A Survey on Retrieval Techniques

Miss. Yogini Dingorkar[1]
M.tech Scholar
Department of Computer Science and Engineering,
Tulsiramji Gaikwad-Patil College of Engineering & Technology
Nagpur, India.
yoginibangde@gmail.com[1]

Mr.Ankush Maind[2]
Assistant Professor, Department of Computer Science and Engineering,
Tulsiramji Gaikwad-Patil College of Engineering & Technology
Nagpur, India.
ankushmaind@gmail.com

Miss.Roshni Talmale
Assistant Professor, Department of Computer Science and Engineering,
Tulsiramji Gaikwad-Patil College of Engineering & Technology
Nagpur, India.
roshnikambe@rediffmail.com

Abstract— To obtain high query performance the efficient information retrieval must be needed. Today there are various methods are proposed for storing and retrieving the data. Many of this technique use compression while storing which require decompression while searching. To improve performance of the searching various union and intersection algorithms are available. The different encoding methods are presented to improve the compression ratio which works on indexes of the words. Inverted lists are usually needed to retrieve the documents. Inverted indexes are very large, so that various compression techniques have been studied to reduce the storage space and disk IO time. To improve the performance various reordering techniques are also presented. This paper gives comprehensive review on various searching, compression, reordering techniques which helps to generate efficient indexing while storing and retrieving.

Index Terms— Inverted indexes, intervals, document reordering.

1 INTRODUCTION

Key word search is done by queries, and queries must be processed efficiently, to improve the searching performance various compression techniques are used. Traditional keyword search technique uses the concept of inverted indexing to search the keywords from documents like web pages, database tables, XML documents. Indexing generates the posting list which specifies the addressing in which documents and position term appears. Many technique such as Variable_Byte_Encoding and PForDelta calculate d-gaps and stores binary representations of it. By using d_gaps we can improve compression of inverted index.

We can detect d-gap sequential patterns using a novel data structure For example, given the following inverted lists, (a) <1 3 4 5 5 2 6 1 1 7 4 5 3 2 9 5 5 2 6>; (b) <7 4 5 3 2 9 3 4 1 5 5 2 6 1>; (c) <1 7 7 4 5 3 2 9 3 6>, we can find that d-gap sequences <5 5 2 6> and <7 4 5 3 2 9> both occur 3 times in the lists. If we can represent them as a pattern (using a pattern Id) in the inverted lists that contain them, we may store the lists with fewer bits.[d gap] . Compression technique applied on documents required decompression while processing the query which leads to increase time complexity and cost. We are describing different the different search, compression, reordering algorithms and maximizing d_gaps. The searching algorithm based on union and intersection of indexes does not require decompression while searching.

In this paper Section 2 represent various searching algorithms based on union and intersection. Section3 elaborate different encoding technique which is used for compression. Section4 gives different reordering methods which helps to generate the efficient intervals. The paper concludes the performance of different searching, encoding and reordering in Section 5.

2 Search Algorithms

In traditional keyword search algorithms, keywords are searched by using union and intersection operation on IDs .The operation require compressed inverted list for each keywords, which is then decompressed to perform intersection and union. Intersection operation works according to AND operator. The intersection operation returns the result by matching all the keywords present in query. The concept of scanline algorithm is merging by intersection operation to generate SCANLINEISECT algorithm. In this algorithm upper and lower bounds are maintained by using single heap and active intervals are maintained to indicates the interval currently being processed. Though IDs list are very large decompression requires extra computational cost so these methods are very expensive.

2.1 Scanline Union algorithm

Scanline union algorithm uses the concept of union operation. This algorithm is inspired by scanline rendering algorithm of graphics. In this concept to calculate union list boundaries of intervals are stored in ascending order and scanline moves from smallest to largest

boundary. In this algorithm one counter is maintained when scanline hits to lower bound and that interval is stored in a and counter is incremented. When it hits to upper bound counter is decremented and that interval is stored in b. In such way [a,b] is generated as resulting interval. All upper bounds and lower bounds are enumerated in ascending order by heap_merge.

2.2 Scanline Union + algorithm

In this algorithm initially all the pointers are pointing to first interval list and active interval is set to be empty. The functionality of scanline union+ algorithm is improved by maintaining the active interval which indicate the current result interval. This algorithm inserts only lower bound l to the heap Ḥ. In each step, the algorithm pop minimum lower bound in the heap, and then extends active interval if two intervals overlap. When popped lower bound and active interval do not overlap then active interval is returned as a resulting interval and its lower bounds and upper bounds are updated.

2.3 Twin heap algorithm

This algorithm is better than SCANLINESECT algorithm, in this algorithm two heaps are used to store upper bounds and lower bounds to improve the performance. This new algorithm is called TWINHEAPISECT. This algorithm is efficient than SCANISECT algorithm because each heap require for insertion is 50% smaller.
Time complexity C_H =O(log n⊙ Σ_k | R_k |). This algorithm is 20% faster than IDHEAPSECT algorithm.

2.4 Probe-based algorithm

This algorithm is faster algorithm. Probe based algorithm uses the concept of binary search for probing which is efficient than the sequential scan. The probe based algorithm takes R as set of interval lists and sorts the R in ascending order of lower bounds.
The time complexity of PROBISECT is C_P = O(min(log n ⊙ $\Sigma_{K\neq J}$ | R_K | }). When the number of intervals in the list are greater than number of IDs PROBISECT is more costly. PROBISECT is faster in query based keyword search.
PPOBISECT +
In this algorithm the unnecessary probes are avoided by calling Probe function recursively. In this method empty probes are avoided by probing the lists sequentially. It runs faster than the PROBEABSED.

3 Inverted List Compression Schemes

In order to increase the performance of the retrieval we have to use compression on inverted list. Compression reduces the cost of transferring data from memory to CPU, than uncompressed data. As inverted index consist of vocabulary of terms (words from collection) and inverted lists (which are vectors that contain information about the occurrences of words). In this paper we consider integer compression schemes. There are two compression schemes bitwise and byte wise to store inverted list. Elias gama, Elias delta, Golomb_rice codings are bitwise coding schemes.
The basic compression schemes for inverted lists are as follows.

3.1 Elias gama

Elias coding [12] is bitwise compression scheme it is non parameterized that means it uses only fixed or static codes to store integer .In this method positive integer K is represented by 1+[$\log_2$ k] store as unary code, followed by binary representation of K without its most significant bit.This scheme compactly represent small integer but coding is inefficient for storing integer larger than 15.

3.2 Elias delta

Elias Delta [12] is also bitwise compression scheme, A delta code uses the gamma code as building block and represent integer k as 1 + blog2 kc, and then the binary representation of k without its most significant bit. Elias delta codes are suited to coding larger integers, but are inefficient for small values.

3.3 Golomb encoding

Golomb [13,4,8] is the faster retrival than the Elias Golomb coding is a lossless data compression method invented by Solomon W. Golomb in the 1960s. Golomb coding highly suitable for situations in which the occurrence of small values in the input stream is significantly more likely than large values. Golomb coding, encodes an integer b in two parts quotient q and remainder r .quotitent q is stored as unary code and remainder r is stored in binary form. To encode a set of integers, we choose a parameter *b*; a good choice is *b* = 0:69 ¢ *ave* where *ave* is the average of the values to be coded . Then for a number *n* we compute *q* = *n/b* and *r* = *n* mod *b*. If *b* is a power of two, then *log*(*b*) bits are used two store the remainder *r*; otherwise, either *log* (*b*) or *log*(*b*) bits are used, depending on *r*.

3.4 Rice encoding

Rice coding [13,4,8] is invented by Robert F. Rice It is the oldest bit-aligned method It is the oldest bit-aligned method.. Rice Coding is variant of Golomb coding this method is well in compression size but slow compared to recent compression method. ”Rice coding" can refer either to that adaptive scheme or to using that subset of Golomb codes. Whereas a Golomb code has a tunable parameter that can be any positive integer value, Rice codes are those in which the tunable parameter is a power of two. Despite the restriction to powers of two, Rice coding is often significantly 4 to5 factor slower than variable-byte coding. Rice codes convenient for use on a computer, since multiplication and division by 2 can be implemented more efficiently in binary arithmetic. Disadvantage of rice encoding is that the compressed data may require more space.

3.5 Variable byte encoding

Variable byte encoding [10] scheme is 2x faster than the Variable bit encoding scheme. It very simple byte wise compression scheme. Uses 7 bits to code the data portion and the most significant bit is reserved as a flag bit which indicate if the next byte is still part of the current data VBE compression method reduces cost of transferring data from memory to the CPU than that of transferring uncompressed data. Variable-byte coding is simple to implement significantly faster than previous traditional bit oriented methods such as Golomb, Rice, Gamma, and Delta coding.
Variable-byte coding results in significantly faster query evaluation than those previous methods, which were CPU limited due to their large decompression costs.
The disadvantage of variable-byte coding is that VBE is not efficient for ordered collection document, compressing frequencies, or docIDs in very long lists having small d_gaps . Moreover does not achieve the same reduction in index size as bit-aligned methods.

3.6 PFOR Delta encoding

The PForDelta encoding [3,7] compression method classify inverted list into either coded or Exception values .Exception values are stored in to uncompressed form but we still maintain the slots from them in their corresponding positions and coded values are assigns with the arbitrary bit_width b which kept constant within a disk block. Inverted list divided in to blocks, 128 Dgaps each. PForDelta method finds the largest Dgaps in block say x and represent it as b=[log X] bits.For each block header is maintained to indicate compression used within a block. In case of exceptions we store the position of first exception in the header. Each exception stores the offset of next exception in its slot thus forming link list. This method requires large space usage when the lists contain many small numbers.
For decompression PForDelta have specialized function that obtains the b-bit DGaps. PFor-Delta hast efficient compression schemes, achieving a high decompression speed.

4 Document Reodering

Document reordering is required for ordering the data to generate the lists containing fewer intervals which require less space for storage. Studies of document reordering have all been designed for unstructured long documents. Document reordering needed to for structured or short documents.

4.1 TSP heuristic algorithm

Shieh et al. [9] proposed a DocID reassignment algorithm adopting a Travelling Salesman Problem (TSP) heuristic it is graph based system. In this algorithm associated document are searched to built similarity graph G. Each document is consider as vertex and edge is inserted between similar document. Vertex should share at least one term. The TSP heuristic algorithms finds the suboptimal cycle. This cycle is used for reassignment of the DocIDs. The reassignment achieves good compression ratio.
This algorithm is costly for real web collection, because it requires whole graph should be store in memory.
Reordering of 132000 documents require about 23 hours and 2.17GBytes in memory.

4.2 B & B algorithm

Blelloch and Blandford [5] also proposed an algorithm called B&B. This algorithm permutes the document identifiers in order to enhance the clustering property of posting lists. This algorithm creates similarity Graph G from IF index, each document consider as vertex of graph the edges of the graph are weighted by considering cosine similarity measure between each pair of documents. Then graph G recursively splits into smaller subgraphs to generate singleton. The depth_First traversing is applied on tree to reassign the DocIDs.
This approach require whole graph to be strore in main memory which results in high space and time complexity also this approach require expensive graph splitting . B&B algorithm is costly algorithm and not suited for real web collection.

4.3 Greedy NN TSP by using SVD

In [4], Blanco and Barreiro present efficient algorithm for reassignment of document algorithm based on the Greedy-NN TSP algorithm. It uses SVD (Singular Value Decomposition) technique provides dimensionality reduction by rearranging the input data in a lower dimensionality space, which reflects major association pattern between document terms on which Greedy-NN TSP algorithm is applied. This algorithm results obtain good compression ratios with low running times. SVD transformation on the matrix might spoil the scalability of the method. Indeed, the block partitioning approach proposed seems to reduce this effect but costs quite a lot in terms of effectiveness degradation.

4.4 Lexicographical ordering of the URLs

Silvestri[5] show that in the case of collections of Web Documents the performance of compression algorithms can enhance by simply assigning identifiers to documents according to the lexicographical ordering of the URLs. Drawback of all the previous approaches is that they focus on reassigning DocIDs appearing in a previously built IF index. This algorithm require sorting of URls so this is very efficient method and not require any set of intersection operation.
Sorting web pages in lexicographical order based on their URLs as an acceptable solution to the problem. Since the URLs are good indicates of web page content this method is reasonable reordering method.
Drawback of this method is that it is not applicable to datasets whose URLs do not represent meaningful content (e.g., Wikipedia pages), or even do not have a URL field.
URL sorting technique is the most efficient technique for assigning docIDs in the case of Web Search Engines when considering the classic time-space trade-off.

4.5 Signature sorting algorithm

SIGSORT [1] algorithm works by generating signature of the words for that a summary of each document is generated then words are arranged in descending order of their frequencies. The top n (e.g. n D 1000) most frequent words are chosen as signature vocabulary. While sorting the document is sorted word wise. Reordering the document by their signature is very efficient technique because frequent words have consecutive IDs in its inverted list. SIGSORT is more suitable for structured and short text data and can handle large data. It provides higher clustering power.

5 Conclusion

This paper presented the different approaches which are used for indexing which will helps for efficient retriving. Different Searching techniques uses the union and intersection operations to find the results of queries, these methods works on OR query and AND query semantics researchers Hao Wu, Gauliang Li and Lizhu Zhau presents SCANLINEUNION+ and PROBISECT+ algorithms in which PROBISECT+ works better for searching because it is faster and avoids unnecessary probes.
The compression techniques presented by Elias, Golomb, Rice Encoding has some disadvantages which are overcome by VBE and PForDela.
Reordering techniques presented by Shieh et al, Blelloch, Blandford, Blanco, Barreiro, Silvestri, Hao Wu. Among this reordering techniques performance of Signature sorting is better because it provides clustering power.
In future, for improving the indexing we would like to take more information resources or combination of existing system or new one. So that result will be improved.

References

[1] Hao Wu, Guoliang Li, and Lizhu Zhou, Ginix: Generalized Inverted Index for Keyword Search IEEE TRANSACTIONS ON KNOWLEDGE AND DATA MINING VOL:8 NO:1 YEAR 2013

[2] Vijayashri Losarwar, Dr. Madhuri Joshi Data Preprocessing in Web Usage Mining International Conference on Artificial Intelligence and Embedded Systems (ICAIES'2012) July 15-16, 2012 Singapore.

[3] M. Hadjieleftheriou, A. Chandel, N. Koudas, and D.Srivastava, Fast indexes and algorithms for set similarity selection queries, in Proc. of the 24th International Conference on Data Engineering, Cancun, Mexico, 2008,pp. 267-276

[4] J. Zhang, X. Long, and T. Suel, Performance of compressed inverted list caching in search engines, in Proc.of the 17th International Conference on World Wide Web, Beijing, China, 2008, pp. 387-396.

[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] 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.

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

[8] J. Zobel and A. Moffat, Inverted files for text search engines, ACM Computing Surveys, vol. 38, no. 2, pp. 6, 2006.

[9] Wann-Yun Shieh, T ien-Fu Che n, Jean J yh-Jiun Shann, and Chung-Ping Chung.Inve rted file compre ssion through do cument identifie r reas signment. Information Processin g and M anage men t, 39(1):117–131, January 2003.

[10] 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.

[11] [B& B]Dan Blandford and Guy Ble llo ch. Index compression through document reordering. In Proceedings of the D ata Compression Confere nce (DCC'02), pages 342–351, Was hington, DC, USA, 2002. IEEE Computer Society.

[12] P. Elias. Universal codeword sets and representations of the integers. IEEE Transactions on Information Theory, IT-21(2):194{203, Mar. 1975.

[13] S. Golomb. Run-length encodings. IEEE Transactions onInformation Theory, IT {12(3):399{401, July 1966.