International Journal of Modern Engineering Research (IJMER)

www.ijmer.com Vol.2, Issue.3, May-June 2012 pp-960-965 ISSN: 2249-6645

Clustering Technique with Potter stemmer and Hypergraph Algorithms for Multi-featured Query Processing

M. Nithya

Assistant Professor-III, Sri Sairam Engineering College

Abstract

In navigational system, it is important to provide a user with flexible and fast query processing without a search failure as well as to find optimal paths guiding the user to a destination. If a user has to repeat submitting a query several times, the user cannot be satisfied with the system. Using this new technique we can extract the correct information within less time effectively. Applications like multimedia databases or enterprise-wide information management systems have to meet the challenge of efficiently retrieving best matching objects from vast collections of data. Here a technique, "Clustering" supported by potter stemmer and hypergraph algorithm for processing multi-feature queries on diverse data sources is presented. Frequent item sets are generated from the groups of items, followed by clusters formed with a hyper graph partitioning scheme. Furthermore its performance is validated by comparing it with typical search techniques.

1. Introduction

In many emerging database applications, such as multimedia retrieval, exploratory data analysis, market basket applications, bioinformatics, and time-series matching, data are usually described by multiple features, each of which is typically high-dimensional. For example, an image may be described by a 16-dimensional color histogram, a 16-dimensional texture histogram, a 32-dimensional shape feature, and a 64-dimensional text feature. To support multifeature queries, we can build a high dimensional index on the feature space obtained from all dimensions of the multiple features. This system implements an efficient way of processing multifeature queries. In this system the indices are built based on a static combination of feature weights [1]. In multifeature query processing, the weightages of features typically are different for different queries and different weightages correspond to different results. In this process a user normally inputs a query with respect to more than one feature. A database normally is a collection of documents will be maintained separately. The user can give the query for searching the documents which are more relevant. The queries can be formulated in standard keyword form. The better solution is employed and the faster response is given for random search of the documents. From the given query the documents are extracted. By reading the full text document unwanted tags are generally eliminated from the documents and also the unwanted subjects are removed and the keywords are extracted. Initially search process is carried out based on the single feature. This could be compared with searching the database based on multifeature query [2], [3]. The most relevant documents are fetched from the database and separately shown. Weightages are applied to the documents by receiving each keyword and finding the number of occurrences of this key word in each document. According to high preferential weightage the documents are fetched accordingly. Thus we can efficiently process the multifeatured query.

2. Existing System

Today in most of the search engines can only use queries rather than web user profiles due to the difficulty of automatically acquiring web user profiles [4]. The simplistic approach of acquiring user profiles is to describe the profiles through term vector spaces (e.g., a set of keywords) by using machine-learning techniques. The main disadvantage of the simplistic approach is the poor interpretation of user profiles to the users. To obtain an explicit specification to the users, user profiles can be represented in some predefined categories. There are two main drawbacks in using these approaches to acquire web user profiles [5]. The first one is that the effectiveness s largely depends on the numbers of labeled training data. However, we may only obtain some positive documents. The second one is that it is hard to distinguish non interesting topics from interesting topics. Here, we develop an ontology mining technique to overcome the above drawbacks.

- Lacking Data Accuracy
- Computational complexity
- More Time Consumption

3. Proposed System

In the proposed system multifeature query processing, the weightages of features typically are different for different queries and different weightages correspond to differ results. In this process a user normally inputs a query with respect to more than one feature. The queries can be formulated in standard keyword form. By reading the full text document unwanted tags are generally eliminated from the documents and also the unwanted subjects are removed and the keywords are extracted. Initially search process is carried out based on the single feature. This could be compared with searching the database based on multifeature query. The most relevant documents are fetched from the database and separately shown.

www.ijmer.com Vol.2, Issue.3, May-June 2012 pp-960-965 ISSN: 2249-6645

- Efficient filtering of data.
- Takes less computational time.
- Data accuracy can be maintained.
- Lightweight.

4. Architecture

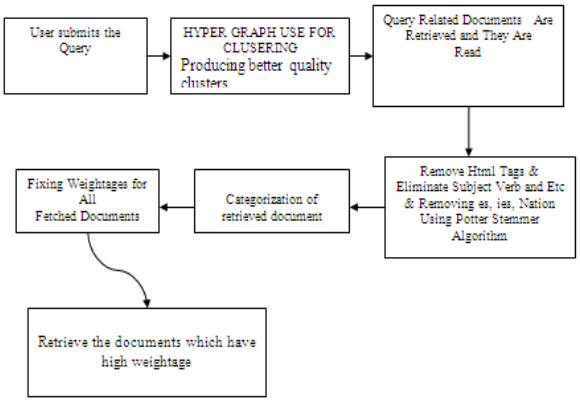

Figure.1 Architecture Design

Figure.1 states the architecture design of the proposed system. It has the following sub modules to get realized as a design.

- User Query
- Keyword Extraction
- Improving Retrieval Effectiveness
- Categorization
- Document clustering
- Mapping Documents

4.1 User Query

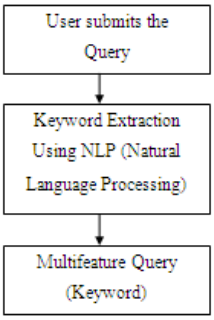

Figure.2 User Query

Figure.2 states the user query module sequence. In this module the user submits the query. The queries can be formulated in standard keyword form. Here the keywords are extracted using Natural Language Processing. Thus the search process is carried out using those keywords extracted and not with the entire standard query.

4.2 Document Clustering

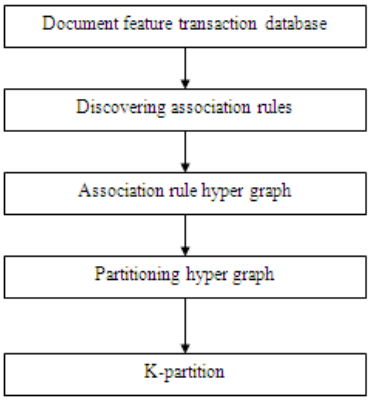

Figure.3 Document Clustering

Figure.3 states the document clustering sequence. Document clustering group all document so that the document in the same group are more similar than ones in other group. Relevant documents tend to be more closely related to each other than to non-relevant document. It founds the association between these documents and sets the partition based on their features found in the document.

- Getting keywords from database.
- Comparing these words with our document needs.
- Grouping on the basis of frequent value.

4.3 Keyword Extraction

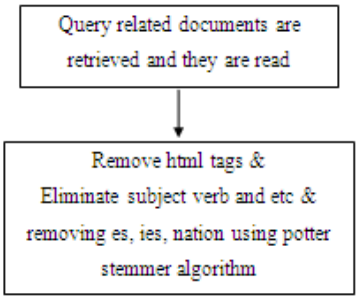

Figure.4 Keyword Extraction

Figure.3 states the keyword extraction sequence. In this module with the given query the document are extracted. By reading the full text document unwanted tags are generally eliminated from the documents and also the unwanted subjects, verb and etc, es, ies are all removed using the Potter stemmer algorithm. The keywords are extracted using Natural Language Processing and stored in the database.

www.ijmer.com Vol.2, Issue.3, May-June 2012 pp-960-965 ISSN: 2249-6645

4.3 Improving Retrieval Effectiveness

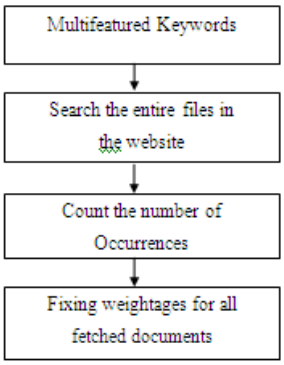

Figure.5 Retrieval Effectiveness

Figure.5 states the sequence of improving retrieval effectiveness. In this module with the extracted keywords the search process is carried out. Normally the search process is carried out by traversing the entire files in all documents. Accordingly the number of occurrences of the keywords in the database is counted. With this count weightages are fixed for all related documents fetched from the database.

4.4 Categorization

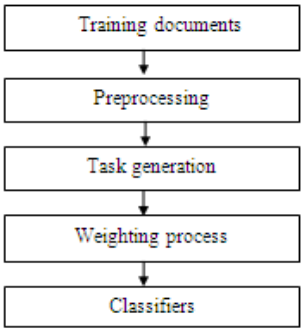

Figure.6 Categorization

Figure.6 states the sequence categorization. Here assigning one or more predefined categories (topics themes) to one document is called as categorization. It generates the task based on the topics evaluated .It consist of a set of training sets and predefined categories. It computes the similarities and assigns weightage for each category and evaluates the scores. Based on these scores it identifies the classifiers.

4.5 Mapping Documents

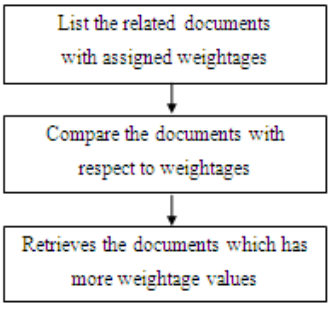

Figure.7 Mapping Documents

Figure.7 states the sequence of mapping documents. In this module all the related documents with assigned weightage values are listed after the search process gets completed. Now comparison is done with respect to the weightage values. Thus the document which has more weightage value is retrieved from the database.

5 Data Flow Diagram

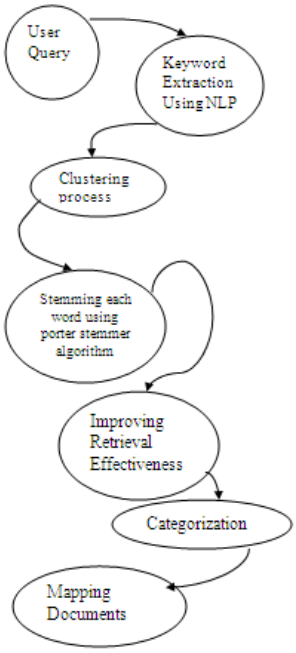

Figure.8 Data Flow Diagram

Figure.8 illustrates the data flow sequence in the proposed model. This model uses the Potter stemmer's algorithm in Keyword extraction sub module and also uses hyper graph in document clustering sub module. Both these algorithms are presented in the subsequent section.

5.1 Potter stemmer Algorithm

Step1:

- Stop words: Words that are discarded from a document representation
  – Function words: a, an, and, as, for, in, of, the, to
  – About 400 words in English.
  – Other frequent words: "Lotus" in Lotus Support db
- Removing stop words makes some queries difficult to satisfy
  – Few queries affected, so little effect on experimental results
  –But, very annoying to people

Step2:

- Group morphological variants:
  Plural: "streets" <=> "street"
  Adverbs: "fully" <=> "full"
  Other inflected word forms: "goes" <=> "go"

www.ijmer.com Vol.2, Issue.3, May-June 2012 pp-960-965 ISSN: 2249-6645

Grouping process is called “conflation”

- Accurate than string matching Current stemming algorithms make mistakes
  Conflating terms manually is difficult, time-consuming.
  Automatic conflation using rules
  Porter Stemmer
  Porter stemming example: “police”, “policy” => “polic”.

Step3:

- Algorithm is based on a set of condition/action rules – old_suffix->new_suffix
- Rules are divided into steps and are examined in sequence

  Step 1a: sses->ss, ies->i, s->NULL
  - caresses -> caress, ponies -> poni, cats -> cat

  Step 1b: if m>0, eed -> ee
  - Agreed -> agree
  - Many implementations available
  - Good performance

5.2 Hypergraph Definition

Hypergraph: *H = (V, E)*
V*:* a set of vertices, here, docs being clustered
E*:* a set of hyperedges which can connect more than two vertices, here, a set of related docs
A weight is assigned to each hyperedge
Each document viewed as an item
Each possible feature word as a transaction

The hypergraph representation

- Represent each document as a vertex item
- Compute all the frequent item-sets, with a given threshold support count
- Represent each frequent set as a hyperedge
- Assign the weight as the average confidence of the essential association rules of the set

6. Realization

The proposed cluster technique is integrated to a typical search engine like YAHOO and the performance of it is compared with normal search techniques.

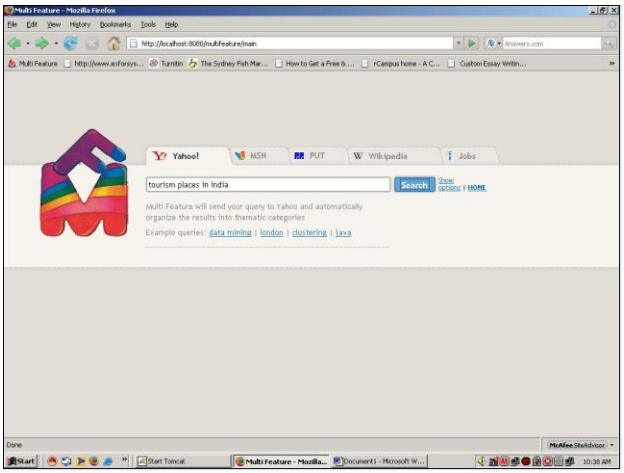

Figure.9 User placing his query

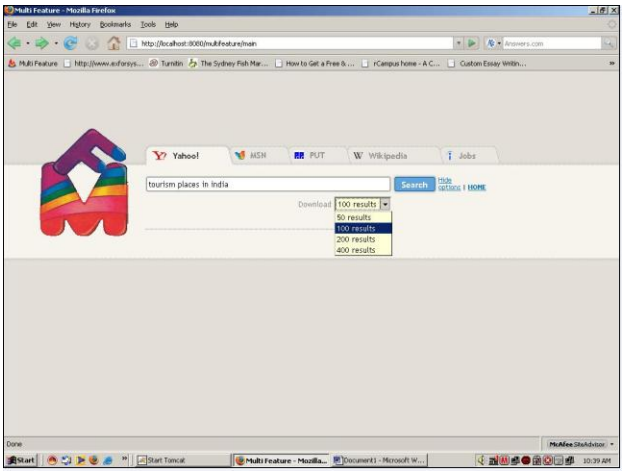

Figure.10 User selecting required search results

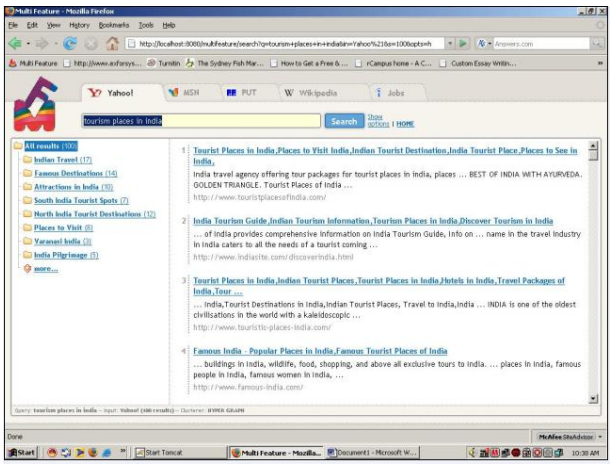

Figure.11 Results for the query

www.ijmer.com Vol.2, Issue.3, May-June 2012 pp-960-965 ISSN: 2249-6645

Figure 9, 10 and 11 gives the normal sequence of searching user query in the yahoo search engine with the proposed technique integrated. In figure 9 the user gives a query. And in figure 10 the user has a option to select number of search results required. Finally in figure 11, the user gets the results displayed.

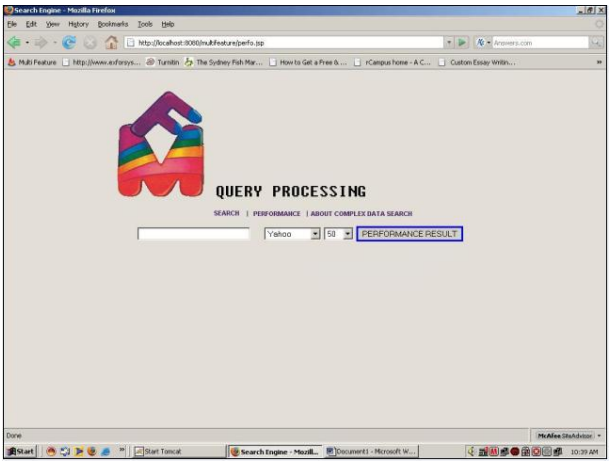

Figure.12 Comparison against typical search technique

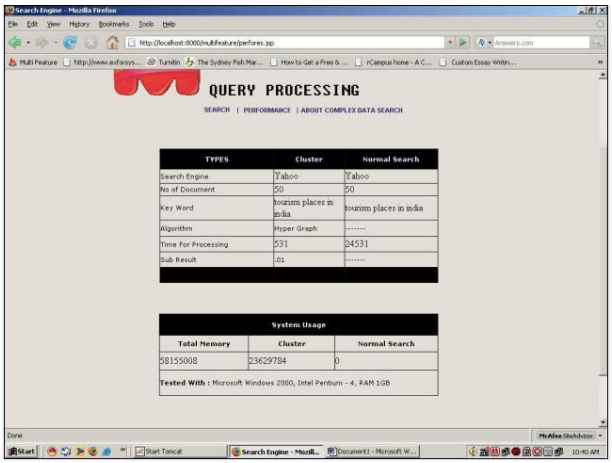

Figure.13 Comparison result

In Figure 12, an option for comparing the performance between the proposed technique and the existing search technique is carried out. Once the performance result button is clicked, the comparison results are displayed as in figure 13. From the comparison it is evident that the proposed cluster technique has superior performance in terms of time required for processing against the typical technique. For an equivalent search, the cluster technique throws result within 531 micro seconds when compared to a long time of 24531 micro seconds.

7. Conclusion

In this paper a new search technique called cluster technique which works on Potter stemmer and hypergraph algorithm is realized. The architecture and subsequently its design are realized. Its performance is compared with typical search techniques and is found that its performance is far superior to them. For an equivalent search, the time taken by this cluster technique is very less compared the existing techniques. Thus it can be concluded that the proposed design can overwhelm the existing techniques by its superior performance. Future scope of improvement will be on the system memory usage which is considerably more using this cluster technique.

8. References

[1]. C. Bohm, S. Berchtold, and D. Keim, "Searching in High- Dimensional Spaces: Index Structures for Improving the Performance of Multimedia Databases," ACM Computing Surveys, vol. 33, no. 3, pp. 322-373, 2001.

[2]. A. deVries, N. Mamoulis, N. Nes, and M. Kersten, "Efficient k-NN Search on Vertically Decomposed Data," Proc. SIGMOD Conf., pp. 322-333, 2002.

[3]. N. Koudas, B. Ooi, H.T. Shen, and A. Tung, "LDC: Enabling Search by Partial Distance in a Hyper-Dimensional Space," Proc. Int'l Conf. Data Eng., 2004.

[4]. Y. Sakurai, M. Yoshikawa, S. Uemura, and H. Kojima, "The A-Tree: An Index Structure for High-Dimensional Spaces Using Relative Approximation," Proc. Conf. Very Large Data Bases, pp. 516-526, 2000.

[5]. R. Weber, H. Schek, and S. Blott, "A Quantitative Analysis and Performance Study for Similarity Search Methods in High Dimensional Spaces," Proc. Conf. Very Large Data Bases, pp. 194-205, 1998.