International Journal of Engineering Research-Online
A Peer Reviewed International Journal
Articles available online http://www.ijoer.in

Vol.3., Issue.2, 2015

RESEARCH ARTICLE

ISSN: 2321-7758

NATURAL LANGUAGE PROCESSING- BASED RELATIONAL KEYWORD SEARCH TECHNIQUE

D.DHAYALAN[1], M.INDUJA[2]

[1]Assistant Professor. [2.]PG Scholar Department of Master of computer Application
Vel Tech High Tech Dr.Rangarajan Dr.Sakunthala Engineering College, Avadi, Chennai

Article Received: 12/04/2015

Article Revised on:16/04/2015

Article Accepted on:18/04/2015

ABSTRACT

The amount of information in this world is increasing rapidly. Keyword search is the most effective in finding and retrieving information online. Unfortunately many systems don't support the familiar keyword search interface that people mostly prefer now –a- days. In the recent fast growing world end users, governments, etc.., use relational database to manage all the information, but relational keyword search in relational database is quite difficult due to data transformations that eliminate redundancy and ensure consistency. The previous approach uses the systematic approach such as Keyword Search with ranking, graph based search and schema based search which standard the relation keyword which minimize the relation Key word finding (ie) it only focus on the closest relations. The existing approach's focus only on search process so the data mining preference is not enough. The proposed system uses Symantec approach which uses Natural Language Processing to find the relation between Keywords which can able to find some distant relation without losing standard. This system also focus on data warehousing which uses stop word removal and stemming algorithm as well as data mining which uses N-Gram algorithm. This approach decrease the search time and produce the accurate result by using relation Keyword search. Annotation based search technique is used to increase the search timing.

Key words: Relational Keyword search, Data Mining, NLP Tool (Natural Language Processing)

INTRODUCTION

In this fast moving world people are very advanced and they don't have some much time in their hand. Now-a-days people prefer to search any information with the help of the relational keyword search techniques [6]. An effective keyword search method for extensive markup Language (XML) is been described [2]. However, we are not aware of any research projects have transitioned from proof- of-concept implemented in Systems. We posit that the existing an empirical based keyword search system [1] gives a detailed analytical of how we can search the information with the help of the keyword that's very useful to all kinds of peoples in this large spread world. The problem of ranking linked data using a ranking framework grouping relationships by their types [3].

Inspite of many significant research papers published in this area, their where not much addressing many important issues related to search accurately [10]. A large number of approaches have been proposed and implemented but among many different research works done in this area, there still remains a severe lack of standardization for system evaluation. To refine the search process rather that getting a accurate results for medical literacy search that's MEDLINE [4].This lack of standardization has resulted in contradictory results from Different evaluations and the numerous discrepancies disorder what advantages are proffered by different approaches. Keyword search is ubiquitions way for the presents a more efficient index structure, the Generalized Inverted Index (Ginix) [5].

A large portion of the deep web is predefined database based, i.e., for most of the search engines, data is been encoded in displayed result page comes from the mentioned structured database. Such a type of search engine is been referred as Web databases (WDB) [7]. A typical result page returned from a WDB has multiple search result records (SRRs). Each SRR contains multiple data units each of which describes one aspect of a real-world entity. The ubiquitous search text box has transformed the way people interact with information. Nearly half of all Internet users use a search engine daily [8], performing in excess of 4 billion searches [9].

MATERIALS AND METHODS

Overview: This system is to use Symantec approach which uses Natural Language Processing (NLP Tool) to find the relation between Keywords which can able to find some distant relation without losing standard. This system also focus on data warehousing which uses stop word removal and stemming algorithm as well as data mining which uses N-Gram algorithm. This approach decrease the search time and produce the accurate result by using relation Keyword search. Annotation based search technique is used to increase the search timing.

This system extending the keyword search pattern to relational data has been an active area of research within the database and Information Retrieval (IR) community. This lack of standardization has resulted in contradictory results from Different evaluations and the numerous discrepancies disorder what advantages are proffered by different approaches. The issues in the existing system, searching wouldn't be in ranking based so the execution time consumed by the user will be more. If a user wanted to search any technical word they wouldn't get much accurate result to overcome that issues we have introduced NLP- based relational keyword search technique.

In our proposed system data warehousing and data mining is been used under Data Warehouse (DW) there are two algorithms been used they are stop word removal which will be removing the non-meaningful words like a, an, the, of, is, are etc…. The next algorithm used under DW is stemming algorithm or stemmers which will remove the suffix such as ions, ing, ion, ed, ive, etc…. In Data Mining (DM) N-Gram algorithm is been used N-Grams is a word prediction algorithm using probabilistic methods to predict next word after observing N-1 words. Therefore, computing the probability of the next word is closely related to computing the probability of a sequence of words.

In this paper we also added annotation by adding the annotation based search technique decreases the search timing as well as increase the search accuracy. We use different type of annotation in our paper that's Table Annotator (TA) with Table Annotator can transform your database schema into an easy-to-read Word document. Query-Based Annotator (QA) text retrieval consists of using textual annotations for obtaining results from a given annotated collection; the retrieved images should be relevant to certain user information needs (queries). Schema Value Annotator

(SA) schema is to help instruct annotators about which words to include in a span when annotating disorders, signs/symptoms, diseases in text. Frequency-Based Annotator (FA) a frequency annotator is performed whenever there is frequent accidence of process or data. Common Knowledge Annotator (CA) some data units on the result page are self-explanatory because of the common knowledge shared by human beings.

Architecture: In the below (Figure 1) overall architecture diagram it clearly represent the functions from the starting to the ending stage in the research paper.

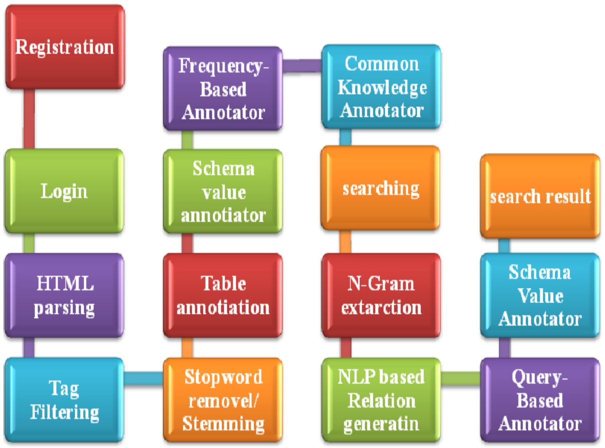

Fig. 1: Overall architecture diagram

Previous approach: A large number of approaches have been proposed and implemented but among many different research works done in this area, there still remains a severe lack of standardization for system evaluation. This system extending the keyword search pattern to relational data has been an active area of research within the database and information retrieval (IR) community. Standardization problem in many research works has resulted in contradictory results from Different evaluations and the numerous discrepancies disorder what advantages are proffered by different approaches.

Disadvantages of previous approach:

- Keyword Search without ranking.
- Execution time is more.

Proposed system: This project uses NLP and explores the relationship between execution time and factors varied in previous evaluations. These results indicate that many existing search techniques do not provide acceptable performance for realistic retrieval tasks. This analysis indicates that these factors have relatively little impact on performance. This work confirms previous claims regarding the unacceptable performance of these systems and underscores the need for standardization as exemplified by the IR community when evaluating these retrieval systems. The proposed system uses Symantec approach which uses Natural Language Processing to find the relation between Keywords which can able to find some distant relation without losing standard. By adding the annotation based search technique decreases the search timing as well as increase the search accuracy.

Advantages of proposed system:

- Keyword Search with NLP.
- Execution Time consumption is less.
- The length of the files and the execution time of each and system can be viewed.
- Fast and accurate searching process

Algorithm used:

- Stop word Removal Algorithm
- Stemming Algorithm
- N-gram Algorithm
- Natural Language Processing (NLP Tool)

List of modules:

1. HTML / Document parsing
2. Term pre-processing
3. Table, Query Based Annotator
4. N-gram
5. Schema Value, Frequency-Based, Common Knowledge
6. NLP – Based Systems Module.

Modules

1. Html/document parsing: It parses terms from input website. In this process first the html page is downloaded and is stored as html documents then the html document is spited in to pieces according to their tags and are grouped then the values of each tag is extracted now we have the values of each tag separately.

It parses terms from input document. The HTML parser reads the content of a web page into character sequences, and then marks the blocks of HTML tags and the blocks of text content. At this stage, the HTML parser uses a character encoding scheme to encode the text. Also any input text document can be read by the parser in a similar way. The two fundamental use-cases that are handled by the parser are extraction and transformation. While prior versions concentrated on data extraction from web pages, Version 1.4 of the HTML Parser has a regular improvements in the area of transforming web pages, with simple and easy useable tag for creating and editing, and strictly to HTML() method output.

2. Term pre-processing: In this phase, various techniques like stemming and stop-Word removal are applied for the reduction of terms. First the stop word removal process take place in which non-meaningful words such as “the, and, are ect” are removes then we go for stemming process in which prefix and suffix of the words are removed thus the size of storage and search time is reduced.

Algorithm: STOP WORD REMOVAL

Input: *an arbitrary stop word dictionary T, interface ‘t’.*

Output: *resultant set of words T.*

```
1.  T ′ <— empty
2.  W= set of all words in the search domain.
3.  Pick a word ω from W.
4.  Check the stop word constraints for ω with
    interface‘t’.
5.  If no stop word is violated then
    T′ <— T′ U {w}.
6.  Go to 3 till nth word in W.
7.  Return T′.
```

Stemming process:

A stemming algorithm is a process of linguistic normalization, in which variant forms of a word are reduced to a common form. There are several approaches to stemming. One way to do stemming is to store a table of all index terms and their stems. For example:

erm	Stem
Engineering	Engineer
Engineered	Engineer
Engineer	Engineer

Terms from queries and indexes could then be stemmed via table lookup. Using a B-tree or hash table, such lookups would be very fast. Porter Stemming Algorithm is one of the most popular stemming algorithms. It takes a list of suffixes and the criterion during which a suffix can be removed. It is simple, efficient and fast.

3. Table, query based annotator

3.1 Table Annotator (TA): With Table Annotator can transform your database schema into an easy-to-read Word document. The following information for each table in our database: Primary Keys, Field Information (type, size, defaults, nullable), Indexes, Check Constraints, and Foreign Keys. The program also lets you annotate your tables and fields. This information is stored in tables in the database you are documenting and can become part of your Word document. When you create your Word document you can select any or all of the tables in your database. You can also organize your tables in titled groups.

Our Table Annotator works as follows: In the first step, it identifies all the column headers of the table. As a next step, for each SRR, it takes each and every data unit in a cell and selects the column header whose area (determined by coordinates) has the maximum vertical overlapping (i.e., based on the x-axis) with the cell. That particular unit is then assigned with that column header and labeled by the header text A (actually by its corresponding global name gn(A)). The remaining data units are processed similarly.

3.2 Query-based Annotator (QA): Text retrieval consists of using textual annotations for obtaining results from a given annotated collection; the retrieved images should be relevant to certain user information needs (queries). Under this approach image annotations and queries are considered as small text-documents that are to be compared. Commonly, a measure based on word matching is used for determining similarity between query and annotations. The documents that are more similar to the query are returned. This is the predominant approach for text retrieval.

Query-based Annotator process as follows: Given a query with a set of query terms submitted against an attribute A on the local search interface, first find the group that has the largest total occurrences of these query terms and then assign gn(A) as the label to the group.

4. N-Gram

4.1 N-Gram indexing – A common part of such a framework is n-gram indexing in which the resultant data is indexed according to search query.

Algorithm: N-Gram Indexing

```
Input : docs, a document collection
Input : n, the length of the n-gram
Output: index, an index including extracted n-gram types with frequencies.
  1) Index Type index:
  2) index .Create();
  3) while docs. Next n gram (n) do
  4) index . Insert(Current N gram());
  5) end
  6) index. Close();
```

4.2 N-Gram building and pre-processing – It creates an N-Gram as a sequence of n terms. Sometimes, N-grams are not shared by text units (sentences or paragraphs).

4.3 N-Gram extraction – The main goal of this phase is to remove duplicate n-grams. The result of this phase is a collection of n-gram types with the frequency enclosed to each type.

ALGORITHM: The basic algorithm of the N-Gram extraction

When a collection fits in the main memory, sorting-based algorithms provide the following time complexity O(Nn + Nn logn Nn), the index-based method provides O(Nn × (logC Tn +1) × log2 C) and O(Nn × k) for the B+-tree and Hash table, respectively, where Tn << Nn and n << C. Moreover, the index-based method provides more efficient space complexity, since it handles a lower count of n-grams (Tn instead of Nn). The results of our experiments described in Section V confirm this theoretical model.

In an n-gram model, the probability $p(\omega_1,\ldots,\omega_m)_y$ *of observing the sentence* $\omega_1,\ldots,\omega_{mis}$ *approximated as*

$$P(\omega_1,..,\omega_m)=\prod_{i=1}^{m} P(\omega_i|\omega_1,..,\omega_{i-1})\approx\prod_{i=1}^{m} P(\omega_{i|}\omega_{i-(n-1)},..,\omega_{i-1}) \quad (1)$$

Here, it is assumed that the probability of observing the i^{th} *word* w_i *in the context history of the preceding i-1 words can be approximated by the probability of observing it in the shortened context history of the preceding n-1 words (*n^{th} *order Markov Property).*

The conditional probability can be calculated from n-gram frequency counts:

$$P(\omega_i|\omega_{i-(n-1)},\ldots,\omega_{i-1})=\frac{\mathrm{Cout}(\omega i-(n-1),..,\omega i-1,\omega i)}{\mathrm{Cout}(\omega i-(n-1),....,\omega i-1)} \quad (2)$$

5. Schema value, frequency-based, common knowledge

5.1 Schema value Annotator (SA): Schema is to help instruct annotators about which words to include in a span when annotating disorders, signs/symptoms, diseases in text. Many attributes on a search interface have predefined values on the given interface. For example, the attribute professional may have a set of predefined values (i.e., professional) in the select list. More attributes in the IIS tend to have predefined values and these attributes are likely to have more such values than those which are present in the LISs, because when attributes from many different interfaces are integrated, their values are also combined. Our schema value annotator utilizes the combined value set to perform annotation.

The schema value annotator first identifies the attribute Aj that has the highest matching score among all attributes and then uses gn(Aj) to annotate the group Gi. Note that multiply the above sum of terms by the number of nonzero similarities is to give preference to attributes that have more matches (i.e., having nonzero similarities) over those that have fewer matches. This is found to be very effective in improving the retrieval effectiveness of combination systems in information retrieval.

5.2Frequency-Based Annotator (FA): A frequency annotator is performed whenever there are frequent accidence of process or data. “Our Pricess” appears in the three records and the followed pricess values are all different in the records. In the other way, the adjacent units have different occurrence frequencies. As argued in the previous research works done, the data units with the higher frequency are liable to be attribute names, as portion of the template program for generating the records, while the data units with the lower frequency most likely to come from databases as an enclosed values. Following this argument, “Our Pricess” can be recognized as the label of the value immediately following it. The process described in this example is widely noticeable on result pages returned by many WDBs and our frequency-based annotator is designed to exploit this process. Consider a group Gi whose data units have a lower frequency. The frequency-based annotator intends to find common preceding units shared by all the data units of the group Gi. This can be easily conducted by following their preceding chains recursively until the encountered data units are different. All found preceding units are concatenated to form the label for the group.

5.3 Common Knowledge Annotator (CA): Some data units on the result page are self-explanatory because of the common knowledge shared by human beings. For example, "available stock" and "unavailable stock" occur in many SRRs from e-commerce sites. The end users understand that it's about the availability of the product because this is common knowledge. So our common knowledge annotator tries to exploit this situation by using some predefined common concepts.

6. NLP-Based systems module: NLP-based approaches support keyword search over relational databases via direct execution of SQL commands. These techniques model the relational NLP as a graph where edges denote relationships between tables. The database's full text indices identify all tuples that contain search terms, and a join expression is created for each possible relationship between these tuples. DISCOVER creates a set of tuples for each subset of search terms in the database relations. A candidate network is a tree of tuple sets where edges correspond to relationships in the database NLP. DISCOVER enumerates candidate networks using a breadth-first algorithm but limits the maximum size to ensure efficient enumeration. A smaller size improves performance but risks missing results. DISCOVER creates a join expression for each candidate network, executes the join expression against the underlying database to identify results, and ranks these results by the number of joins.

RESULTS AND DISCUSSION

In the existing system the execution time, data accuracy, searching and ranking of the search results are very low so as I have been discussed in the below graph. In the figure 3 I have show cased the comparing the results of both the previous and proposed approaches.

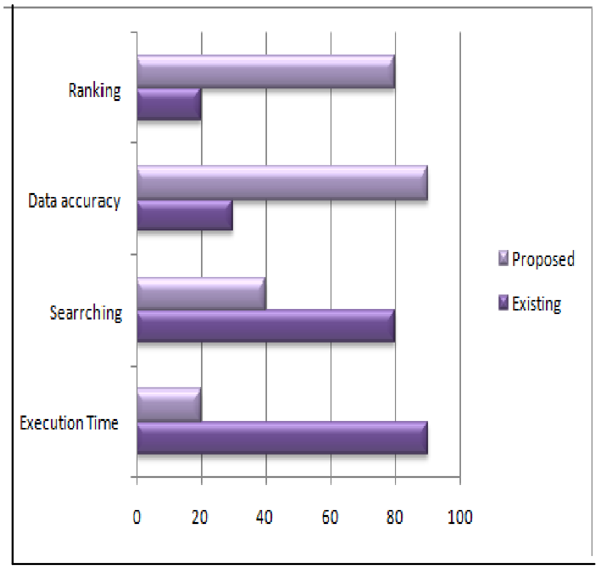

Fig. 2: Stimulation result of the system

In the above stated fig 2 we have showed the ranking difference, data accuracy of the given data searching speed and the execution time consumption.

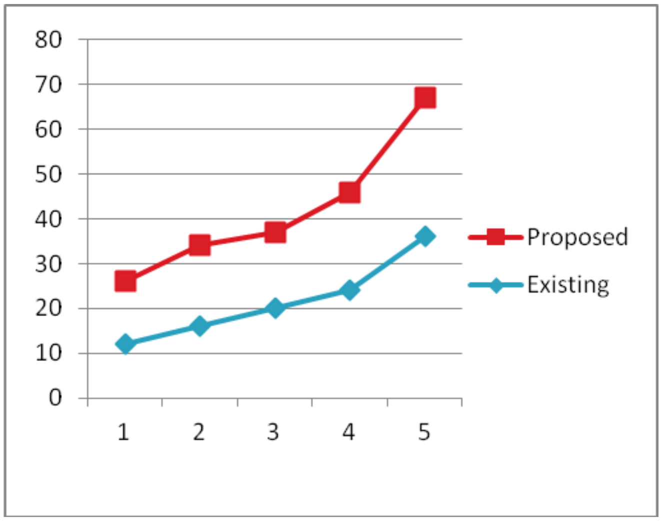

Fig. 3: Comparison of Existing and Proposed Approaches

The range of the existing and proposed approaches as been debited on the above shown graph. The graph shown here is clearly says the use of proposed system is higher than the existing.

CONCLUSION

Unlike many of the evaluations reported in the literature, ours is designed to investigate not the underlying algorithms but the overall, end-to-end performance of these retrieval systems.

Hence, we favor a realistic query workload instead of a larger workload with queries that are unlikely to be representative (e.g., queries created by randomly selecting terms from the dataset).

Overall, the performance of existing relational keyword search systems is somewhat disappointing, particularly with regard to the number of queries completed successfully in our query workload. This system uses Natural Language Processing to find the relation between Keywords which can able to find some distant relation without losing standard. To make the searching faster and accurate annotation based search technique is added. This system also focus on data warehousing which uses stop word removal and stemming algorithm as well as data mining which uses N-Gram algorithm. This approach decrease the search time and produce the accurate result by using relation Keyword search.

REFERENCES

[1]. Joel Coffman, Alfred C. Weaver, 2014 "An Empirical Performance Evaluation Keyword search Systems" University of Virginia Department of Computer Science Technical Report CS-2011-07 IEEE Transactions on Knowledge and Data Engineering, (Volume: 26 , Issue: 1).

[2]. LI Guoliang, FENG Jianhua ,ZHOU Lizhu, February 2009 "Keyword Searches in Data-Centric XML Documents Using Tree Partitioning" TSINGHUA SCIENCE AND TECHNOLOGY ISSN 1007-0214 02/21 , Volume 14, Number 1, pp 7-18.

[3]. ZHANG Jing, MA Chune, ZHAO Chenting, ZHANG Jun, YI Li, MAO Xinsheng, 2010," A Novel Ranking Framework for Linked Data from Relational Databases" TSINGHUA SCIENCE AND TECHNOLOGY ISSNll1007-0214ll04/14, Volume 15, Number 6, pp 642-649.

[4]. WANG Yan, WANG Cong, ZENG Yi, HUANG Zhisheng , Vassil Momtchev, Bo Andersson, REN Xu, ZHONG Ning ,December 2010," Normalized MEDLINE Distance in Context-Aware Life Science Literature Searches", TSINGHUA SCIENCE AND TECHNOLOGY ISSNll1007-0214ll12/14 Volume 15, Number 6, pp709-715.

[5]. Hao Wu_, Guoliang Li, and Lizhu Zhou, February 2013, Ginix: Generalized Inverted Index for Keyword Search", TSINGHUA SCIENCE AND TECHNOLOGY ISSNll1007-0214ll10/12 Volume 18, Number 1, pp77-87.

[6]. Y. Chen, W. Wang, Z. Liu, and X. Lin, June 2009 "Keyword Search on Structured and Semi-Structured Data," in Proceedings of the 35th SIGMOD International Conference on Management of Data, ser. SIGMOD ', pp. 1005–1010.

[7]. J.Coffman and A.C.Weaver, October 2010 "A Framework for Evaluating Database Keyword Search Strategies," in Proceedings of the 19th ACM International Conference on Information and Knowledge Management, ser. CIKM '10, pp. 729–738.

[8]. D. Fallows, "Search Engine Use," Pew Internet and American Life Project, Tech. Rep., August 2008, http://www.pewinternet.org/Reports/2008/Search-Engine-Use.aspx

[9]. "Global Search Market Grows 46 Percent in 2009," http://www.comscore.com/Press Events/Press Releases/2010/1/ Global Search Market Grows 46 Percent in 2009, January 2010.

[10]. H. He, W. Meng, C. Yu, and Z. Wu, September 2004 "Automatic Integration of Web Search Interfaces with WISE-Integrator," VLDB J., vol. 13, no. 3, pp. 256-273.