"THE QUERY CONSTRUCTION INTERFACE FOR ANALOGOUS INFORMATION RETRIEVAL BASED ON KEYWORD SEARCH"

[1]**RADHIKA S. MUNDHADA**
Student, PLITMS, Buldana, India
radhikamundhada5@gmail.com

[2]**DR. P. M. JAWANDHIYA**
Principal, PLITMS, Buldhana, India
pmjawandhiya@gmail.com

ABSTRACT: *Today's web search engines, such as Google, MSN, yahoo, and Amazon are generally work on a keyword search interface and a variety of statistical methods to catch users' information needs. This query construction approach has been highly successful, as it is proven to be very intuitive and easy even for native web users. However, keyword search lacks the expressiveness to make use of the rich semantics in the Semantic Web. The Keyword Search approach makes it easy and the freely usable at the price of query expressiveness. This paper presents IQP; a novel approach to bridge the gap between usability of keyword search and expressiveness of database queries. IQP enables a user to start with an arbitrary keyword query and incrementally refine it into a structured query through an interactive interface.*

Keywords**:** IQP Incremental Query Plan, Semantic Web

1. INTRODUCTION

Keyword search, originally developed for retrieving unstructured documents, is an intuitive and convenient interface for accessing unstructured data. However, since keyword search ignores most structured information, it may fail to understand the user's intent and to identify the appropriate information. On the other hand, database queries allow users to exploit available structures to achieve more precise queries and corresponding result sets. However, using a database system requires adequate knowledge of the database schema and proficiency of the query language, making data access a difficult task for unskilled users. Even for professionals, creating a database query that exactly matches the existing schema can be a laborious and error-prone process, especially when accessing a large and complex database. Today's heterogeneous data management environments demand search interfaces that are not only sufficiently expressive to exploit structured queries, but also as intuitive and easy to use as keyword search. Structured queries are a powerful tool to precisely describe a user's informational need and retrieve the intended information from a database. This task requires exact knowledge of the database schema as well as proficiency in a query language, which are typically beyond the expertise of end users.

For example, if the majority of users who issued the keyword query "London" were interested in a city guide of London, the results referring to Jack London as a book author will receive a low rank. If a ranking function fails to place the user intended structured query within the top results, the user will need to examine all interpretations prior to the intended one. This process is tedious and error prone.

2. LITERATURE REVIEW & RELATED WORK

Title of Paper	Advantages	Remark / Drawbacks / Future Scope
1.Learn from Web Search Logs to Organize Search ResultACM,2007	Learnt "interesting aspects" of a topic from Web search logs & organized search results accordingly	Their method does not work if we try to discover user search goals of one single query
2.Using Implicit Feedback and Query Patterns to Organize Web Documents ACM, 2008	Presented a new document representation model based on implicit user feedback obtained from search engine queries	Their method does not work if we try to discover user search goals of one single query in the query cluster rather than a cluster of similar queries
3.Incremental Query Processing Based on Top K Results IJDCST 2016	Supporting efficient top –k processing in database system is a relatively recent and active line of research.	They should avoid maintaining information about objects that they know they never be included in the result.
4.Natural Language Interfaces to Databases – An Introduction	The information stored in a database has traditionally been achieved using formal query languages, such as Sql	NLIDBS is that the system's linguistic capabilities are not obvious to the user

5. A Novel Approach for Keyword based Incremental Query Construction in Data Base Schemes IJCSIET	The experimental results show that IQP is highly helpful when user intended structured queries cannot be found within the top ranked results.	Their system is implemented in3-tier architecture, consisting of User Interface, Server and Database
6.A survey on query and keyword search in database IJSETR	For searching the information from the database is analyzed, instead of using structured queries the keyword is used with incremental query construction	By using keyword search the results are produced but it may irrelevant to the user's need, so with keywords the incremental query construction is discussed.
7.ESTER: Efficient Search on Text, Entities, and Relations	Search engine for combined full-text and ontology search,	Looking forward to standard entity ranking Benchmarks.
8.DBXplorer: A System for Keyword-Based Search over Relational Databases	DBXplorer has been implemented using a commercial relational database and web server allows users to interact via a browser front end.	Alternative to keyword search is structured search where users direct their search by browsing Classification hierarchies.
9.A Probabilistic Scheme for Keyword-Based Incremental Query Construction	IQP enables a user to start with an arbitrary keyword query and incrementally refine it into a structured query through an interactive interface.	This paper presents the detailed design of IQP, demonstrates its effectiveness and scalability through experiments over real-world data and a user study.
10.From Keywords to Semantic Queries Incremental Query Construction on the Semantic Web	Xuan introduced QUICK, a system for guiding users in constructing Semantic queries from keywords. A user starts with an arbitrary keyword query & incrementally transforms it into the intended semantic query.	The paper has drawback in performance of QUICK which can be further improved and extended in usability as well as efficiency.

3. PROPOSED WORK

A structured query for the Semantic Web is composed of multiple concepts, properties and literals. The construction process of the structured query can be modeled as a hierarchy of query components. At the bottom of the hierarchy is the smallest component, where each is comprised of a single concept, a single property and a single keyword. The higher is the hierarchy more complex is the query components. Structured query lets users start with the smallest query components, and gradually evolve them into larger query components by climbing up the query hierarchy. For example, to search for the movie "Random Hearts", a user might issue a keyword query "random crash Alcee". For each of the terms, structured query provides a list of term-property combinations. For example, the user can specify whether "Alcee" should appear in the actor name, character name or movie title.

After the user specifies some basic query components, the system offers larger components that contain the selected smaller ones. For instance, after the user specifies the character name "Alcee" and movie title "Random", the system can suggest the query component that connects these two term-property combination using the "actsin" property, as shown in the middle left of Figure afterwards. The user specifies that "crash" should appear in the plot-text, and the system can suggest the query component at the top, which is already the complete structured query required by the user. Usually, a user does not need to go through the complete construction process.

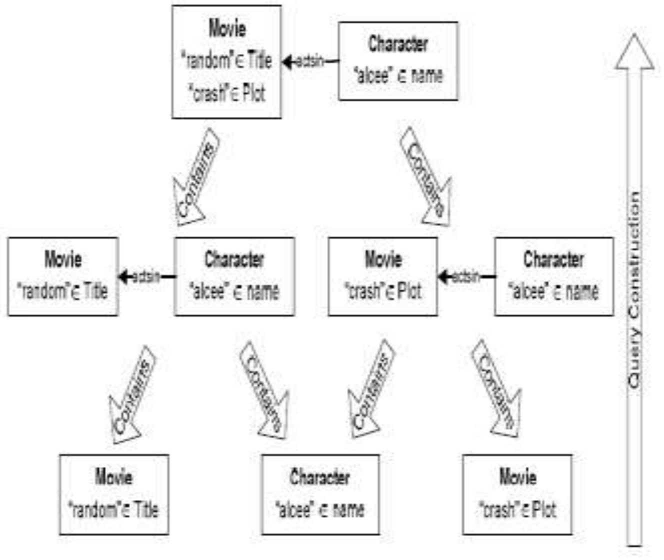


Figure 1: Structure of Query Construction

4. DESIRED IMPLICATIONS

Keyword search approaches gives the various modified results by applying various techniques such as,

1) The working of query construction program will work on the API approach and it gives to the NLP query language and by using this NLP language the keyword entered by a user get linked with the another query and this query is modified by the new keyword.

2) User first entered the query to search engine as in any form to request by user this search engine results are fetched by the API Application program Interface. This API helps to user for searching the results and then search click by user in search results.

3) After the clicking by user on search engine results gives to the NLP Natural Language Programming is applied to the search results to get the modified keyword.

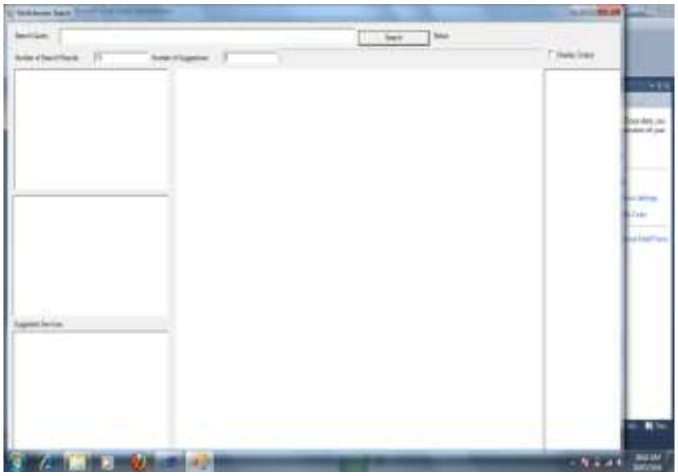
4) This modified keyword gets linked with the query which is given to the user from the search results and this query is modified by new keyword.
5) This modified keyword is now click by user and this modified keyword is now shown to user .for example user entered “apple green” to search engine the this time search engine reads all possible datasets which are in the database and this results to all possible query which contains an keyword “Apple”, such as “Apple green”, ”Apple Eye”, “Apple Company”, ”Apple Image” and so on .
6) This time the search engine gives lots of query formation which contain an keyword apple and the user have to click on the any one of the selected query such as “Apple Green” and this keyword get linked with the new keyword and the search engine now works on the selected query and this selected query then gives to search API.

7) The search API is applied to NLP and the modified keyword which contains various Adjectives, Nouns, Pronouns, and get sorted by those NLP language and the database is updated to get the requested output.

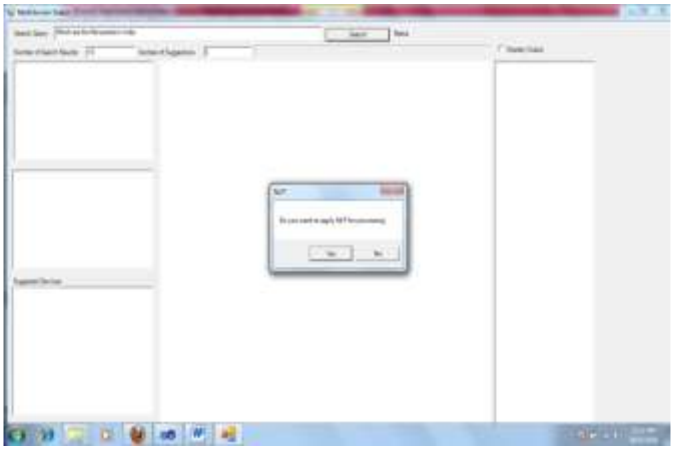
8) The process is continues in loop up to the user don’t get satisfied with the requested result. This is the mechanism which helps user to get satisfied in all manners in all drastic ways.

5. EXPERIMENTAL RESULTS AND DISCUSSIONS

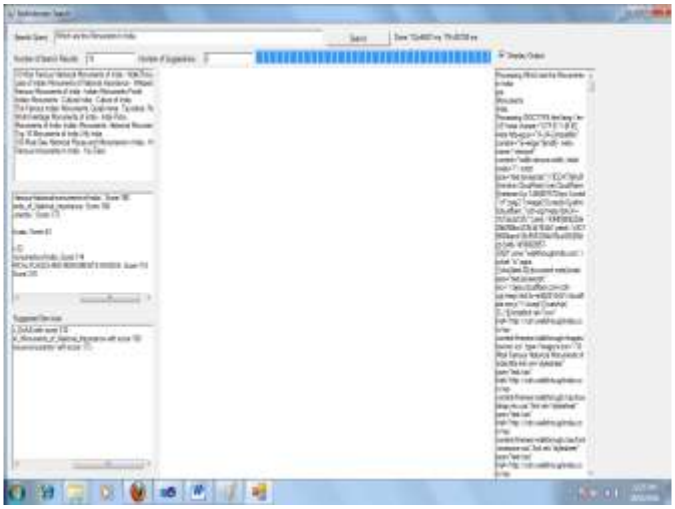
Screenshot 1: Multi-domain Search Text



Screenshot 2: Enter Search Text



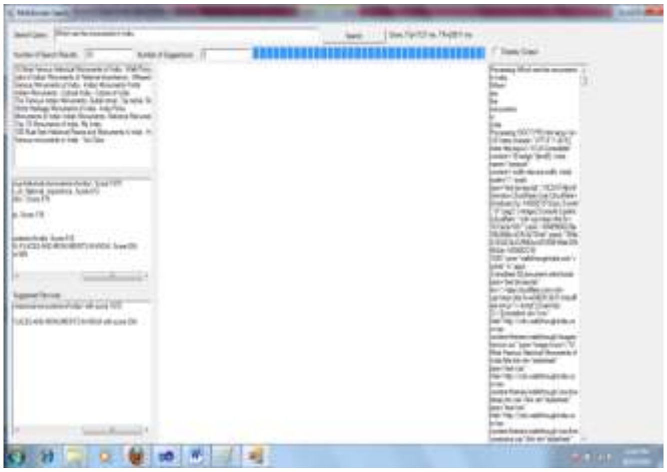
Screenshot 3: Web Page Score by NLP approach



Screenshot 4: Query Searching Text without NLP

Lists of Indian Monuments of National Importance

Screenshot 5: Web page score without applying NLP



Lists of Indian Monuments of National Importance

5.1 Performance Evaluation

Performance evaluation of the proposed approach and the search engine is done based on Precision measure.
Precision measure is calculated based on the following formula.

$$\text{Precision} = \frac{tp}{tp + fp}$$

Where,

tp – True Positive (Correct result)

fp – False Positive (Unexpected Result)

Table represents the matching of manual ranking against proposed approach ranking and search engine ranking. Document SR1, 2, 5, 7, 8, 9 represents the mismatching of manual ranking against content based ranking and documents SR6, 9 represents the mismatching of manual ranking against content and usage based ranking.

Different Methods	Tp	fp	Precision
Search Engine Ranking	2	8	0.2
Content Based Ranking	4	6	0.4
Content & Usage Based Ranking	8	2	0.8

Table 1: Precision Measure

From the table 1, it is understood that precision of the search-engine is 0.2, precision of content mining is 0.4 and precision of combine approach of content and usage mining is 0.8 out of 1. The results of the performance measure are plotted in Fig

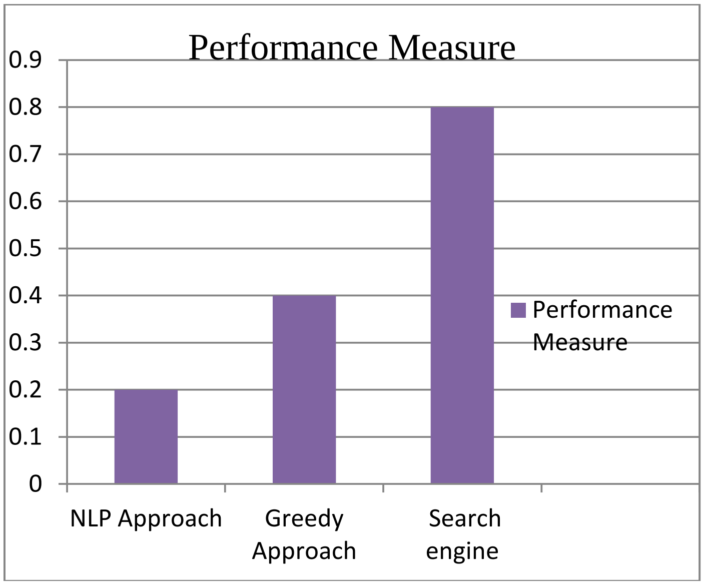


Figure 5.1 Performance Measure

6. REFERENCES

[1] Barbara Poblete, Ricardo Baeza-Yates, "Using Implicit Feedback and Query Patterns to Organize Web Documents" , WWW 2008 / Refereed Track: Data Mining - Log Analysis April 21-25, 2008.
[2] S.Durga Malleswari1, K.Kriran Kumar2," Incremental Query Processing Based on Top K Results" IJDCST @October Issue- V-1, I-6, SW-16

[3] Elena Demidova, Xuan Zhou, and Wolfgang Nejdl, Member, IEEE Computer Society "A Probabilistic Scheme for Keyword-Based Incremental Query Construction ", IEEE transactions on knowledge and data engineering, vol.24, no.3, march 2012.

[4] K.R.N.Sandeep, Ch.Madhavi Latha, "A Novel Approach for Keyword based Incremental Query Construction in Data Base Schemes", IJCSIET--International Journal of Computer Science information and Engg., Technologies ISSN 2277-4408, 01112014-022

[5] Xuanhui Wang, ChengXiang Zhai, "Learn from Web Search Logs to Organize Search Results", Department of Computer Science University of Illinois at Urbana-Champaign Urbana, IL 61801 xwang20@cs.uiuc.edu

[6] Pavithra.G1, Prabakar.D2, Dr. Karthick.S3, "A Survey On Query And Keyword Search In Database", International Journal of Science, Engineering and Technology Research (IJSETR), ISSN: 2278 – 7798, Volume 2, Issue 1, January 2013.

[7] Androutsopoulos, I., Ritchie, G.D., Thanisch, P., "Natural Language Interfaces to Databases - an Introduction", Journal of Language Engineering, 1/1, 1995.

[8] Bast, H., Chitea, A., Suchanek, F., Weber, I., "ESTER: Efficient Search on Text, Entities, and Relations", Proc. of SIGIR 2007.

[9] Agrawal, S., Chaudhuri, S., Das, G., "DBXplorer: A System for Keyword-Based Search over Relational Databases", ICDE 2002.

[10] Demidova, E., Zhou, X., Nejdl, W., "IQP: Incremental Query Construction, a Probabilistic Approach", Proc. of ICDE 2010.

[11] E. Demidova, X. Zhou, G. Zenz, and W. Nejdl, "SUITS: Faceted User Interface for Constructing Structured Queries from Keywords," Proc. 14th Int'l Conf. Database Systems for Advanced Applications (DASFAA), 2009.

[12] M.M. Zloof, "Query-by-Example: A Data Base Language," IBM Systems J., vol. 16, no. 4, pp. 324-343, 1977.

[13] E.Demidova, X.Zhou, and W.Nejdl, "IQP: Incremental Query Construction, a Probabilistic Approach," Proc. 26th IEEE Int'l Conf. Data Eng.(ICDE),2010.

[14] G. Zenz, X. Zhou, E. Minack, W. Siberski, and W. Nejdl, "From Keywords to Semantic Queries—Incremental Query Construction on the Semantic Web," J. Web Semantics, vol. 7, no. 3, pp. 166-176, 2009.

[15] D.Fensal, W. Wahlster, H. Lieberman, "Spanning the semantic web: Bringing the worldwide web to its full potential, "MIT Press 2003.

[16] Q. Zhou, C. Wang, M. Xiong, H. Wang, and Y. Yu, "SPARK: Adapting Keyword Query to Semantic Search," Proc. Sixth Int'l the Semantic Web and Second Asian Conf. Asian Semantic Web Conf. (ISWC), 2007.

[17] S. Tata and G.M. Lohman, "SQAK: Doing More with Keywords", Proc. ACM SIGMOD Int'l Conf. Management of Data (SIGMOD), 2008.