University Search Engine Based on Semantic Approach

Ankit Kanojia and Varsha Sharma

***Abstract*—As we know Semantic analysis of input query is future of web lead us to new era of web; makes our searching process more efficient while reducing searching time. The paper highlights and outlines USSE (University Semantic Search Engine) based on a Semantic web. In this search engine we analysis input query on the basis of noun and find out exact solution of input query. This makes our search more effective and efficient. Provide a new platform in which machine can understand information and process it without need of human interaction. This technique makes you to enter in era of semantic web (future web) and offer you new opportunities to develop simple applications based on semantic web.**

***Index Terms*—RDF, semantic search engine, semantic web, NET.**

I. Introduction

Natural Language is amazing, which allow us to analyze information or thoughts form different views. So one can ask stranger how to find the nearest coffee shop; without any effort.

As a simple example, think about the following two sentences. Both are of the form "subject-verb-object," one of the simplest possible grammatical structures:

1) Colin enjoys mushrooms
2) Mushrooms scare Jamie

Each of these sentences represents a piece of information. The words "Jamie" and "Colin" refer to specific people, the word "mushroom" refers to a class of organisms, and the words "enjoy" and "scare" tell you the relationship between the person and the organism. Because you know from previous experience what the verbs "enjoy" and "scare" mean, and you have probably seen a mushroom before, you are able to understand the two sentences. And now that you have read them, you are equipped with new knowledge of the world. This is an example of semantics: symbols can refer to things or concepts, and sequences of symbols convey meaning that you derived from the two sentences to answer simple questions such as "who likes mushrooms?"[1]

A. Semantic Web

Semantic web is refers to group of methods and technologies to allow machine to understand the meaning of information on www [2].

Benefits of semantic web:

- Provides standardized ways for data to be combined.
- Allowing Developers to focus on building data rich–applications rather than getting stuck on problems of obtaining and integrating data.

B. RDF

RDF provides a standard way of expressing graphs of data and sharing them with other people and, perhaps more importantly, with machines. [1]

Advantage:

- A large collection of tools and services has emerged around RDF.
- Used to remove ambiguity when transmitting semantic data between machines that may have no other knowledge of one another.

C. Ontology

Ontology is an explicit specification of a set of objects, concepts, and other entities that are presumed to exit in some area of interest and the relationships that hold them.

D. .NET

.NET provide platform to build web enrich application which are dynamic in nature. It provides many predefine function and tools.

E. SQL

SQL provide platform to store data or information. It also provides security and authentication.

This search engine programming is done in .NET (Microsoft Visual Studio 2008) and SQL (SQL server 2005) is used to keep data or information. Function of RDF and Ontology is done by using ID (by assigning a unique number) assign to different field or rows in data tables. This all things combine and build a semantic search engine which performs information retrieval not data retrieval.

II. Our Approach

In every online application database plays very crucial role. In this application database also play major role even their tables names also play crucial role in this search engine. Basically this search engine works for particular field related to university such as Student, Department, Employees and affiliated colleges; each of field have following attributes such as:

Students: Name, enrollment number, email, address, date of birth, course.

Employees: Name, Registration number, phone number, position.

Department: Name, location, phone number, HOD, department number, department details.

So any query could be fire related to these fields and their attributes in search engine. By analyzing input query search

Manuscript received June 17, 2012; revised July 2, 2012.

The authors are with the B. P. I. T, Affiliated to GGSIPU, Delhi, India (e-mail: ankitkanojia89@gmail.com).

engine can find accurate and precious result e.g. Name of students in xyz (college name)? , HOD of different departments? Etc...

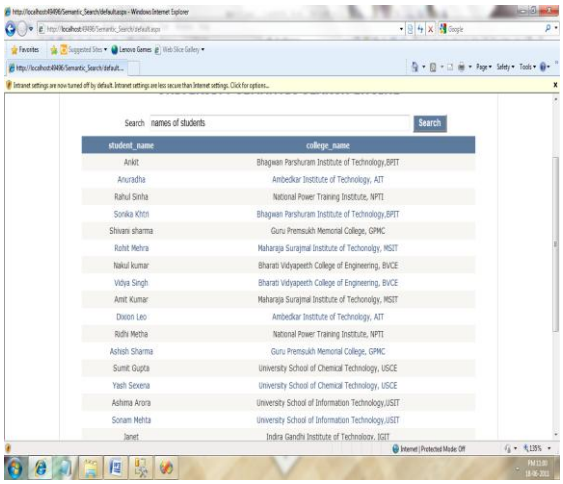

Fig. 1. Result of input query: names of students.

As we analyze all input queries related to above University fields, we able to reveal that each input query contains noun related to university field such as student, department, name, HOD etc. so, we constructed names of tables according to related field and this concept of noun and ID we basically used for searching in this search engine.

A. *Tables in Database:*

- College: contain names of all affiliated colleges.
- Table master: contain names of all tables.
- Column master: contain column names of all tables.
- Question master: used for synonyms purpose.
- Student master: contains data belongs to students.
- Employee master: contain data belongs to employees
- Department master: contain data belongs to departments in different colleges.

B. *Working:*

As query enters in text box and user press search button; firstly all punctuation marks (‘,’, ’.’, ‘!’, ‘?’, ‘:’, ‘;’) are removed from query. Then this one dimensional query is converted in two dimensional query such as query [ ]: “Names of students” converted as query [ ] = {Names of students}

In this engine we suppose that most of search query are related to college names, so we fire all words present in 2D array one by one on college table and gets ID of college name that occur in input query; if college name is not present in input query then after execution of input query we initialize college id by null. All IDs are auto generated as administrator enters data, correspondence id is generated. In this operation functional word (of, if, him, her, over … etc) are not fire on data tables.

Similarly again one by one all words fire on different tables such as: table_master, column_master, question_master and correspondence table id, question id and column id got according to input query. If somehow we don’t get id then we initialize correspondence id by null. At the end for retrieve of precious information we have to construct final query, which brings out exact solution of input query. We use column id and table id to find column name and table name for final query construction. By using these column name, table name and ids; we construct final query which have inner join operation with college table by using college id. This query fire on database and accurate information is retrieved from database for end user. In this way this engine finds out accurate and precious information without human interaction and capable to analyze information.

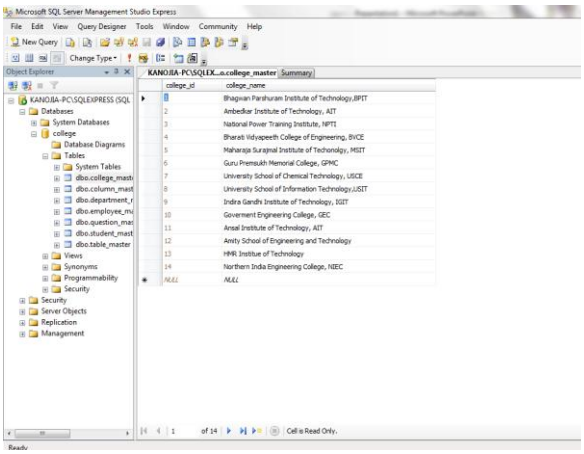

Fig. 2. College master table with unique ids.

C. *Special Features*

Synonyms Property: One important aspect of this search engine is that it is able to analyze synonyms e.g. location, place, position.

In this we implement this property by question table by keeping column id and table id same for synonyms present in different rows of database e.g.

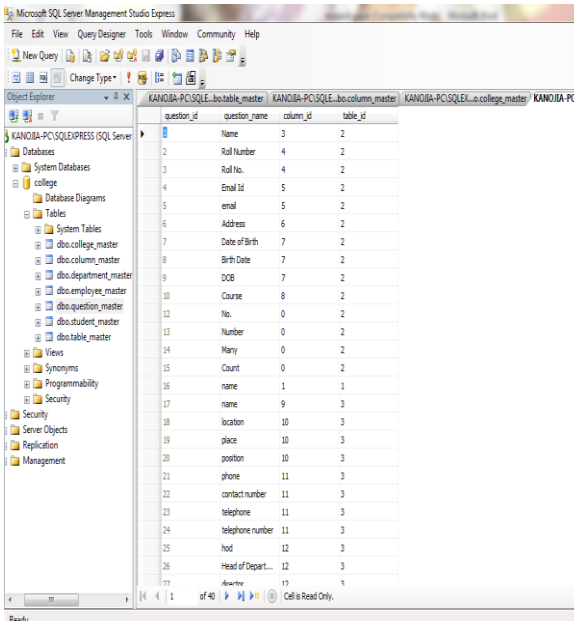

Fig. 3. Question master for synonyms analysis.

As we see in figure that place, location and position have same column id=10 and table id=3. By this way this search engine implements synonyms property. This property makes this search engine more powerful.

Addition Properties: For counting purpose we also provide special method in this search engine. As search engine detect keywords such as count, numbers, many in input query; it simply execute addition method using column name and table name to perform addition operation. Provided method

query such as "str + = " select count(*) as Number, college_name from #temp group by college_name"."

More functionality can be added to this search engine. We can easily enhance this engine by research and development.

III. Advantages

3) Remove inconsistency and redundancy for database.
4) Provide accurate and precious result.
5) This search engine perform information retrieval not data retrieval.
6) Reduce searching time
7) Allow machine to understand information and reduce human interaction.
8) Provide user friendly platform.

IV. Limitation

In this search engine we don't interlinked every row of tables. So, we can't find out result by using data present in each row e.g. Phone number of Mrs. Payal? (Payal name of employee, present in employee table.)

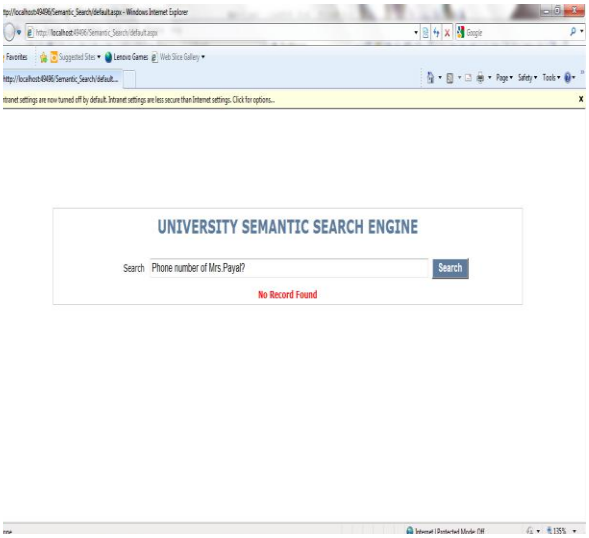

Fig. 4. Unlinked query result.

But by referring column name we can find out result because column names are include in question_master table. To remove this limitation we have enter data of each row into question_master table and interlinked it according to corresponds ID, which increase complexity too much. We are trying to find out alternative solution of this limitation.

V. Conclusion

The University Semantic search engine we have outlined in this paper lays the foundation for the construction of simple semantic application that can produce more accurate and precious result as compare to any other search engine. Allow machine to understand information and take decision like humans. The current form of search engine works on the search performed by humans but in this search engine the task performed by computer program. There are a number additional issues we have yet resolve but that further discussion, including strategies for enabling users to contribute to the semantic web, interference, trust and ontological mismatch.

VI. Future Development

In this application we can include administration module, which automatically enters data in all the data tables required. So, one can avoid manual entries in database.

With research in this field, we can easily enhance this application.

Similar type of application can be develops for school and office. So strategic information can be easily search in less time.

VII. Other Semantic Search Engines

OntoKhoj: Ontokhoj[Patel rt al., 2003] returns a ranked list of ontologies for a given wordNet-sense. Onto Khoj searches according to wordNet Synonyms and hypernyms of the input. Appropriate ontologies are ranked according to their interconnectivity, i.e. ontologies that are referred to more often are rankedhigher. Relationships considered by Onto Khoj are e.g. an rdfs: sub Class relationship to an element of another ontology. Onto Khoj differently weights different relations and also considers chained relationships. [3]

Swoogle: Swoogle [Ding et al. 2004] is again a search engine for ontologes (Semantic Web documents in Swoogle terminology). Ding et al. implement various crawlers, e.g. one using Google, another monitoring given websites and a third analyzing retrieved ontologies and spreading out using hyperlinks. Retrieved ontologies are indexed by keywords and metadata like ontology language and ranked. An ontology's rank is defined by where the ontologies lies on a continuous lies on continuous line between "database" and "schema". [3]

References

[1] T. Segaran, C. Evans, and J. Taylor, *Programming with sematic web*
[2] Wikipedia. [Online]. Available: http://en.wikipedia.org/wiki/MLearning
[3] Information retrival on the Semantic Web- Does it exits? [Online]. Available: http://knowcenter.tugraz.at/download_extern/papers/lwa-cr_HP.pdf

Ankit Kanojia was born in Delhi, India on September 23, 1989, He abtained his B. tech (IT) in 2011, He's now studying information technology in B. P. I. T (affiliated to GGSIPU). Email: ankitkanojia89@gmail.com

Second B. Varsha Sharma was born in Delhi, India on October 31, 1982. She is now Pursuing her M. Tech (IT) and studying information technology in B. P. I. T (affiliated to GGSIPU). Email: varshasharma@dce.ac.in