e-ISSN:2582-7219

INTERNATIONAL JOURNAL OF MULTIDISCIPLINARY RESEARCH IN SCIENCE, ENGINEERING AND TECHNOLOGY

Volume 6, Issue 2, February 2023

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 7.54

International Journal Of Multidisciplinary Research In Science, Engineering and Technology (IJMRSET)

| ISSN: 2582-7219 | www.ijmrset.com| Monthly, Peer Reviewed & Referred Journal |

| Volume 6, Issue 2, February 2023 |

| DOI:10.15680/IJMRSET.2023.0602007 |

Keyword Search Engine Optimization Concepts and Techniques

Dr.P.Anitha, Vishwa K

Professor & Head, Department of Computer Applications (MCA), K.S.R. College of Engineering (Autonomous), Tiruchengode, India

Department of Computer Applications (MCA), K.S.R. College of Engineering (Autonomous), Tiruchengode, India

ABSTRACT: keyword search is the most common application in data mining, for the fast search of keywords over the documents the search index method was introduced. The search index contains the keywords and the document list which is large in size. Several compression techniques introduced to reduce the size of the search index. In the compressed index it always needs to de-compressed during the document retrieval. The Optimized inverted index uses the document Id's as interval lists which reduces the size of search index. This search index requires less space to store and has high search efficiency, and the system also proposed the document reordering method to increase the continuous ID lists. The documents are reordered which shares more words common stay near. A novel relevance oriented framework is used to match the result. This can reduce the documents which are not in the continuous manner and also reduces the number of ID to store in the search index.

General terms: efficiency; search; size

KEYWORDS: compression; document retrieval; Inverted index.

I. INTRODUCTION

Keyword search is used to get the intelligible information from a large-scale information, web search engines have popularized keyword based search to support the keyword search. Users submit the keywords to the search engine for searching and ranked list of documents is returned to the users. Current keyword search system use the inverted index to retrieve the result for user query. Inverted index is an index data structure which stores the documents with corresponding ID's in ascending order, whenever the document is added to the database, knows posting file or inverted file. The two main variants in inverted index is record level inverted index and word level inverted index. Record level inverted index contains a list of references to documents for each word. Word level inverted index contains the list of references and positions of each word within the document. The real world datasets are so large that keyword search systems usually use the various compression techniques to reduce the space of storing inverted indexes. This compression of inverted index reduces the storage cost, but increases the disk I/O time for query processing. Since ID's are stored in ascending order, many techniques such as variable-Byte Encoding[2] and PForDelta[2], used to find the differences between the ID's where it is stored, by using this method storage space if ID's in inverted index is highly reduced, but decompression is required during query processing, which leads to additional I/O cost in time of retrieval during keyword search.

The existing method merges the continuous id's in the document as the single id using the upper and lower limit e.g{3,4,5,6,7,8,11,12,13,14,15,16,17}
{[3,8],[11,17]} L=[3,11] U=[8,17]. By encoding the continuous id's into single id, index size is reduced, but it does no support the word stemming. If the word stemming is not applied to inverted index more id's are occupied in inverted index. for e.g Two queries such as 1.Searching for hidden web database and 2. Navigation system for product search, among these two queries both searching and search related to same in keyword search due to removal of stop words, but it consume two different id's in inverted file, by applying word stemming two id's are reduced to one. In Optimized inverted index a relevance oriented frame work is supported to match the result for the user typed query.

| ISSN: 2582-7219 | www.ijmrset.com| Monthly, Peer Reviewed & Referred Journal |

| Volume 6, Issue 2, February 2023 |

| DOI:10.15680/IJMRSET.2023.0602007 |

II. RELATED WORK

Keyword-based search is a well studied problem for text documents and Internet search engines world among the users. Inverted files are common data structures used for solving query processing using keywords. our approach supports the relevance oriented framework to provide the relevant result in keyword search.

Yan and Ding [6] reported that the compression of inverted index in query processing techniques has a lot of benefits in keyword search. Previous work has focused on finding document orderings that minimize the index size under standard compression schemes such as PDICT, PFOR and PFOR-DELTA[2]etc., This motivates the several interested open questions. It shows that query processing benefit from more efficient skipping in reordered indexes. This was a natural side product of reordering, but additional improvements might be possible by combining reordering with the ideas in for selecting block boundaries in compressed indexes. Second, there is an interesting relationship between compression of reordered indexes and efficient indexing of archival collections.

Compression reduces both the time required to process the queries submitted by the users and inverted index size. The compression of inverted lists of document postings that store the position and frequency of indexed terms are revisited. Two approaches are considered to improving retrieval efficiency: better implementation and better choice of integer compression schemes. First, several simple optimizations to well-known integer compression schemes is used, this lead to significant reductions in time. Second, the impact of choice of compression scheme on retrieval efficiency is explored. Inverted indexes are used to evaluate queries in all practical search engines. Compression of these indexes has three major benefits for performance.

First, a compressed index requires less storage space. Second, compressed data makes better use of the available communication bandwidth; more information can be transferred per second than when the data is uncompressed. For fast decompression schemes, the total time cost of transferring compressed data and subsequently decompressing is potentially much less than the cost of transferring uncompressed data. Third, compression increases the likelihood that the part of the index required to evaluate a query is already cached in memory, thus entirely avoiding a disk access. Thus index compression can reduce costs in retrieval systems.

Shieh and Chen [3] method reduce the average gap values in an inverted file. This reassignment is based on a similarity factor between documents. The similarity between two documents as the number of common terms in their contents.

All documents hence form a similarity-graph in which a vertex represents a document and an edge between two vertices represents their similarity. The reassignment order can be generated by traversing all vertices in this graph with a gap-reduction strategy and transform this problem to the travelling salesman problem (TSP).

The similarity between two documents as the number of common terms both appearing in the documents contents. If two documents have large similarities, their identifiers will both appear in many inverted lists. Therefore, it is worth reassigning new closer identifiers to these documents. This similarity factor is used to measure how closer identifiers should be reassigned to the documents. A gap-reduction strategy in reassignment method and transform the problem to the TSP.

Internet search engines have popularized the keyword-based search paradigm. While the traditional database management systems offer powerful query languages, they do not allow keyword-based search. DBXplorer [9], a system that enables keyword based search in relational databases. It is efficient and scalable keyword search utility for relational databases. The search component of DBXplorer bears the resemblance to work on universal relations.

Data cleaning for matching similar data is done by the set similarities queries [8], similarities queries are work by the join operation. There are two main approaches in the design of efficient set similarities selection algorithm. The first approaches is based on relational database technology, e.g., using table, indexes and SQL. Second one is to design specialized disk resident indexes, in the form of inverted lists and TA/NRA algorithm is used to evaluate similarities.

| ISSN: 2582-7219 | www.ijmrset.com| Monthly, Peer Reviewed & Referred Journal |

| Volume 6, Issue 2, February 2023 |

| DOI:10.15680/IJMRSET.2023.0602007 |

III. INDEX SIZE REDUCTION

Optimized Inverted index (Opix), which is an extension of the traditional inverted index, used to improve the performance of keyword search.

ID	Content
1	Keyword querying and ranking in databases
2	Keyword searching in XML databases
3	Keyword search in relational databases
4	Efficient fuzzy logic search
5	Navigation system for artificial intelligence
6	Keyword search on spatial databases

Table 1 Sample database

The above table shows that IDs with the corresponding queries for keyword search.

Features of optimized inverted index

- Opix merges the consecutive IDs in inverted list and converted into intervals.
- Index size can be reduced by applying the word stemming to inverted index and using the concept of document reordering[3-7], which is used to reorder the documents by reassigning IDs to them to make the user to achieve high performance in keyword search.
- The optimized inverted index is also support the relevancy for the user submitted queries in text data for the keyword search.

Sample Words	IDs	Intervals
Keyword	1,2,3,6	[1,3],[6,6]
Databases	1,2,3,6,7	[1,3],[6,7]
Searching	2,7	[2,2][7,7]
Search	3,4,5,6	[3,6]
.....	.....	.....

Table 2: Inverted index& generalized inverted index

The intervals in table 2 shows that the two limits are needed to represent the intervals as lower bounds and upper bounds. Thus if there is many single-element intervals in the
interval list, the space cost will be more. The extra overhead for storing the intervals lists is reduced by splitting each original interval inst into 3 ID lists with one for single element intervals and the other two for the lower and upper bounds of multi-element intervals. Document reordering is used to reduce the size of inverted file, by switching the words. The below table 3 shows that by applying word stemming the size of the inverted index is further reduced.

| ISSN: 2582-7219 | www.ijmrset.com| Monthly, Peer Reviewed & Referred Journal |

| Volume 6, Issue 2, February 2023 |

| DOI:10.15680/IJMRSET.2023.0602007 |

Sample words	Intervals
Keyword	[1,3],[6,6]
Database	[1,3],[6,7]
Search	[2,2],[6,7]

Table 3: Optimized inverted index

IV. SEARCH ALGORITHMS

Keyword search usually supports union and intersection operation for the relational databases. union operation is the source operation for the ARE query semantics. Intersection operation is the source operation for the AND query semantics. The probe intersection algorithm usually runs faster for ID lists than the merge – based algorithm for real applications.

ALGORITHM

Input: *R* A set of interval lists.
Output: *G* The resulting interval list.

1: L be a max-heap and U be a min-heap

2: for all k Є [1,n] do

3: Let r_k be the frontier interval of R_k
4: Insert lb(r_k) and ub(r_k) to *L* and *U* respectively

5: while $U \neq \Phi$ do

6: Let *l* be the top (max) element in L

7: Let u be the top (min) element in U

8: if *l* and u is then Add [*l*, u] to *G*

9: Let r Є R_j be the corresponding interval of *u*

10: Remove lb(r) from *L* and pop *u* from *U*

11: Let r' be the next interval (if any) of r in R_j

12: Insert lb(r') and ub(r') to *L* and *U* respectively

13: return G

PROCESS:

Step 1: Consider L as the max-heap and U be a min- heap

Step 2: Repeat the process for all n

| ISSN: 2582-7219 | www.ijmrset.com| Monthly, Peer Reviewed & Referred Journal |

| Volume 6, Issue 2, February 2023 |

| DOI:10.15680/IJMRSET.2023.0602007 |

values.

Step 3: Insert the lower bound value to
maxheap and upper bound value to min heap.

Step 4: While U not equal to NULL
continue the process

Step 5: Consider *l* and *u* be top
maximum and minimum element respectively.

Step 6: If l and u is added to the resultant
list and consider r be the corresponding interval of u.

Step7: Repeat the process for all
elements.

V. EXPERIMENTAL ANALYSIS

The following graph show that index sizes varies using different compression techniques, it compares the inverted index, the inverted index compressed by VBE, the generalized inverted index, ginix by VBE, the optimized inverted index, opix, opix by VBE. Experiment shows that compression of opix + VBE is better than the other compression techniques.

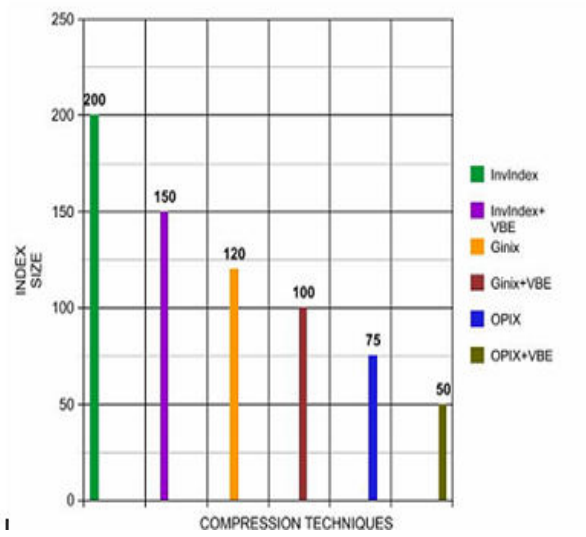

Figure 1

VI. CONCLUSION

Keyword search is usually improved by reducing the inverted index size and disk I/O time. This paper deals with the optimized inverted index for the text database to enhance the keyword search speed. The optimized inverted index supports the framework for the relevant in generated results for user queries.

REFERENCES

[1] F. Scholer, H. E. Williams, J. Yiannis, and J. Zobel, Compression of inverted indexes for fast query evaluation, in Proc. of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Tammpere, Finland, 2002, pp. 222-229.

[2] M. Zukowski, S. Hman, N. Nes, and P. A. Boncz, Super-scalar RAM-CPU cache compression, in Proc. of the 22nd International Conference on Data Engineering, Atlanta, Georgia, USA, 2006, pp. 59.

[3] W. Shieh, T. Chen, J. J. Shann, and C. Chung, Inverted file compression through document identifier

| ISSN: 2582-7219 | www.ijmrset.com| Monthly, Peer Reviewed & Referred Journal |

| Volume 6, Issue 2, February 2023 |

| DOI:10.15680/IJMRSET.2023.0602007 |

reassignment,
Information Processing and Management, vol. 39, no. 1, pp. 117-131, 2003.

[4] R. Blanco and A. Barreiro, TSP and cluster-based solutions to the reassignment of document identifiers, Information Retrieval, vol. 9, no. 4, pp. 499-517, 2006.

[5] F. Silvestri, Sorting out the document identifier assignment problem, in Proc. of the 29th European Conference on IR Research, Rome, Italy, 2007, pp. 101-112.

[6] H. Yan, S. Ding, and T. Suel, Inverted index compression and query processing with optimized document ordering, in Proc. of the 18th International Conference on World Wide Web, Madrid, Spain, 2009, pp. 401-410.

S. Ding, J. Attenberg, and T. Suel, Scalable techniques for document identifier assignment in inverted indexes. in Proc. of the 19th International Conference on World Wide Web, Raleigh, North Carolina, USA, 2010, pp. 311-320.

[8] S. Agrawal, S. Chaudhuri, and G. Das, DBXplorer: A system for keyword-based search over relational databases, in Proc. of the 18th International Conference on Data Engineering, San Jose, California, USA, 2002, pp. 5-16.

Impact Factor
7.54

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INTERNATIONAL JOURNAL OF MULTIDISCIPLINARY RESEARCH IN SCIENCE, ENGINEERING AND TECHNOLOGY

| Mobile No: +91-6381907438 | Whatsapp: +91-6381907438 | ijmrset@gmail.com |

www.ijmrset.com