second stage), in this stage, a new method of preprocessing and parsing that documents were proposed this method called Enhance Inverted Index (EII), that are illustrated in the details in this paper. The second stage is to use an algorithm of machine learning that algorithm.

Describe of Dataset

It is contains World Wide Webpages that gathered from various universities of computer science departments in the 1997, January by the worldwide knowledgebase (Web->KB) project of Carnegie Mellon University group of text learning. Type of this document (Web->Kb) is semi-structure that mean it is HTML (Hypertext Markup Language)) documents. Data set consists of 8284 web documents (pages) that is manually classifies into seven directory these directory illustrate in table 1, named :department, students, staff, faculty, projects and courses as well as to the 60 other random web pages that downloaded from other universities ;therefore, the total number of the data set is 8280 documents [5].

Table (1): directory of the data set

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

Total	**8280**

Table 1 that represents seven directories, inside each directory, four classes each of which represents university that names:

1. Texas contains 827 documents
2. Cornell contains 867 documents
3. Washington contains 1205 documents
4. Wisconsin contains 1263 documents

The remaining 4120 collected from other universities and assigned to the field called Miscellaneous. This data set used by different researchers such as (Amar al Dallas, 2014; Dong et al, 2008; Craven et al, 1998) and others. This data set is to be used in the first stage of the proposed system. It should be converted to the text by using new method proposed called (EII). It will be explained in details in the next section and store results in Database (DB) by using Microsoft Office access 2013.

Documents of data set written in HTML language and this format exhibits different feature as following:

1. HTML (Hypertext Markup Language) consists of tags, and these tags can be used to calculate weighting of each terms (Shih ,Cutler and Ming, 1997) as will be illustrated in the section of the EII.
2. (Zhang and Kim, 2003; Cutler et, al, 1999) using tags to assign weight of each term because each tag is assign a special weight therefore; it will distinguish special characters from the normal text within the document.
3. Documents that written in HTML are better descriptive content documents (Cutler, Shih and Ming, 1997).
4. Most format of documents web written in HTML (Zhang and Kim, 2003) so that IR is application of the web mining.
5. HTML tags and the layout are better reflecting terms inside documents (Cutler et al, 1999).

Related Work

Uematsu researcher used the inverted index in 2008 that index is a structure used to store word position data, as well as 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 as the inverted index. . Kuia and Juan in 2012who applied an improved version of term frequency – inverse term frequency (TF-IDF).

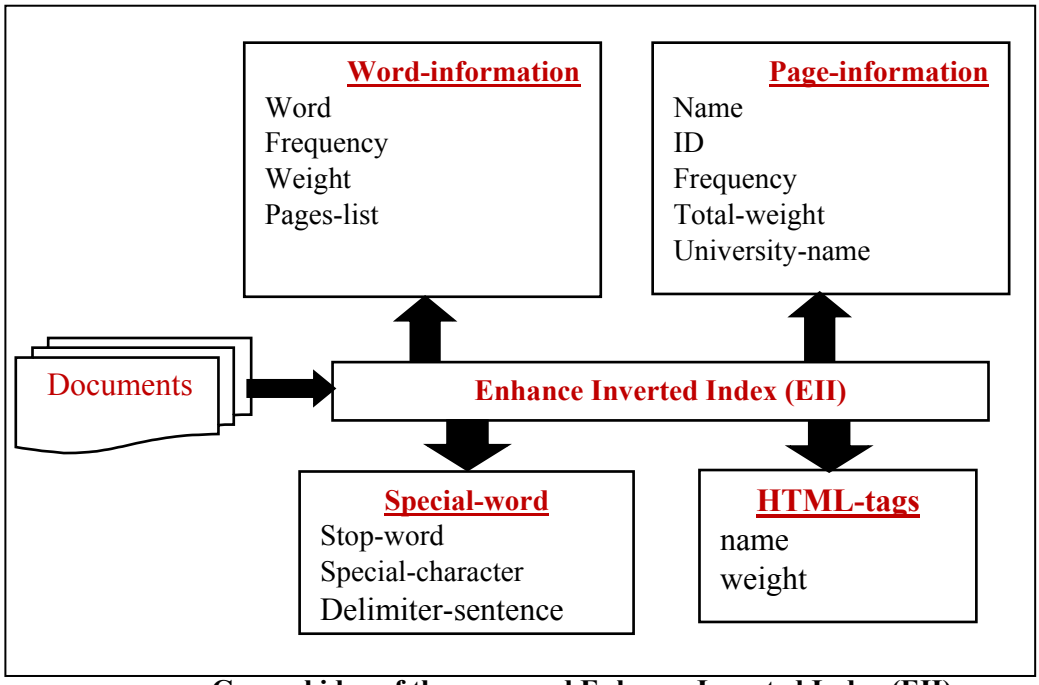
These models may not use any more in the areas of IR for the following problems:

1. That index needs a large amount of space to store it.
2. Need a long time to retrieve the specific keyword.
3. All documents should be checked against each query.
4. That index stores a few details about the information in the documents (data set) (Al Dallas and Abdul Wahhabi, 2011), and this is due to the large amount of space required to include addition data per keyword.

An enhanced inverted index will be presented in this work to overcome the above problems.

Proposed the Enhance Inverted Index (EII)

The Dataset (documents) represents as a webpage that should be converted to the text and stored in Microsoft Office Access in two tables. Those tables' names: pages-information and words-information, the first table consists of information about each page that is (id, name, frequency, university-name), while the word-information table contains information about each word such as: (word, frequency, weight, pages-list), the general idea of the proposed Enhance Inverted Index (EII) illustrate in Figure (2).



General idea of the proposed Enhance Inverted Index (EII)

While **HTML-tags** table contain a tag and corresponding weights, **special-word** this table include three main tables:

- **Stop-word,** which includes list of stop words [4] such as "the, as, of, and, or, to etc. that should be removed from the document this step is very essential because it has some advantages: It reduces the size of indexing file and it also improves the overall efficiency and make effectiveness.

- **Special-characters table** this process removed set of characters, these characters: semi-colon, colon, exclamation, etc.
- **Sentence-delimiters table** a subset of table above those will be used to separate the statement from other statements. These delimiters should be discarded from the document.

All these tables and documents (data set) above represents as inputs to the **EII**, the outputs are two tables that stored in database and considered as input to the second stage.

Table (2): tags and corresponding weighting

Name of tag	Weighting of tag
Title	7
HEAD,H1,H2,H3	6
A:anchor	5
B:bold;I:Italic	4
Body	2

Algorithm1: Enhance Inverted Index (EII)
Input: documents (data set)
Output: two tables: word-information and page-information
Step1: get page (P) from data set then open the source code of it.
Step 2: (preprocessing)
Initialization: Weight: = 0
From each P get, source-code1 (bring source code of webpage and perform tokenization process from each word (W)) then check each (W) as following:
While not EOF do
Get (W)
If W in the open tag then
Weight: =weight+ weighting of tag
Else
if W not stop-word then
Weight: =weight
Calculate frequency W (how many W repeated for all the webpage)
Calculate pages-list W (bring all pages that contain the same W)
Then store these results (weight, frequency, and pages-list) in the table of the word-information.
Step3 Get source-code2 of the same webpage in the step2, remove all tags in the page P and bring frequency W from word-information table then calculate of each P the following**:**
ID consist of three parts (university-name, directory, page-name)
Name of page
Frequency W
Total weight
University-name
Then store in the pages-information table (id, name, frequency, total-weight and university-name)
Step4: end

To describe Algorithm 1 each step.

1. Get page from documents (dataset) WEBKB.

This process is to bring each page from the dataset.

2. Parser(preprocessing):

Get page_source1, Get page_source2.
That means convert page to the source code written in HTML to easy treatment with this page and take important information from it during the tokenization process, then apply some preprocessing on it and store it in a database to use it in retrieval. Figure (3) explain the source code of the web
Page.

```
<! DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xylenes="http://www.w3.org/1999/xhtml">

<Head>

<Meta http-equip="Content-Type" content="text/html; charset=iso-8859-1" />

<Title>How to add/Update/Delete Record in Access Database</title>

</head>

<Body>

<p>this source code has a tutorial at my website for the systematic explanation
on how to connect to a database and make changes like add/update/delete.</p>

<p>the primary purpose of this code is to teach beginner programmer to
familiarize the concept of database programming. </p>

<p>this is particularly for beginner but may also applicable for intermediate
programmer. </p>

<p>If you have questions don't hesitate to visit my website at <a
href="http://www.sourcecodester.com">http://www.sourcecodester.com</a>.<br
/>
```

Figure (3): the source code of the webpage

That preprocessing applied to each page source1 and page source2 but with few different in processing in two pages.

Applied stop word removal and remove special delimiter from two, but in the page_source1 must remove each tag, and this process cannot be applied in page source2 such as:(body , title , head, h1 , h2, h3 ,I ,b) ,show the next example explain these steps, that page written in HTML , must remove tags ,stop words and delimiter from it in the step of the source_page1.

Next, show this process and the result of the preprocessing as following in figure (4).

MIME Version 1 0
Server CERN 3 0
Date Wednesday 20 Nov 96 20 17 24 GMT
Content Type text html
Content Length 2942
Last Modified Tuesday 19 Nov 96 16 38 51 GMT

681 Design Analysis Algorithms Homepage Instructor Reknit Rubin Feld TA Evan Moran Time MWF 2 30 3 20 Location Upson 111A Text Cozen Design Analysis Algorithms Springer Verilog Handouts Course announcement Syllabus Homework's Homework 1 (last modified 9 5) Homework 2 (last modified 9 11) Homework 3 (last modified 9 22) Homework 4 (last modified 9 27) ***see addendum* (last modified 10 2) Homework 5 (last modified 10 11) ***see addendum*** (last modified 10 18) Homework 6 ***don't see addendum see copy HW*** Homework 7 (last modified 11 6) Homework 8 (last modified 11 13) Solutions Solution 1 Solution 2 Solution 3 Solution 4 Solution 5 Solution 6 Solution 7 Announcements There exam Thursday Nov 21 7 in Upson 111 111A Talk to me Evan reschedule make time You refer Kohen text 8 5x11" cheat sheet class notes home works Rajeev Motswana's lecture notes approximations**

Figure (4): preprocessing web page

Then token page into words, make each page with unique ID that consists of three parts explain in next item.

A. Page_source1 computed word count (frequency) of each word that will be used in the page_source2 to compute weight.

B. In the page_source2, that tags stayed not removed, because can be used to compute the weight of each word then, add it to the table of the word_information.

C. check if that word is in the table of the word or not:

- If not exist then add it to the word table as following (word, frequency (word count), weight, pages list).
- If found, can be updated of the (frequency (word count), weight, pages list).

3. the last step is to retrieve the information of the page through the ID,

that retrieval is fast so that ID reduce the time of the search space, because, that ID consists of three parts (university, directory , page-sequence).

When user need information from the specific university cannot need to search all datasets and directory to find the specific information, through the ID can easily find that university that search about it, ID reduces the time of the search space and efficient to find the required information.

Implementation

The proposed algorithm EII, implemented by using Visual Basic.net 2013 using the dataset in the access 2013 to store 8242 document as web page.

The proposed algorithm executed with 100 documents and the result is shown in Figure (5) and Figure (6) that compare the results of the preprocessing between the traditional algorithm and by using the new method that proposed in this paper.

Performance evaluation

(1) **Strategy**: There are two alternatives of strategy, tokenization with preprocessing and tokenization without preprocessing. Preprocessing involves acquired the specific information details from the webpage and store these information in two tables in the database. These tables are word information and page information. The tokenization with pre-processing generates more accurate and effective tokens with more efficient manner, while without pre-processing strategy simply parses input documents and generates tokens.

(2) **Overall-Time Value**: Time consumed in entire tokenization process is directly proportional to performance measure of an IR system, as it deeply affects the Indexing & storage aspects.

Simulation

In tokenization with preprocessing 200 numbers of tokens generated while for same set of input documents, another strategy (without preprocessing) generates more than 300 tokens. The more is the number of token generated, the bigger is the challenge to manage them into storage space & effect in the accuracy of results retrieval.

Simulation result of the proposed algorithm and standard algorithm (inverted index) compared in the number of generated and processing time.

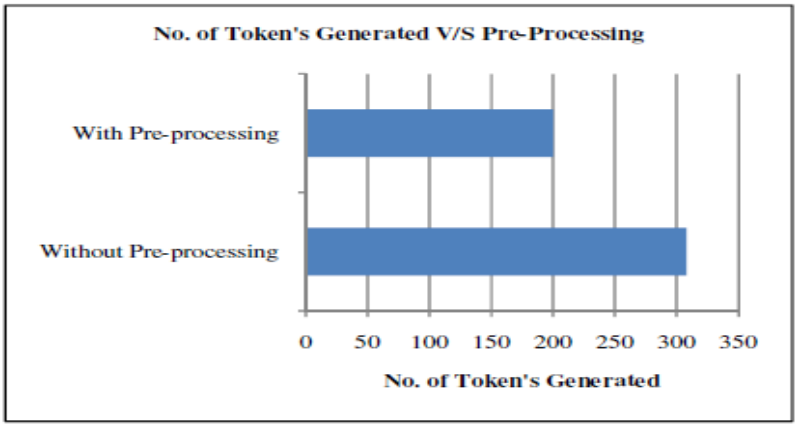


Figure (5): Document Tokenization Graph

Another result graph is illustrated in Figure (6); overall time consumed by the strategy is an important factor, which affects overall efficiency of an IR system. The Tokenization with Preprocessing using (EII) leads to effective and efficient approach of processing, as shown in results strategy with preprocessing process 100 input documents and generate 200 distinct and accurate tokens in 156 (ms), while processing same set of documents in another strategy takes 289 (ms) and generates more than 300 tokens.

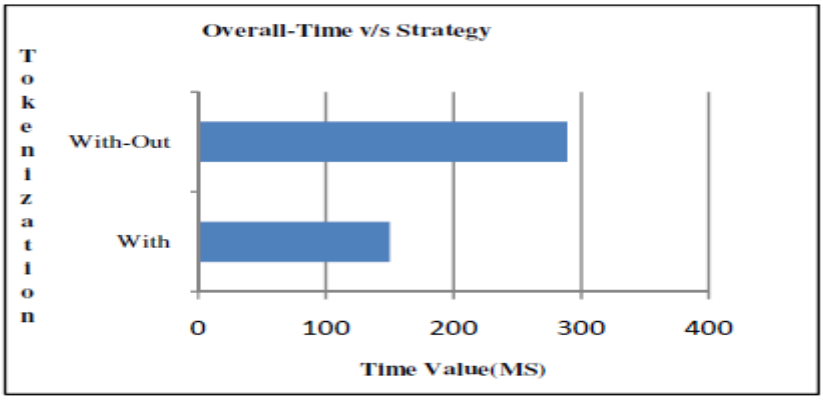


Figure (6): Overall-Time Graph

CONCLUSION

In this paper, EII algorithm proposed for web document information retrieval.
Simulation result to the proposed algorithm then compared with same traditional algorithm of the inverted index approved the efficiency of the proposed algorithm in term of storage space and processing time.

REFERENCES

[1] Christopher D. Manning, Prabhakar Raghavan and Hinrich Schütze, "An Introduction to Information Retrieval", Book, Cambridge University Press, February 16, 2008.
[2] Cristina Lopez-Pujalte, Vicente P. Guerrero-Bote, Felix de Moya-Anegon, "Genetic algorithms in relevance feedback: a second test and new contributions", Proceedings in Information Processing and Management 39, 2003.
[3] A. P. Siva Kumar, 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 four 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].