International Journal of Engineering Research and General Science Volume 3, Issue 5, September-October, 2015
ISSN 2091-2730

Multiterm Keyword Searching For Key Value Based NoSQL System

Pallavi Mahajan[1], Arati Deshpande[2]

Department of Computer Engineering,
PICT, Pune, Maharashtra, India.
Pallavinarkhede88@gmail.com[1], ardeshpande@pict.edu[2].

Abstract— Today, the enterprise landscape faces large amount of data. The information gathered from these data sources are useful for improving on product and services delivery. However, it is challenging to perform searching activities on these data sources because of its unstructured nature Due to unstructured nature of these data, NoSQL storage has been adapted by many enterprises because it provides better storage facility. NoSQL storage can store schema oriented, semi structured, schema less data and its document append storage has received high attention because it provides the flexibility to store JSON based data. However to provide effective indexing and searching on these data is a challenging task. This work discuss concept of multiterm inverted index which is used with index pruning algorithm. The multiterm inverted indexing reduces the overhead for intersecting the inverted lists of keys in multi-term queries, by storing combinations of terms as keys in the index.

Keywords— Data mining, inverted indexing, NoSQL system, multikeyword searching, ranking, key-value store, index pruning.

I. INTRODUCTION

The concept of Index has been devised in order to enable flexible use of recorded data and make these large data sets easily accessible. One of the main design concepts to allow access to data in multiple ways is using Indexing. There are some indexing techniques available but the most widely used indexing technique is inverted indexing. Inverted index is an index data structure that maps a number or word to its location in database or in document or a set of documents. The purpose of inverted index is to provide full text searches. There are two main types of inverted index. First, A Record Level Inverted Index (or inverted file index or inverted list) contains the list of documents that contains the word. Second type is a word level inverted index (inverted list or full inverted index) contains the position of each word within document. Almost all of the large scale text retrieval systems make use of compressed inverted list indexes, which are considered the most useful indexing technique for very large collections. here are two ways to create inverted index that are on single term and on multiple terms.

The inverted index on multiple terms performs faster than single term. However the data being generated nowadays have no schema and has no particular structure. The data comes in several formats such as multimedia data, files, but the RDBMS systems are designed to store schema-based data in certain predefined format and do not support these types of data. Thus, the NoSQL storage has been proposed to support both large amount of data and data in unstructured formats. Multimodal databases, graph databases, object databases, are the different types of NoSQL storages. In this work, we shall focus on the document-append style NoSQL storage (mongoDB) for storing the documents. This style of storage supports structured, semi-structured, and unstructured data. The NoSQL system implements key-to-value map in their core featuring like hash table which uses PUT/GET /DELETE methods for inserting, accessing and updating data. As the NoSQL supports large amount of data, the problem however that is, we are faced with the challenge of performing data mining tasks from such storage facilities.

There are various solutions existing that perform searching only with single term inverted index which does not scale very well on large databases. Since multi-term queries represents the majority of user queries, to support this various approaches that utilizes multi-term inverted index are proposed in previous work which consider all the n terms and in that case the number of possible term combinations are in O(n) which is the worst case. To select the meaningful subsets of terms is one of the main considerations of the proposed system. The two main contributions of the paper are 1) Selection of meaningful terms for combination. 2) Use of index pruning algorithm to minimize the scale of the term set index. The remainder of this paper is organized as follows: section II discusses related work. In section III the design of the system is presented. Section IV shows the half experimental results and the paper is concluded in section V

II. RELATED WORK

There has been significant amount of work done on indexing with distributed environment (peer to peer networks) in which several different ways to organize a text index (inverted index) are mentioned such as local index, global index and hybrid index. The traditional solutions based on relational database systems, are limited to key based queries. i.e. queries where data object can be retrieved using primary key. The use of inverted index has a long history in information retrieval. Their use in computer vision was pioneered by Sivic and Zisserman. The inverted index data structure is central component of search engine. Basically An inverted file is the sorted list (or index) of keywords (attributes), with each keyword having links to the documents containing that keyword.

A lot of work has been done on peer to peer search engine using inverted index which are in distributed manner using single keyword and multiterm keyword search. P. Reynolds [5], proposed efficient peer to peer keyword search in which he designed fully distributed search system using inverted index with keywords evenly distributed among available servers. This work describes the design of distributed inverted index including three techniques: bloom filters, caches, incremental results which minimize the bandwidth during keyword searches. Bloom filters are useful for eliminating the need to send entire document match list among servers. Caching reduces the frequency with which servers must transfer the Bloom filters. Incremental results allow search operations to halt after finding a fixed number of results, which reduces the cost of searching.

The use of bloom filter is an effective way to reduce the communication cost but simply using the bloom filter to minimize the false positive rate raise the high traffic cost [9]. Here they address the problem of optimizing the settings of a BF and show, through mathematical proof, that the optimal setting of BF in terms of traffic cost is determined by the statistical information of the involved inverted lists, not the minimized false positive rate as claimed by previous studies. The advantage of this design is it significantly reduces the search traffic and latency of the existing approaches.

In [8], the general keyword index and search scheme for structured p2p networks is proposed to avoid several problems such as unbalanced load, fault tolerance, storage redundancy that are in peer to peer network. This paper presents general keyword index and search scheme for structured peer to peer network. In this paper a general keyword index and search scheme for DHT network is proposed where each object is map to an r-bit vector according to its keyword set and view this r-bit vector as points in an r-dimensional hypercube. This hypercube index technique is efficient when more keywords are given. This hypercube index technique provides the facility of ranking and query expansion.

The index scheme proposed in above paper is decomposable that means instead of using single large hypercube to index objects, the entire keyword set is divided into smaller subsets and on that small subsets the hypercube is used. While some approaches [4] used discriminative keys for indexing and retrieving, in which the keywords that are appear in most of documents are marked as discriminative keys, and the keywords having low discriminative power are marked as non discriminative keys.

To reduce the number of generated keys proximity filtering method is used. In this contribution a novel indexing/retrieval model that achieve high performance, cost efficient retrieval by indexing with highly discriminative keys (HDKs) stored in a distributed global index maintained in a structured P2P network. HDKs correspond to carefully selected terms and term sets appearing in a small number of collection documents. The results shown in this paper despite increased indexing costs, the total traffic generated with the HDK approach is significantly smaller than the one obtained with distributed single-term indexing strategies.

The existing techniques for keyword search in structured peer-to peer (P2P) networks support only to single- keyword or exact match lookups. In multiple-keyword search, the query contains more than one word and in storage the multiple terms are associated with data item which contain that word. The search result for a query should include all the words contained in query. Traditional DHT-based approaches performed the multikeyword searching by storing a data item or its index multiple times, that is one keyword is stored against its associated document. A query is processed by searching each of the query keywords once. Therefore, the storage cost and search cost are both linear with the number of keywords.

In [3], a hybrid structured network called Mkey is proposed to address this problem. Its backbone is a structured network in which the data store in the form of clusters. At the time of inserting a data item, multiple copies of the data and its index are stored in a few different clusters. When user enters the query, the query is also mapped to multiple clusters, and the searching is performed within these clusters. After searching the union of all the search results is returned to users as the final result. In [10], a more efficient index

structure, the Generalized Inverted Index (Ginix) is proposed in which consecutive IDs in inverted lists are merges into intervals to save storage space. Along with this index structure, more efficient algorithms can be implemented to perform basic keyword search operations such as the union and the intersection operations. A keyword search using Ginix can be more efficient than traditional inverted indices. The results shows that Ginix requires less storage space as well as it also improves the keyword search performance, compared with traditional inverted indexes.

In [11] extension of the Generalized Inverted Index (GINIX) is presented. Ginix merges the consecutive IDs in inverted lists into interval lists and it reduces the size of the inverted index. The new index structure is called Extended Inverted Index (XINIX) which extends the structure of Ginix. The primary objective of Xinix is to minimize the storage cost. Xinix not only reduces the storage cost, but also increase the search performance, compared with traditional inverted indexes.

All the work mentioned above is for distributed environment but as the amount of unstructured data has growing day by day, there is a need to propose the effective approach to search over this unstructured approach. In [2], Multiterm keyword searching is performed in NoSQL system in which to perform the multiterm keyword search inverted index is created on set of combination of terms, the creation of the combination of terms is a crucial part. As each document contains thousand of keywords making the combinations of all these terms is very difficult task. To address this issue selection of more representative keyword is one of the considerations of this work.

III. SYSTEM DESIGN

This section introduces the design overview and then discusses the Multiterm indexing scheme with proposed pruning method.

A. System outline

The system involves three modules: data preprocessing module, construction of inverted index module and query processing (fig.1). Every document in document collection goes through some preprocessing steps like tokenization, stemming, normalization, stop word removing etc. are applied on each document which gives list of tokens. Before indexing the terms the TF (Term Frequency which denotes the frequency of term within the document) and IDF (Inverse Document Frequency which denoted number of documents containing the term) of each term is calculated. The values of TF and IDF are used to calculate the weight of terms. To calculate the similarity between document and term set and to rank the results vector space model is used. After calculating the similarity between set of combination of terms and documents and the top term set-document pairs are stored in the index. In order to reduce documents and the top term set- document pairs are stored in the index. In order to reduce the size of index pruning method is used which inserts the meaningful and top term set-document pairs in the index.

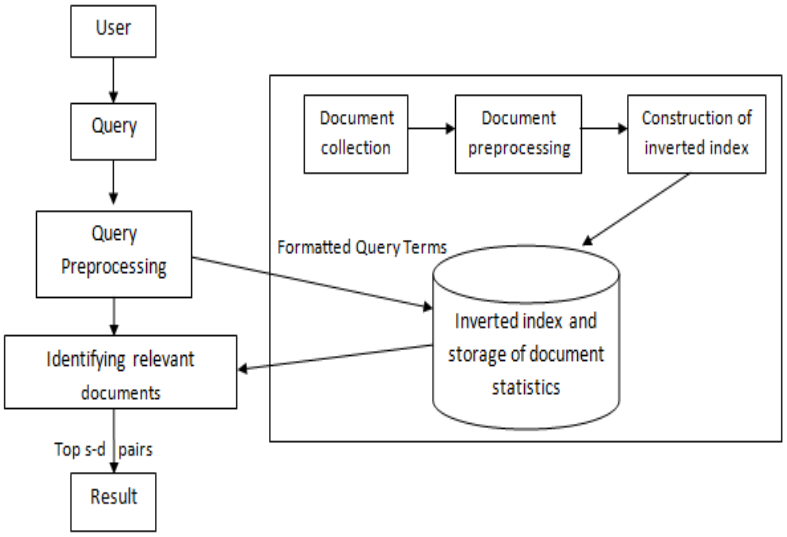

Fig. 1: Creation of Inverted Index

B. Multiterm indexing

The entire document collection goes through various data preprocessing function which gives list of terms. Now with the list of terms and the list of documents, the inverted index can map the term set to the list of document. In a given document there are n distinct terms and hence the set of all possible combined terms has 2^n element which is the worst case hence the pruning method is proposed to address this issue. To select the meaningful terms the concept of TF $\times$ IDF is used by using which the weight of each term is calculated and only top w_{max} terms are selected for the combination.

$w_{max} = \lambda\, n \log n$…… (1)

Where n is the number of documents and λ is the pruning parameter used to set pruning scale. The basic strategy of the pruning method is to index the short term combinations only ($S_{max} = 3$) as the long term combinations have little opportunity to be queried. With ($S_{max} = 3$) the index for n-term document can be significantly reduced. To calculate the Similarities between term set and documents the vector space model is used.

$Sim_{(s,d)} = \frac{\sum w_{s,t} \times w_{d,t}}{\sqrt{s} \times \sqrt{d}}$ ….. (2)

In above equation d is document and we compute the relevance of each document with term set s by cosine similarity: where $\omega_{s,t}$ and $\omega_{d,t}$ represents the weight of term t in s and d respectively.

$\omega = TF \times IDF$ ……... (3)

To calculate the weight of term $TF \times IDF$ scheme is used. TF denotes the term frequency of the term associated with the document.

$TF = 1 + \log(f_{d,t})$……. (4)

Here $f_{d,t}$ is number of times the term present in document d. IDF denotes the inverse document frequency of the term.

$IDF = (1 + \frac{N}{f_t})$…… (5)

Here, N denotes the total number of documents in collection and f_t is the number of document that contains the term t.

C. Steps in Index Pruning

1) For each document in collection perform document preprocessing.
2) Compute w_t i.e. weight of each term associated with the document using $TF \times IDF$.
3) Compute w_{max} using (1) and based on weight select only top w_{max} terms for indexing.
4) Similarly, for $S_{max} = 2$ repeat step 3.
5) For $S_{max} = 3$ check if the combination has frequent subsets or not, if yes then create an index on the set, otherwise keep the subsets in index as it is.

IV. EXPERIMENTAL SETUP AND RESULTS

An implementation is carried out with the help of JAVA Eclipse IDE, and NoSQL database The experiment is carried out using Reuters dataset which was originally collected and labeled by Carnegie Group, Inc. and Reuters, Ltd. The dataset is converted into mongodb dataset which contains document id, category, tokens. All data set was preprocessed with various document preprocessing operations to reduce the number of words.

Fig 2 plots average precision over recall level and reflects the search behavior. The results show that MTKS has better precision at almost all given recall levels than STKS. Fig. 3 shows the precision of top k that is top ranked documents returned by MTKS and STKS

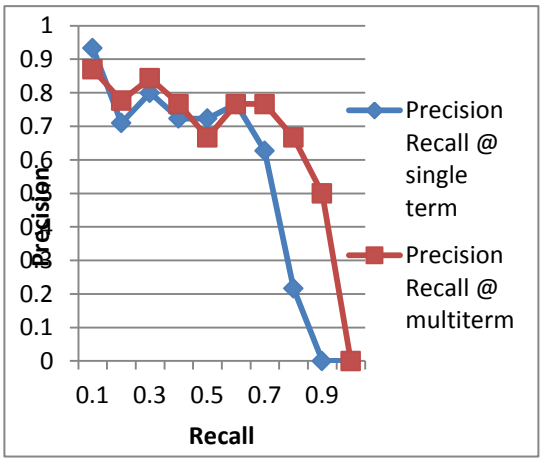

Fig. 2: Precision recall curve

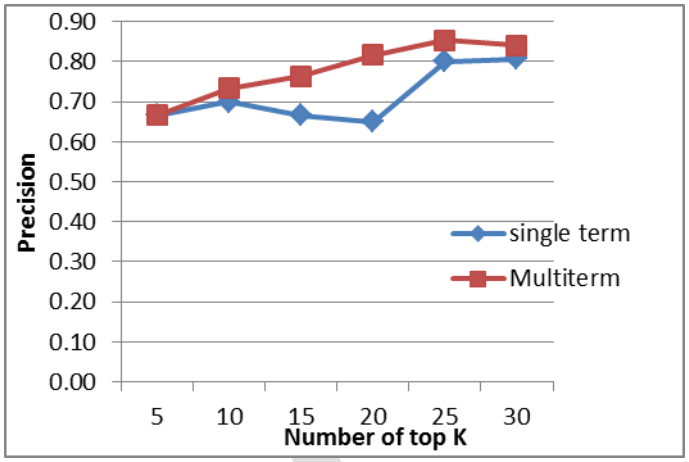

Fig. 3: Precision @ top k documents

V. CONCLUSION

The NoSQL systems are more and more deployed as a backend for various applications nowadays which support both Semi structured and unstructured data. The keyword based search based on single term inverted index performs poorly on such type of data. Therefore in above work the inverted indexing technique has been discussed to perform multiterm keyword search over the unstructured data. The effective index pruning method is used to reduce the index size in which the more representative keywords are selected to create term set with the aim that to provide efficient result to the user.

REFERENCES:

[1] H. Chen et al., TSS: Efficient Term Set Search in Large Peer-to- Peer Textual Collections, IEEE Trans. Computers, vol. 59, no. 7, 2010, pp. 969-980.
[2] C. Wetch, Anwitaman Data, Multiterm Keyword Search In NoSQL System,IEEE Internet Computing, 2011.
[3] Xing Jin, W.-P. Ken, Yiu S.-H., Supporting Multiple-Keyword Search in A Hybrid Structured Peer-to-Peer Network, Gary Chan Department of Computer Science The Hong Kong University of Science and Technology Clear Water Bay, Kowloon, Hong Kong
[4] I. Podnar et al., Scalable Peer-to-Peer Web Retrieval with Highly Discriminative Keys, Proc. Intl Conf. Data Eng. (ICDE 07), IEEE Press, 2007, pp. 1096-1105.
[5] P. Reynolds and A. Vahdat, Efficient Peer-to-Peer Keyword Searching, Proc. Middleware, Springer, 2003, pp. 21-40.
[6] J. Zhang, T. suel, Efficient Query Evaluation on Large Textual Collection in a Peer to Peer Environment, CIS dept. polytechnic university Brooklyn, NY 11201.
[7] H. Chen et al., Efficient Multi-Keyword Search Over P2P Web,Proc. Conf. World Wide Web (WWW), ACM Press, 2008, pp. 989-998.
[8] Yuh-Jeer Joung, L. Yang, C. Fang, Keyword Search In DHT Based Peer To Peer Networks IEEE conference 2007
[9] Hanhua Chen, Hai Jin, Lei Chen, Optimizing Bloom Filter Settings in Peer-to-Peer Multi keyword Searching, Liu, Senior Member, IEEE, and Lionel M. Ni, Fellow, IEEE.
[10] 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.
[11] Uniform access to NoSQL system paolo atzeni, Francesca bugiotti, luca rossi.
[12] D. Li, J. Cao, X. Lu, and K. Chen, Efficient Range Query Processing in Peer-to-Peer Systems, IEEE Trans. Knowledge and Data Eng., vol. 21, no. 1, pp. 78-91, Jan. 2009.
[13] O.D. Gnawali, A Keyword-Set Search System for Peer-to-Peer Networks, Masters thesis, MIT, 2002.
[14] J. Lu and J. Callan, Content-Based Retrieval in Hybrid Peer-to-Peer Networks, Proc. Intl Conf. Information and Knowledge Management (CIKM), 2003.
[15] I. H. Witten, A. Moffat, and T. C. Bell, Managing gigabytes: Compressing and indexing documents and images, Morgan Kaufmann Publishing: San Francisco, May 1999, PP. 36-56.
[16] E.-J. Goh, Secure indexes, Cryptology ePrint Archive, Report 2003/216, 2003.
[17] P. S. Mahajan, A. R. Deshpande, Review on Multiterm Inverted Index, CIIT 2015, in press.

[18] J. Lu and J. Callan, Content -Based Retrieval in Hybrid Peer-to-Peer Networks, Proc. Intl Conf. Information and Knowledge Management (CIKM), 2003.