AN EVALUATION AND OVERVIEW OF INDICES BASED ON ARABIC DOCUMENTS

Hanandeh

This research is funded by the Deanship of Research and Graduate Studies in Zarqa University Jordan

ABSTRACT

The paper aims at giving an overview about inverted files , signature files, suffix array and suffix tree based on Arabic documents collection. The paper also aims at giving the comparison points between all these techniques and the performance of this techniques on each of the comparison points. Any information retrieval System is usually evaluated through efficiency and effectiveness of this system. Moreover, there are two aspects of efficiency: Time and Space. The time measure represents the time needed to retrieve a document relevant to a specified query, while space represents the capacity of memory needed to create the two indices.

In this paper, four indices will be built: inverted-file , signature-file, suffix array and suffix tree. However, to measure the performance of each one, a retrieval system must be built to compare the results of using these indices.

A collection of 242 Arabic Abstracts from the proceeding of the Saudi Arabian National Computer Conferences have been used in these systems, and a collection of 60 Arabic queries have been run on the there systems. We found out that the retrieval result for inverted files is better than the retrieval result for other indices.

KEY WORDS:

Information retrieval, Arabic document, Indices, Recall, Precision

1. INTRODUCTION

Information Retrieval (IR), refers to the processing of user requests, commonly referred to as queries to obtain relevant information from collection of documents [6]. Due to historical reasons, documents in a collection are frequently represented through a set of indexing terms or keywords; these keywords could be extracted directly from the text of the document or might be specified by human specialist. To increase the search speed among index terms, several data structures are proposed to store these terms (called indices) [1].

The most commonly used structures for information retrieval can be classified as lexicographical indices (indices that are sorted), and indices based on hashing. One type of lexicographical index is the inverted file. On the other hand, an example of hashing index is the signature file [2].

DOI : 10.5121/ijcsea.2014.4402

We used here tow mechanisms in inverted file, the first is word-oriented mechanism for indexing a text collection and the second is block-oriented mechanism in order to see what is better for searching, while signature-files are word-oriented index structures based-on hashing [1]. The suffix array and suffix tree are word-oriented index structures .We built all indices one time by root and the second by full word.

An inverted file index consists of a record, or inverted list for each term that appears in the document. A term's record contains an entry for every occurrence of the term in the document collection identifies the documents and, possibly, gives the location of the occurrences or a weight associated with the occurrences [7]. The set of all different terms in the document is referred to as (the vocabulary) [1].

A signature file uses hash function (or 'signature') that maps words to bit masks (signatures) [1]. The most common way for generating signatures from a text is the Super-imposed coding. In super-imposed coding, a text is divided into text blocks containing the same number of unique, non trivial words. Each word in a text block is hashed into word signature. A block signature is generated by super-imposing all word signatures generated from the block [8].

A qualitative comparison of the signature-base method versus inverted method is as follows: signature-based method requires less space overhead (typically 10-20 %)[1], as apposed to (50-300%)[2] that inversion requires, on the other hand, inverted mechanism has the advantage of better search speed than signature file mechanism.

2. GENERAL PROBLEM

Information retrieval deals with representation, storage, organizational and access to information items. Information retrieval can be problematic for some factors. We have huge amount of data, the desire to keep related items together; documents have unstructured domain, data. The information is distributed and interlinked without forgetting the performance criteria.

As the volume of data and documents have reached a limit which is statistically unsuitability, and as the search processes, whether in the Internet or in the libraries, have become more complicated, it is necessary to improve the search process.

The efforts was done by researchers in other natural languages such as English language. but the efforts is still limited in our language, I mean the Arabic language and not satisfy our ambition . This was the motivation behind this research hoping this work will enhance information retrieval system degree.

3. OBJECTIVES OF THE STUDY

We summarize the objectives as follows:

1. Investigating and evaluating different indexing method such as Inverted file, Signature file, Suffix tree and suffix array to carry out the candidate indexing term.
2. Applying the work on Arabic information retrieval system using Arabic Root and Arabic full word as index terms.
3. Evaluating the retrieval effectiveness using the recall, precision measures factors.
4. Comparing between Different indexing method based on Arabic Documents

4. EXPERIMENT AND TESTING

To compare the four indices structures, inverted file, signature file, suffix array and suffix tree we run our system over 242 Arabic documents and pre-defined 60 Arabic natural language queries. However, the comparison will be based totally on search-speed, storage overhead and average recall, precision.

Our system was implemented in C# NET language, and Runs on IBM/PCs and compatible microcomputer. However, running the system each time with different structure will allow us to measure the search-speed for each one. By search-speed, we mean the time needed to retrieve relevant documents associated with a specified query.

After we have built the files, the storage space of each one has been identified and, at this point, we can compare between them to determine which mechanism requires less storage overhead. To measure the efficiency of retrieval, a retrieval evaluation must be used. Recall, precision evaluation measurement are used to determine the goodness of retrieved documents. In other words, high recall means high number of relevant documents retrieved while high precision means high number of relevant retrieved documents, where the optimal is to have a system with high recall and high precision .

To compare the efficiency of the all structures, we have computed the average recall and precision over the 60 queries, the steps followed to compute average recall and precision are:

1. Defining manually the relevant documents for each query.
2. defining a thresh-hold
3. for i = 1 to number of quires

Retrieve all documents that have similarity greater than the thresh-hold Compute the precision at the 11 standard recall level (0.1,0.2,…..,1)

End

4. for r = 0 to 10

$$\overline{P}(r) = \sum_{i=1}^{Nq} P_i(r) /Nq\ldots\ldots\ldots\ldots\ldots\ldots..[1]$$

Results

Figure 1 illustrates the space requirement of the original documents for files (inverted, signature, suffix array, suffix tree) relative to the number of documents in the database.

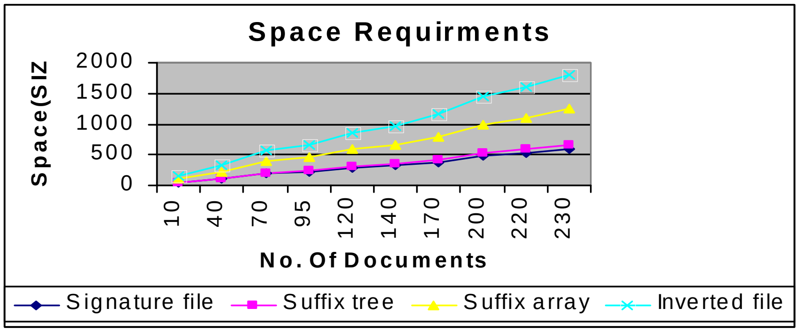

Figure 1. space requirement

Figure 2 illustrates the search time of files (inverted, signature, suffix array, suffix tree) relative to the number of documents in the database. The search time represents the time taken to run the 60 queries in the system.

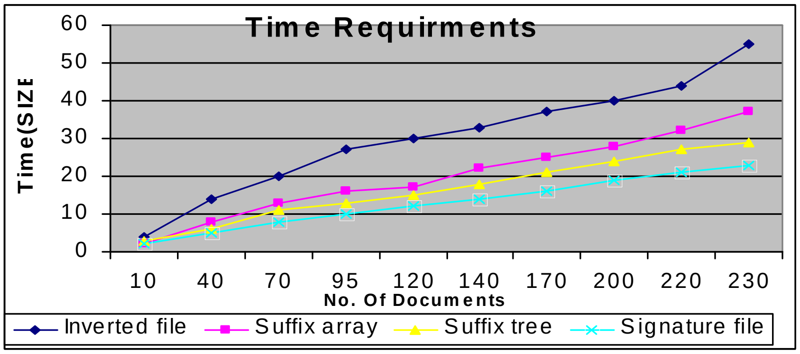

Figure2. Time requirement

Figure 3. Illustrates the Average Recall Precision of files (inverted, signature, suffix array, suffix tree) for the 60 query

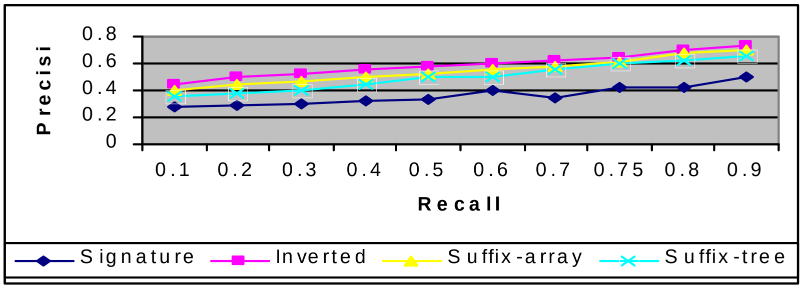

Figure 3. Average Recall Precision for the 60 queries

Figure 4 illustrates the average recall/precision for the 60 queries when using the all files after minimizing the false drops for the signature file.

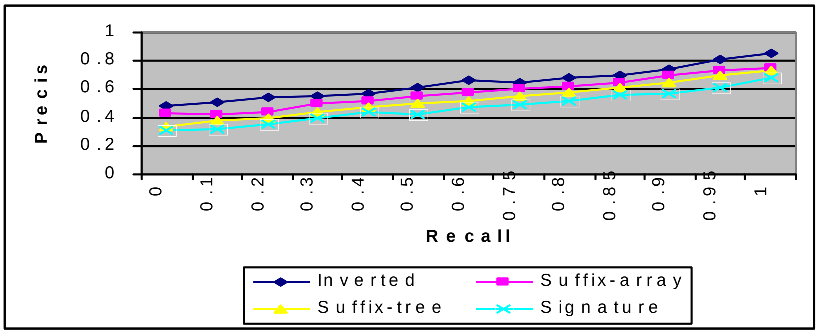

Figure 4. Average Recall Precision for the 60 queries after minimizing the false drops

5. CONCLUSION

After running the system and getting the results, a conclusion can be made about the four indexing mechanisms. However, the comparison between such mechanisms is based on three factors: Time, Space, and average recall/precision evaluation.

From figure 1 we can conclude that signature file is much smaller than the original size of the collection. It takes around 16% from the original collection size like almost suffix tree while inverted file takes around 173% from the original collection size but the suffix array is take around 154% from the original collection size. Baeza-Yates and Ribeiro-Neto [1] refer in their book that the signature needs around (10-20%) space while Faloutsos [2] refers that inverted file needs space from 50-300% of the original size.

Figure 2 illustrates that the search time needed by inverted file is less than the time needed by other files. Since the signature file has to be searched exhaustively if signatures are organized in a single sequential file [8]. Donna Harman et al [2], refer that the use of inverted file will improve search efficiency by several orders of magnitude. Figure 3, we conclude that inverted file has better average recall/precision than signature file, suffix tree and array tree,. Figure 4 shows that the average recall/precision for the inverted file is better from the average recall/precision for the signature file but the difference between inverted and signature less than the Figure 3, this happens due to the effects of the false drops in the signature file [4]. Lastly, Figure 4 shows that when reducing the number of words per block in the signature file, the average recall/precision for the signature file has improved. This is due to Equation 1 [4].

$p=(1-(1-w/m)^{s})^{w}$ ………………………………. Equation 1[4]

Where

p : false drops probability

w: number of bit set by a word

m: signature length

s : number of words hashed into a block signature

REFERENCES

[1] Kanaan,G., Comparing Automatic Statistical and Syntactic Phrase Indexing for Arabic Information Retrieval, PhD. Thesis, University of Illinois ,Chicago USA,1997

[2] Baeza-Yates, R., and Ribeiro-Neto, B. (1999). "Modern Information Retrieval". Addison Wesley

[3] Darwish,K., Building a Shallow Arabic Morphological Analyzer in one Day, Acl Workshop on Computational Approaches to Semitic Language,2002,PP.47-57.

[4] Salton, G.,Automatic Text Processing: the Translation Analysis and Retrieval of Information by Computer, Addison-wesley publishing, USA,1988.

[5] Gname, Mohm'd, The Computing Systems for retrieval by Natural Language, Ph.D. Thesis, University of Cairo, Egypt, 2003

[6] Khaib, Ahmad. the specification term and their application in Arabic language.Amman,1995. the fifth teen season for Jordanian Arabic language meeting.P 177-213.

[7] Al-Yaseen, M., Kanaan, G., Al-Shalabi R., Kanaan T. and Bsoul A (2007). Applying Partition Around Medoids like Clustering Algorithm for Enhancing Retrieval Processes in Arabic Text using different Similarity Measures. Proceedings of the Information and Communication Technologies International Symposium ICTIS'07. fez-Morocco.

[8] Kanaan, G., Al-Shalabi R., Bsoul A., and Kanaan T., (2007). Arabic-English Cross-Language Information Retrieval using Latent Semantic Indexing with Singular-Value Matrix Decomposition. Proceedings of the Information and Communication Technologies International Symposium ICTIS'07. fez-Morocco.

[9] Kanaan, G. Riyad, Al-Shalabi, Abd-Allah Al-Akhras, (2006). KNN Arabic Text Categorization Using IG Feature Selection, The 4th International Multiconference on Computer and Information Technology, CSIT 2006, Amman, Jordan.

[10] Riyad Al-Shalabi, Ghassan Kanaan, and Manaf H. Gharaibeh, (2006). Arabic Text Categorization Using KNN Algorithm, The 4th International Multiconference on Computer and Information Technology, CSIT 2006, Amman, Jordan.

[11] Al-Shalabi R., Kanaan G., Jahjah L. (2005). Question Answering Arabic System based on Passage Selection. Proceedings of the 5th International Business Information Management Conference (IBIMA). Cairo, Egypt.

[12] Al-Shalabi, R., Kanaan, G., and Muaidi, H. (2003). New Approach for Extracting Arabic Roots. Proceeding of the International Arab Conference on Information Technology. Alexandria, Egypt.

[13] Ghwanmeh S., Kanaan G., Al-Shalabi R., Gharaibeh J. and Samara S. (2005). Comparison between Inverted and Signature Files based on Arabic Documents, International Journal of Applied Science and Computations, , 12(3): 174-193.