Efficient Textual Web Retrieval using Wavelet Tree

Arun Kumar Yadav, Ajay Kumar Garg Engineering College, Ghaziabad, India
Divakar Yadav, Jaypee Institute of Information Technology, Noida, India
Rajesh Prasad, Yobe State University, Damaturu, India

ABSTRACT

Searching on the web is one of the most progressive and expanding field nowadays. A large amount of information is available on the World Wide Web, motivating the need of efficient text indexing method that support fast text retrieval. In the past, two main indexing techniques: Signature files and Inverted files have been proposed. First require much larger space to store index and are more expensive to construct and update than inverted files. Second has been efficiently implemented using different structures like Sorted array and B-Tree. Sorted array was very expensive in updating the indices while appending a new keyword and B-tree method breaks down if there are many words with the same prefix. This paper presents a modified index structure for text retrieval that keeps a good result to optimize the space needed to store and time to search document. The proposed index is designed using the Wavelet Tree (WT), which was originally designed as wavelet transform for images. Experimental results show that on increasing the query length, the WT based index performs better than others.

KEYWORDS

Information Retrieval, Inverted Files, Text Searching, Wavelet Tree, Web Retrieval, Web Searching and Indexing Techniques, Web Searching

INTRODUCTION

There is no right or wrong way to carry out search over the Web. One need to do experiment by adding and deleting words and trying the different techniques. No search engine has indexed the entire web, and no two such engines are the same. Search engine stores document in the repository. It maintain index to receive user query and give result thereafter.

The problem of searching over the Web mainly decomposes into three components Jimmy & Dyer (2010): gathering web content i.e. called as crawling; construction of the inverted index i.e. called as indexing; and ranking of documents for a given query i.e. called as retrieval. Crawling and indexing have similar characteristics and requirements, but are very different when compared with retrieval. Gathering web contents and building inverted indexes are mostly done online. Both, crawling and indexing need to be scalable and efficient, but not to be operated in real time.

Indexing is basically a batch process that runs periodically, the frequency of refreshes and updates is usually dependent on the design of the crawler (Yadav, Sharma, Sanchez-Cuadrado, & Morato, 2012). Some sites like news organizations update their contents quite frequently and need to be visited often; other sites like government regulations are relatively static. However, even for rapidly changing sites, it is usually tolerable to have a delay of a few minutes until contents are searchable.
Furthermore, since the amount of content that changes rapidly is relatively small, running smaller-scale index updates at greater frequencies is usually an adequate solution. Retrieval, on the other hand, is an online problem that demands sub-second response time. Individual users expect low query latencies, but query throughput is equally important since a retrieval engine must usually serve many users concurrently. Furthermore, query loads are highly variable, depending on the time of the day, and can exhibit undesirable behavior due to special circumstances. For example, a breaking news event triggers a large number of searches on the same topic. On the other hand, resource consumption for the indexing problem is more predictable.

The suffix tree and suffix arrays method to index the text has been independently discovered in (Gonnet, 1987; Manber & Myers, 1990). Their study revealed the fact that the suffix arrays are more space efficient than suffix trees but it requires longer time to build it than suffix tree. Suffix arrays could have a similar performance in word searching as inverted file, which is logarithmic. The main advantages of suffix arrays are their availability to support certain kinds of searches which are difficult to perform over inverted files. For example, searching for phrases, searching by regular expressions, etc. is easy to search by using suffix array. Inverted files, consisting vocabulary, dictionary and inverted list has been efficiently implemented using different structures like sorted array and B-Tree. Sorted array are very expensive in updating the indices while appending a new keyword and B-tree method breaks down, if there are many words with the same prefix.

Signature files, an indexing technique applied on textual information retrieval had been proposed in (Zheng & Faloutsos, 1992). In this, a fixed-width signature, called bit-string is assigned to each record in the information retrieval system. Both conjunctive and disjunctive queries could be processed using the approach of signature files. However, signature files require much larger space to store index and are more expensive to construct and update than inverted files.

Wavelet tree, a space efficient data structure to represent a sequence and answer queries on it has been described in (Roberto, Gupta, & Vitter, 2003). A wavelet tree can be treated as a representation of a sequence, representation of a reordering of elements and representation of a grid of points (Navarro, 2007). New algorithms using wavelet trees for document retrieval has been developed by (Navarro, Puglisi, & Valenzuela, 2011). The algorithms were elegant and simple to implement, but, as per best of our knowledge and belief, inverted index has not been defined and implemented using wavelet tree. Wavelet matrix for alternative representation of wavelet tree for large alphabets has been discussed in (Francisco, Navarro, & Ordóñez, 2015). It is comparatively faster but, bit complex to construct.

This paper presents a modified index that keeps a good result to optimize space needed to store the index and time to search the documents. This modified index is designed using the wavelet tree implementing inverted list of inverted files. Performance of this new index with B-tree index and signature file index on database of one million Wikipedia pages has been performed. Experimental result shows that on increasing the query length, the wavelet tree based index performs better than others.

Rest of the paper is organized as follows. Sec. 2 describes the literature review. Sec. 3 presents our proposed wavelet tree based indexing technique. Sec. 4 presents experimental results. Section 5 presents research implication. Sec. 6 presents Limitation of work. Finally, Sec. 7 concludes the work and gives the future direction.
Related Content

Cloud4NFICA-Nearness Factor-Based Incremental Clustering Algorithm Using Microsoft Azure for the Analysis of Intelligent Meter Data

Performance Evaluation of Hierarchical SOAP Based Web Service in Load Balancing Cluster-Based and Non-Cluster-Based Web Server
Video Data Management and Information Retrieval
[www.igi-global.com/chapter/video-data-management-information-retrieval/30759?camid=4v1a](www.igi-global.com/chapter/video-data-management-information-retrieval/30759?camid=4v1a)

Searching and Mining with Semantic Categories
[www.igi-global.com/chapter/searching-mining-semantic-categories/64423?camid=4v1a](www.igi-global.com/chapter/searching-mining-semantic-categories/64423?camid=4v1a)