PROVIDING SECURITY TO THE SENSITIVE DATA BY USING THE MULTI DIRECTIONAL FUZZY TECHNIQUE

P. Kishore Babu[1], P.Lavanya [2]

[1]M.Tech Scholar (CSE), [2]Assistant Professor

Nalanda Institute of Technology (NIT), Siddharth Nagar, Guntur, A.P., (India)

ABSTRACT

With the explosion of semi structured data users information stored or retrieved in the personal information management system, there is require of using some searching tools to retrieve the heterogeneous information efficiently. All the previous tools are only supporting IR-style ranking on text part of the query.For clarifying these techniques we need to consider the structure of a file (file directory) and file metadata (file type). Here we are proposing a novel multi dimension key search technique, in this methodology user will perform fuzzy key word, structure searches and metadata searches in addition to keyword conditions.This technique wills separately scores individuals dimension after that it assembles all these3 types of dimensions scores in to single unit as an appropriate meaningful score. We also make algorithms and indexes to effectively find out the most related files thosematch to multi-dimensional query. We are also performing experimental systematic mechanism evaluation that also shows in non-content dimension scoring situation will emotionallyprogress exactness of ranking. We also present the query processing policy to make the fuzzy search for usage every day.

Keywords: Personal Information Management, Multi-Dimensional Search, IR-Ranking, Query Processing.

I. INTRODUCTION

In day to day environment the data storing in personal information management system is usually increasing, the reason behind this is to dropping the price and the data persistencecapacity is increasing. This improvement of storage is leads to a problem for searching tools to content storage and information retrieval. These tools will helps us in both efficient query processing capacity and high quality scoring frame work techniques.

Many tools are available in the market to perform keyword search to find the personal information in the local system, such as spotlight search andgoogle desktop search. However these tools are form IR ranking for textual part of the query. It is similar to what was completed in information retrieval (IR) collective.The filtering tools only considering structure of the file (file directory) and metadata of the file (e.g type of file, date). In recent days the searching techniques concentrate focused on data spaces and personalized information. Whereas the informationencloses is variable in nature. Thus, the commercial file searching tools are working on IR-style keyword query search, and needs to use other system files information to monitor the keyword search.

The keyword search is only not sufficient, as described by using the bellow mentioned Examples.

Example1: This will consider the user will store his content in personal computer system. In the process of content calculation it will consider structure location of the file (file directory) and metadata information of the file (file type) which was stored in the same file system.

In some times user might ask the query like:

[File type = *. doc AND

createDate = 21/10/2014 AND

content = "Proposal Draft" AND

structure = /docs/way of file /Propose].

The prevailing tools will give response by recurring the file type is *. Doc and made on 21/10/2014 and require content corresponding to "Proposal Draft" (ranking) which are presented in the directory /docs/wayf/Propose (filtering conditions), how much effectively the information matching to novel Draft by using some natural text scoring mechanism. the total information except content which was used by filtering conditions, files that are related to the query, but this do not satisfied the particular conditions was overlooked. For example a file type is typed as *. txton 21/10/2014 presented in the directory /docs/wayf/Propose includes the content "Proposal Draft" will notreturn.

In many search conditions we are discussion that allowingresistance structure and metadata will increasing the usage and quality of search results. In the above examples if the user mightbe not remembering the exact date of file created, but remember that it was created around 25/10/2014. Thusprimarily the user need the file type is *.doc, he also want deliberate the associated files which are associated to the type of *.text or *.txt. At lastly, the user not reminisce where exactly the file was stored, at this movementwith the help of structure conditions and file size are only not as filtering conditions, as part of query the ranking conditions will returns better answers as part of search results.

Here in this paper we are proposing, a new technique, that lets the users to efficiently perform fuzzy keyword search using 3 types of dimensions: metadata, structure and content. We are definingan independent IDF-based scoring technique for all dimensions it represent a united scoring framework for multi-dimensional query systems on the personal information system. To finding scoring fuzzy matching more effective we are giving index construction optimization and new data structures.

With the help of our proposed work is improvement to different data space applications and requests, here we are going to concentrateonly on file search situation, and wehave the granularity of the complete search results to a single file in our personal information system. Might be our technique was enhanced to the flexible query models, where the small amount of data in the files can be returned as files.

Our detailed connection includes

- We introduce IDF-based technique for scoring structure,content and metadata, and a context to assemble the independent dimensions scores to unified multi dimension scores.
- We are inheriting the current top-k query dispensation algorithms and recommend optimizations to increase access to structure dimension index.
- Experimentally we evaluate our scoring framework and denote that our method has expressively will improve searching exactness with the current filtering techniques.
- We strongly determine the efficient of our technique optimization

- Operation on query time dispensation and represents that out technique will improve the effectiveness of query and result in better scalability.

In both IR ranking and databases the conversation on incorporating technologies from two fields with query results on structure based to groupings of content – only searches. Our proposed mechanism makes available a period in this way as they assembled DB-style structure with IR-style info score estimation scores.

II. UNIFIED MULTI-DIMENSIONAL SCORING

In this section, we are going to discuss about the unified context for giving scores to files depends on how close within different query dimensions equal to query conditions. We distinguish three types of scoring dimensions: The data in conditions on the files text related data, metadata for the situations on the file those are related to system information, and structure of the condition for accessing the files those are based directory path.

Here we are showing the directory path and metadata information using a XML document. We are using easy version of XQuery in the key based content conditions to represent structure and metadata consequences.

If any of the file related to more query conditions then we will get a quick and prospective answer for a particular query.

It was allotted with a score for every dimension based on how much it matches to associated query condition. Integrated all these individual scores in to a single unit for ranking of answers.

Our scoring mechanism is worked on clarification of IDF-base scores, as per defined below. For every query condition we will score file that is based on last relaxed form of condition that every file matches. Scoring through all the dimensions is similar IDH-based those allows us to prepare meaning full b combining all the dimensions.

2.1 Scoring Content

For content query conditions we use standard IR relaxation and scoring techniques. Especially we inherit the TF- IDF scoring formulas from a position of the art keyword search tool. These formulas as follows:

$$\text{Score}_c(Q,f) = \sum_{t \in Q} \frac{score_{c,tf}(t,f) X score_{c,idf}(t)}{NormLength(f)} \qquad (1)$$

$$Score_{c,tf}(t,f) = \sqrt{No.times\ t\ occurs\ in\ f} \qquad (2)$$

$$Score_{c,tf}(t,f) = 1 + \log\left(\frac{N}{1+Nt}\right) \qquad (3)$$

Here Q represents query condition for file content, f is the one of scored file, and N specifies the total number of files, N_t denotes the number of files comprises the word t, and NormLength (f) is a common factor that is a f's length function. Reminder that relaxation is an essential part of the above formulas from the score files that comprise a subset of terms in the query conditions

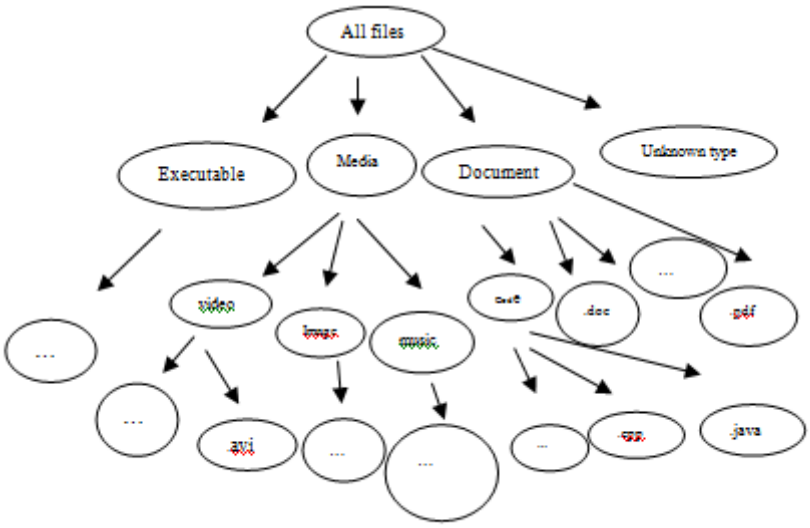

Fig. 1 Fragment of Relaxation DAG for File Type (Extension) Media

2.2 Scoring Metadata

To maintenance scoring we are introducing a confidential relaxation technique for all type of searchable metadata. Relaxation levels for the file types shown in Fig 1, represents as a DAG. In the above figurethe leaf nodes denotes file types. Every internal node denotes as most common file types that is integration of media files and its children files and therefore is a relaxation of their child. A key feature of this hierarchy is containment; means, the set of files matching to node must and should matching or equal to its children nodes.

This specifies that the score of a file matching to more relaxed form of a particular query condition is equal to or less score of matching a file.

Then we say that the metadata condition is matches to a DAS node if in case metadata node range is equals to or contains the query condition. If a specified query condition comprise a type of file "*.cpp" which will match the nodes defining the files type "Code" and file type "Document" so on. The query statementwhich is depends on the file creating time, it will deliberate variant time granularities like date, month and year, a node in the deepest matching path node to the root of the DAG, then it signifies all relaxations that how much we can score for the query statements. Similarly, if all files match to all the nodes in DAG, then it equals to the value of files metadata.

At last, given query Q contains a metadata condition M, with respect to Q the metadata score file f is computed as:

$$scoreMeta(Q,f) = \frac{\log\left(\frac{N}{nFiles(commonAnc(nM,nf))}\right)}{\log(N)} \qquad (4)$$

Here N denotes total number of files, n_m denotes the deepest node that matches with M, Here nf is deepest DAG node matching with the file f, *commonAnc* (x, y) is returns the nearbyprecursor of the node x and y in the relaxation ladder, and nFiles (x) represents return number of files which matches with node x.

The score is comprehensive by log (N), so by this the highest possible score value will give 1.

2.3 Scoring Structure

More number of the users arranges their data by using hierarchical structure. At the time ofsearching for that file he requires to remember the directory path or just rough ordering instead of exact file location. Thus the guess on query condition is required, because it allows the user to with is his partial memory he has an option to search his files even though he doesn't know exact location.

Our structure scoring mechanism is some advanced to this existing work of XML structural query relaxations. Especially node transposal relaxation is presented bellow to handle the passible miss-directive pathname when representingstructure query condition in personal file management system. Let's imagine the query conditions are given as file queries.

The relaxations are:

- **Edge Generalization:** Use of this is to relax a parent-child relationship to antecedent sliding relationship.
- **Path Extension:** Use of this is to extend the path P, every file those are available in the directory have p is answer for the query. Example of extension path P to /a/b will be result as /a/b// *.
- **Node Inversion:** The use of this to change a node to within query path P. To represents possible changes, we are presenting a concept as path is made by group of nodes, whereplace of node is fixed or nodes may be change. The difference is applicable to itsnearby nodes or node group excluding for *.nodes and root. The variation associations' neighbouring nodes or group of nodes as a single node group while providing the relevant order on edges in p.
- **Node Deletion:** This is used to completely delete a node from path, which is appropriate to path query or node group, but it is not applicable for delete the root node and *.nodes.

To delete a node n from path query P.

- If n is a leaf node, then n will dropped from P and n should extended with //*. This is to assurance repression to accurate answer to P in the group of answers to P'

- if in case n is an internal node in the p at that time the parent node (n) and child node n should be connected in the P with //.

2.4 Score Aggregation

We gatheredall above described single dimension scores in to uniformed multi dimension scores to give costumed ranking valuesrelated to multi-dimensional query.

To achieve this we proposed a query vector V_Q structure with exact value is 1for all dimension and for file vector, VF, with respect to Q encompassing of the single dimension scores for a file F. We then process V_F projection on V_Q and length of vector outcome is used as accumulated score file F. With its current form, this is simply a lined grouping of component scores using equal assessing.

III. QUERY PROCESSING

We are getting a current algorithm is recognized as threshold algorithm (TA) to perform query processing. To remove every possible match to the query TA uses a threshold condition, focussed instead of identifying k best answers. It will takeseveral sorted lists as input, each individual sorted list contains system object according to their specific attributes its sorted in ascending order, and vigorously access the sorted lists till the thresh hold extents to find the k best answers.

To fetch individual attribute scores TA reliable on random access. By considering sorted accesses the sorted list declared in the above, desires the files to be return in the descending order based scores for particular dimension. The random access desires any files for particular computation score of any given file. Random access appears when Thresh hold algorithm recognize a file fromidentified list equivalent to some of dimensions, then it require the score of files which are available in all the dimensions to perform the computation its combined scores. To use TA with our arrangement, indexed structures and algorithms requiresupporting both random access and sorted each of our three dimensions.

3.1 Evaluating Content Score

We are using the existing TF-IDF technique to score the informationdimension, this is performing according to describedabove. Random access is supporting overstandard inverted list implementations. Here, for each query time, we could easily watch the time frequency in total file system as well as ina single file. Her we are performing sorted access by observancethe reversed lists in sorted order that is group of files contains the specific term, keep them in to sorted order according their TF scores, and perform normalize the file size.

3.2 Evaluating Metadata Scores

By using an appropriate relaxation DAG index we are implementing sorted access for a metadata condition. First matching content is recognized by the deepest DAG node that matching in given condition. Exact matching conditions are retrieved from N's leaf nearby nodes, the estimated matching nodes are passing up DAG to reflect most approximate matching things. Everyparent node contains larger values than its children node, it represents the matches are done in decreasing order of metadata scores. Equivalent to information dimension, we are using TA algorithm to retrieve the values repeatedly in sorted for esteemed queries that contains multiple metadata conditions.

3.4 Evaluating Structure Score

The structure scores for a particular file depend on matches to a query condition in what directory exactly the file was located. All the files having same structure within in the same directory.

To compute the structure score of a file f in the directory d and structure condition P for a given query, we first required to recognise all the directory paths with comprising of d that match with P.

IV. CONCLUSION

In this paper we are introducing a scoring technique to multi-dimensional queries on personal information systems. Especially, we are describingmetadata information and structure information and we proposed IDF-based scoring mechanism for metadata, content and structure query condition. This constant scoring integrated the individual scoring in to one form. We also require to proposal query processing optimization and indexing structure and the dynamic index construction to efficient valuation of multi-dimension queries. We calculated and implemented scoring frameworks and the query processing techniques. Our valuation shows that our multi-dimensional score aggregation mechanism standbyfeatures of individual scores and comprises significantly improve the exactness of ranking. We also denotes that our directories and optimizations are completed multi-dimensional search is efficient and in daily search use.

REFERENCES

[1] S. Agrawal, S. Chaudhuri, G. Das, and A. Gionis.Automated ranking of database query results.In *Proc.Of the First Biennial Conferenceon Innovative Data Systems Research (CIDR'03)*, 2003.

[2] S. Amer-Yahia, P. Case, T. R¨olleke, J. Shanmugasundaram, andG. Weikum. Report on the DB/IR panel at SIGMOD 2005. *SIGMODRecord*, 34(4), 2005.

[3] S. Amer-Yahia, S. Cho, and D. Srivastava. Tree Pattern RelaxationIn*Proc. Of the Intl. Conference on Extending Database Technology(EDBT)*, 2002.

[4] S. Amer-Yahia, N. Koudas, A. Marian, D. Srivastava, and D. Toman.Structure and Content Scoring for XML.In *Proc.Of the Intl.Conference on Very Large Databases (VLDB)*, 2005.

[5] S. Amer-Yahia, L. V. S. Lakshmanan, and S. Pandit.FleXPath:Flexible Structure and Full-Text Querying for XML. In *Proc.Of theACM Intl. Conference on Management of Data (SIGMOD)*, 2004.

[6] Lucene. http://lucene.apache.org/.

[7] R. A. Baeza-Yates and M. P. Consens.The continued saga of DB-IR integration.In *Proc. Of the Intl. Conference on Very Large Databases (VLDB)*, 2004.

[8] C. M. Bowman, C. Dharap, M. Baruah, B. Camargo, and S. Potti.A File System for Information Management.In *Proc.Of the Intl. Conference on Intelligent Information Management Systems (ISMM)*,1994.

AUTHOR PROFILE

	**Kishore Babu P**is currently pursuing M.Tech in the Department of Computer Science & Engineeringfrom Nalanda Institute of Technology (NIT), siddharth Nagar, Kantepudi(V), Sattenapalli (M), Guntur (D), Andhra Pradesh , Affiliated to JNTU-KAKINADA.
	**T Lavanya**working as Assistant Professor at Nalanda Institute of Technology (NIT), siddharth Nagar, Kantepudi(V), Sattenapalli (M), Guntur (D), Andhra Pradesh , Affiliated to JNTU-KAKINADA.