Computer Engineering and Intelligent Systems www.iiste.org
ISSN 2222-1719 (Paper) ISSN 2222-2863 (Online)

Web-based Text Mining

M. Amarendra

Department of CSE, Sri Sai Aditya Institute of Science & Technology, Surampalem, E.G. District, Andhra Pradesh India. Email: amar.muppalla@gmail.com

R. V. S. Lalitha

Sri Sai Aditya Institute of Science & Technology, Surampalem, E.G. District, Andhra Pradesh, India. Email: lalitha_cse@yahoo.co.in

Abstract
Text mining deals with retrieval of specific information provided by customer search engines. With the massive amount of information that is available on the World Wide Web, text mining provides results in the order of highest relevance to the key words in the query. Text expresses a vast, rich range of information, but encodes this information in a form that is difficult to decipher automatically. For example, it is much more difficult to graphically display textual content than quantitative data. In this paper we describe a method for choosing a subset of the Web, an approach to create a search a flexible service to adopt a new way to generate highly effective results for expert searches. Retrieval of information poses the problem of redundancy in retrieval of same data repeatedly. This paper presents an optimized solution for fast recovery of data and also finds methods for regenerating the queries from the queries posed.
Keywords: Text data mining, Frequent item sets, Query Regeneration, Projected databases, Transaction database

1.Introduction: Applications of text classification technology are becoming widespread. In the defense against spam email, suspect messages are flagged as potential spam and set aside to facilitate batch deletion. News articles are automatically sorted into topic channels, and conditionally routed to individuals based on learned profiles of user interest. In content management, documents are categorized into multi-faceted topic hierarchies for easier searching and browsing. Shopping and auction web sites do the same with short textual item descriptions. User poses queries to the server. This query is evaluated by surfing over the net the information from different nodes and finally relays the result. This evolves increase in network traffic and unnecessary processing. In the existing system, when a query is submitted, it is outputted to all the nodes that contain the information, each time the query is submitted. This increases the query execution time. If we find a method to stream line the information retrieved then search cost reduces. For this to be done, we store queries posed in one database called User Query Database, and the responses in a database called Query Response Database. These databases are to be refreshed periodically so as to get intact with the changes.

Instead of direct submission of query, the query is checked in all the above four phases and then the result is displayed. Maintain two databases UserQuery database for storing queries submitted, Attribute Database for storing attributes information, and finally QueryResponse for storing response information. These three databases are updated frequently to sustain frequent updates. When the query is submitted it is scanned and the information is stored in relevant databases. After the query is processed, the response is also stored in the database. We compare the query submitted with the query database to analyze the response from previous information stored.

2.Related work:
When huge amount of online information is to be searched, then query processing itself takes much amount of time to execute. We need to build systems to create use and reuse the system. Not all text is available on the web. Web information is processed data plus data gathered from the data marts. Statistics and linguistics are combined to create a knowledge database. Previously Greedy transformation techniques are used for partial parsing. Research that exploits patterns in text does so mainly in the service of computational linguistics or information access, rather than for learning about and exploring text collections. Text mining, sometimes alternately referred to as text data mining, roughly equivalent to text analytics, refers to the process of deriving high-quality information from text.

High-quality information is typically derived through the devising of patterns and trends through means such as statistical pattern learning. Text mining usually involves the process of structuring the input text (usually parsing, along with the addition of some derived linguistic features and the removal of others, and subsequent insertion into a database), deriving patterns within the structured data, and finally evaluation and interpretation of the output. 'High quality in text mining usually refers to some combination of relevance, novelty, and interestingness. Typical text mining tasks include text categorization, text clustering, and concept/entity extraction, production of granular taxonomies, sentiment analysis, document summarization, and entity relation modeling. In this paper we discuss how to obtain the retrieve data from the processed response is done by using three phases.

Parsing Query : The query parsing and redirecting response is done by evaluating the query in the following three approaches.
A. QueryMatch approach
B. QuerySum approach
C. QueryMapAttribute approach
D.PatternRecognition process

A. QueryMatch approach: This is the situation where the query is posed is exactly same as that of the query already posed by the previous user. Scan the attributes and operators in the query[2] and store them in the Attribute database. Attribute database maintains the field information from various databases available in the nodes A,B,C,D,E. Strictly speaking, it maintains transaction data that is derived from various tables in the network. Transaction table is maintained for each table in the database, like A_{trans}, B_{trans}, C_{trans}, D_{trans} and E_{trans} .Instead of traversing the query over the entire network, the above algorithm finds appropriate solution to get the output from the previously stored responses (Fig.1). In this paper we use, base station to store centralized database and then the query parameters are distributed over the network.

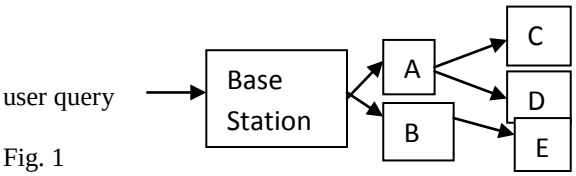
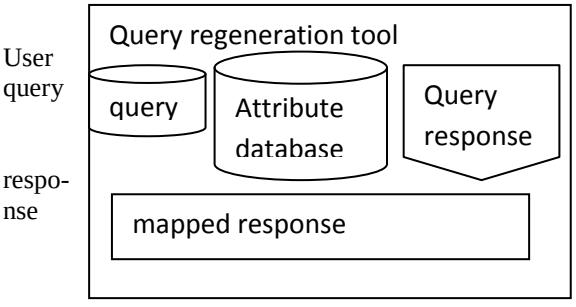


Fig. 1

B. QuerySum approach: This happens if query attributes are equal to the sum of the two query attributes previously stored. This can be analyzed by comparing the attributes from the attribute information database. We are combining the sum of the two queries response which is equal to the response of the current query.

C. QueryMapAttribute approach: If query input is equal to sum of the two queries and some additional attribute information, then obtain the sum of two queries response from the pre-obtained response and process the rest of the attributes over the network. After attributes are processed, map the new response with the previous result and then sent the result back to the user. Store the computed response as a new query response (Fig .2).



D. Pattern Recognition approach: This approach analyzes the query to retrieve prefix matching using fp-tree growth tree. Fp-tree is constructed from the query response (data marts that are created time-to-time) and then text mining sequences bi-grams, n-grams are then retrieved. Bi-grams or di-grams are groups of two written letters, two syllables, or two words, and are very commonly used as the basis for simple statistical analysis of text. They are used in one of the most successful language models for speech recognition. They are a special case of N-gram. The

attribute information in the query is divided into words, characters. Words are the subsets of attribute values. And bi-grams are the prefix values containing attributes.

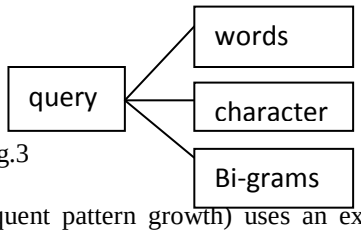


Fig.3

FP-growth (frequent pattern growth) uses an extended prefix-tree (FP-tree) structure to store the database in a compressed form. FP-growth adopts a divide-and-conquer approach to decompose both the mining tasks and the databases. It uses a pattern fragment growth method to avoid the costly process of candidate generation and testing used by Apriori.An fp-tree is trie data structure which is a prefix-tree structure for storing information about frequent patterns. In this tree root is labeled as NULL and set of frequent sub trees are children of root and a frequent header table. The algorithm can be illustrated with the help of the following transaction database. In this database the names and the corresponding frequent pattern names required are stored in the database. By using this database, we construct fp-tree[1] for prefix matches. The support supp(X) of an item set X is defined as the proportion of transactions in the data set which contain the item set. Ordering items in the frequent item sets are done based on the support of that item. First, minimum support is applied to find all frequent item sets in a database.

Tid	Name(s)	Prefix string
1	a,b,d,,t,y	a,b,d
2	a,x, r, k	a
3	b,c,d	b,d
4	l,a,p	a
5	a,d,m	a,d
6	b,d	b,d

Database Fig 4

In the Fig.4 Name(s) containing attribute values retrieved during a particular period. And prefix string is the field containing frequent item sets. We can retrieve items containing the frequent items, if construct fp-tree from prefix trees. The screening is done in the following sequence.

The fp-tree construction step:

1. Scan the database and order the frequent item sets (Fig.4 Name(s) column)
2. Sort the items by support in descending order (Prefix string Fig.4)
3. Create fp-tree with root labeling NULL (Fig.5)
4. Scan the transaction database again to build fp-tree with ordered items[2] (Fig.6 and Fig.7)

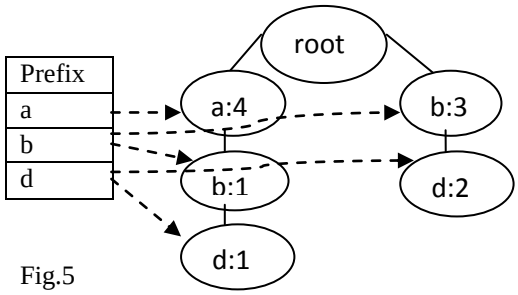


Fig.5

This illustration shows to retrieve subsets containing prefix with ‘a’.

i) Items with a

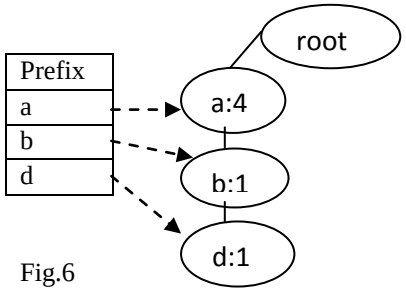


Fig.6

By applying fp-tree algorithm, we can find comfortably the values that are prefixed with a.

This illustration shows to retrieve subsets containing prefix with ‘b’.

ii) Items with b

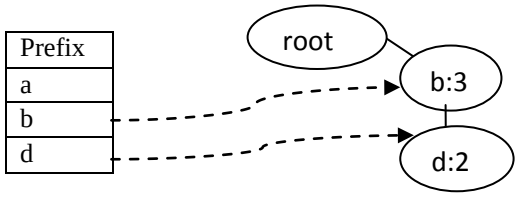


Fig.7

Applying fp-tree algorithm, we can find comfortably the values that are prefixed with b.

3. Query Evaluation Process: We will show how a query is evaluated for the required pattern. The following is the segment of a state chart that shows the idea of parsing the query and storing it in different databases.

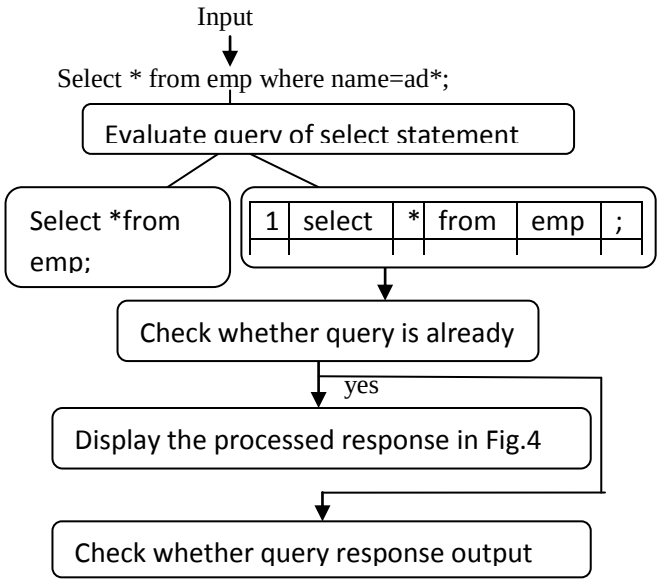


Fig.8

Pattern Recognition: Pattern growth[5] is a method of frequent-pattern mining which does not require candidate generation. Fig.4 gives transaction database for the fp-growth algorithm. Fig.5 fp-tree construct for the transaction database given in Fig.4. Fig.5 is called as projected database[4] which contains at least one frequent item. The general idea is find frequent items and compress them into fp-tree.
Given a sequence $\alpha=\langle e_1,e_2 \ldots e_n\rangle$ (where each e_i corresponds to a frequent event in a given database S)
,a sequence $\beta=\langle e_1',e_2' \ldots e_n'\rangle (m\leq n)$ is called a prefix of α if and only if (1) $e_i'=e_i$ for $(i\leq m-1)$;(2) $e_m' \subseteq e_m$.
Let sequence s=‹ a(bd)› are the prefixes of S.

4. Results: In the previous work, each time the query is submitted, it checks for the database first, and then retrieves data, even though the query is recently submitted and the response is same. Here we are mapping the previously posed queries with the currently posed queries. This involves reading data from buffer and the pre-existing data marts. This is useful, when the information is to be enquired on conditional base approach. Obviously, for exact-match the query is to be submitted for through execution. This gives idea about the database, from the information analyzed previously.

5. Future Extension:

The method describes mining process with no candidate set generation. However it generates projected databases, one for each prefix subsequence. Forming large number of databases recursive is a costly approach. In such a case they have to generate physically. As an extension to this paper we can use pseudo-projection[7], which registers the index(or identifier) of the corresponding sequence and the starting position projected suffix[3] in the sequence instead of performing physical projection. That is, a physical projection of a sequence is replaced by registering a sequence identifier and the projected position index point.

Markov chains and hidden Markov models are probabilistic models [4] in which the probability of a state depends only on that of the previous state. They are particularly useful for the analysis of biological sequence data. This method gives best analysis for training set sequences.

6.Performance issues:
This is adhesively used in web searches. Users have some knowledge about the database information before submitting the query. With this understanding, the analyst can perform more directed analyses providing evidence for making certain conclusions. Text mining can take unstructured data and process it to lead to greater understanding. Text mining offers a
valuable tool to support the process of public input analysis, and knowledge discovery and reporting.

7.Conclusion: This paper presented an extensive comparative study of knowledge discovery of datamining related web text mining. It revealed an outstanding of fast information retrieval from the preprocessed outputs. It gives best chances of obtaining best performance in Information retrieval as well as suffix analysis.

8.References:

1. "A Fp-tree based Approach for Mining All Strongly Correlated Pairs without Candidate Generation", ZengYou He,Xiaofei Xu, Shengchun Deng Department of Computer Science and Engineering Harbian Institute of Technology,92 West Dazhi Street P.O.Box 315 P.R.Chaina 150001
2. "Fp-Tree", Oskar Kohonen
3. "Two Tier Multiple Query Optimization for Sensor Networks", Shelli Xiang Hock Beng Lim Kian-Lee Tan Yongluan Zhou, Department of Computer Science, National University of Singapore {xiangshi, limhb, tankl, zhouyong}@comp.nus.edu.sg
4. "The use of Text Mining to Analyze Public Input",Josh Froelich, Megaputer Intelligence Sergei Ananyan, Megaputer Intelligence David L.Olson, University fo Nebraska
5. "Data Mining Concepts and Techniques " Jiawei Han and Micheline Kamber
6. "Text Classification Using Clustering", Antonia Kyriakopoulou and Theodore kalamBoukis Department of Informatics, Athens University of Economics and Bussiness 76 Patission St.,Athens GR 104.34 {tonia,tzk}@aueb.gr
7. "Text Data Mining Issues, Techniques, and the Relationship to Information Access", Marti A. Hearst July 1997