Volume 2, Issue 2, February 2012 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

LANLI: A Natural Language Interfacing Tool for Relational Database Query Generation

ENIKUOMEHIN, Oluwatoyin[1], SADIKU, J.S.[2]
Department of Computer Science
[1]Lagos State University, Lagos, Nigeria,
[2]University of Ilorin, Ilorin, Kwara state, Nigeria

***Abstract*--Most conventional databases in use today are based on the relational model. Values in a relation are taken from a finite set of strictly typed domain. Each relation in the database represents a proposition and each record in a relation is a statement such that it evaluates to 'true' for that proposition. Sql is known to be rigid and thus will not retrieve relevant items when users can't formulate appropriate queries. Problem exist when user do not have prior knowledge of the database and thus will not be able to formulate relevant queries. We think of the use of a generalized search interface that can handles natural language request by introducing a LANLI based item search technique. The approach uses the interrelationship of the Syntactic and Semantic tree for generating sql statements after due fuzzification has taken place to handle quantification. This paper opens a new research area for LANLI based fuzzifier.**

***Keyword*--: Semantic parser, Syntactic parser, Fuzzy hedges, quantification, trees**

I. INTRODUCTION

Query results do not satisfy the users to a large extent thus users are forced to make a decision or a choice based on the displayed output. The same problem occurs also within an organization when staffs are searching for a data from a single database. This can best be explained by the incident of the December 25th 2009 bombing attempt of a Detriot, USA based flight by a Nigerian. In that incident, which can attributed to Query misrepresentation, Adam Brookes released a bulletin on the said bombing of the flight 253 and I quote "*Once again, it is the failures of the US intelligence agencies that, we are told, are to blame. The report found out that the US government did have 'sufficient information' to disrupt the Christmas day attack. But that information was scattered around databases. It was never pulled together to present a coherent picture of the threat. A '**series of human errors**' occurred, apparently **someone misspelled Umar Farouk Abdulmutallab's name as they entered it in a database** and that is why no-one realize he had a US visa.* These could exist in a singular database or on a multidatabase system evenly spread and distributed on the internet. Present DB systems cannot process approximate queries. This means that it will not attempt to retrieve relevant items if the user cannot formulate the query exactly. They suffer the problem of rigidity. Even with the advent of modern research technique in this field, Quantifications has not been handled by current NLIDBs. e.g. NOT VERY etc there is a need to provide a centralized search interface to help information seekers in more effective and efficient search thereby presenting human usable results. This is where we need the Local Appropriator. An Appropriator is local if it works on a centralized system. We think of an appropriator as linker with enhanced capabilities. The capabilities involve the use of other tools such as fuzzy logic, natural language, machine learning etc.

The main method to store big amount of data nowadays is to store it in databases. This has led to the acceptance of Database Technology by organizations [1]. It means that more non database expert will need to interact with databases. This non expert does not understand SQL (the language for databases). Formulation of sql query statements by searchers/users for submission into relational databases and information retrieval systems have been a serious challenge that often leads to irrelevant search results. This is compounded by the level of uncertainty about the user's information need and in some cases, unfamiliarity with retrieval system. The task has therefore being generation of an appropriate system to precisely represent human requirement. This is the essence of NL processing.

The World Wide Web presents a more established challenge in this area considering the fact those searchers has little or no training on search techniques on the web. This research program recognizes the need for a NL interfacing tool to close the gap between automated systems and human thinking. It could be argued, however, that this required precision which does not actually gives a true representation of the world. The model is grounded in binary black-and-white but much of reality actually exists in shades of gray. As such, the conventional relational database model has limited usefulness. Human reasoning, talking etc contain high level of ambiguity in contrast the relational database is crisp and discreet. There is a need to develop a flexible technique for Meaning Representation with the database to assist ordinary user. The explosion of massive data sets created by businesses, science and government necessitates that such nested data emanating from the above should be accurately represented semantically. In this paper, we attempt to understand the process of structured parsing for an authentic meaning representation system.

II. SOME LITERATURE

Ninety-Five percent of the database-based applications used in recent times are based on relational model[2]. Such databases are those we refer to as relational databases. Every relation in the database represents a proposition and each record in a relation is a statement such that it associates a value of True or False to the proposition[3] also [4]. This hidden records or data are thought as organized data (information) such as document text. The need is prone to develop or change during this time and evolves from an initial, vague state into one known and understood by the searcher [5]. As the information need evolves the searcher's ability to articulate query statements improves based on his or her level of understanding of the problem [6]. The formulation of query statements can be a cognitively demanding process resulting in queries that are approximate or 'compromised' representations of information needs [7].To model the creation of the search query, Taylor suggests a continuum where searchers' abilities move initially from questions, to problems, to finally sense-making, although the boundaries between these three stages appear blurred [8]. Kuhlthau (1999) found in an empirical study that cognitive uncertainty increases during the initial stages of a search due to interpretative problems with the retrieved data. When the information needs are vague [9], there is an anomalous state of knowledge [10] or searchers are unfamiliar with the collection makeup and retrieval environment [11] problems with query formulation are magnified. We provide a fundamental for a deep meaning representation usable for non keyword search, such will handle the ambiguity contained in human expression (quantification) with features that are not yet handled in current NLIDbs.

III. NATURAL LANGUAGE IN USE-THE NLIDB

Natural Language Interface to databases means provided an easy access to database system without the need for the user to use formal query languages, such as SQL. Database query languages can be difficult to the non-expert users and learning these formal queries takes a lot of time. In our proposed approach, system specification may contain contradictions, ambiguities, incomplete statements etc. These are features that are not handled in general relation dbs queries. Our approach called LANLI-LOCAL APPROPRIATOR NATURAL LANGUAGE INTERFACE extract sufficient information from each user presentation and generate appropriate user response. Theoretically, it answers the question of how people intentions are interpreted by adequate Meaning Representation; this bridges the gap that exists in the formulation of queries to represent the actual use intention. As shown in Section II, the main problem with the continued acceptance of relational database is that it has to be continuously updated with appropriate sql queries. The current architecture does not support operators such as: greater than, less than, count, average and sum. It does not resolve dates as in: before, after, between. The generated SQL does not support imbricate queries. The proposed method eliminates all tokens that cannot be matched with either the semantic sets or with the index files and it works for semantically stable databases. The preprocessor must be used after each semantic update of the database in order to modify the index files. The context disambiguation is limited to the semantic sets related to a given schema errors related to tokenizing and the human intervention propagates in the SQL query. The existing method completely disregards the unmatched tokens and thus it cannot correct the input query if it has errors.

IV. METHODOLOGY

To help ordinary user find information scattered around databases locally or on the internet, we propose the use of Unguided Loose Search- users can just write their request in terms of natural language or by presenting a set of keywords to express their information need without having to worry about the database name, schema and syntax. An added knowledge in our approach is that the ULS provides loose answers such that result of any request is flexible and not crisp allowing for quantification within answers. Different organizations present millions of databases all together on the internet. It is worth to note that each organization has an interface to its database though they are majorly crisp and cannot be useful to an ordinary user who do not have prior knowledge about the database formulation however we need to provide a centralized search interface to help information seekers in more effective and efficient search thereby presenting human usable results. We expand the use of the Local Appropriator earlier proposed. An appropriator is local, if it works on a closed centralized system. We think of an appropriator as linker with enhanced capabilities. The capabilities involve the use of other tools such as fuzzy logic, natural language, Machine learning etc. The appropriator is difficult to design due to its underlining functions. In this system, the result of a search on a dbms is presented as a table. Answers to a query are not limited to individual tuples but result assembled from joining of several tuples. Data extraction, document retrieval, query answering from relational databases has been extensively studied over sometimes now. To discuss our intervention, consider the following example:

A Business woman goes to Lagos, (Nigeria commercial city) for a five day trading business, she is interested in staying in a hotel that is not very expensive and not very far from Idimota market(A major market): She formulates the query thus: A convenient hotel not very far from the idimota.

In the Example above, *not very expensive* and *not very far* are statement of preference and the semantic interpretation can defer depending on the person. To handle such conditions, we need to implore the context of the Fuzzy Set theory [12.]. Since most relational databases as presented by codd[13] are based on relational model of simple algebra, it means that, for the above example to be implemented as a query over a database, the query will have to be reformulated. Then the user will have to reformulate it to:

A hotel that is not more than N5000 and not more than 2km to the market. So if there exist a table for hotels then an SQL is formulated from the above as:

SELECT * FROM table WHERE price <= N5000 and distance <= 2km.

User can then view the available result under the chosen condition. The above is an example of the classical query system whose main disadvantage is that it does not give discriminated answers to user preference. The example below shows

S/ N	Name	Cost/ (Naira)	Distance to market(KM)
1	AA	1000	3
2	AB	4000	2
3	AC	8000	5
4	AD	7000	2
5	AE	7500	8
6	AF	5000	1
7	AG	13000	8
8	AH	10000	2
9	AI	8000	7

Table 1: All available Hotels, Cost of Hotels and Distance to Market

Then the query: SELECT * FROM table WHERE price <= N5000 and distance <= 2km satisfies

s/n	name	cost	distance to market	
2	AB	4000	2	
6	AF	5000	1	

Table 2: Hotels that meet the stated criteria

Given the result above, the system has not helped the user to find the best result thus user will still have to use his/her preference. A deeper problem exists if the answer set is empty assuming that no hotel is in less than 2km in distance. This is called the problem of rigidity which is a consequence of the Boolean logic system[14] and displayed in the performance of SQL. This forms the basis of the disadvantages of the classical querying system.

In this, users need to have a prior knowledge of the domain of interest or the database schema and also the query syntax; this act becomes a burden to an ordinary user. Therefore, the degree to which the resulting query representation reflects the users information need is based on the user's ability to formulate a precise query. Unguided Loose Search (ULS) allows users to input query against databases at pure linguistic level. To do this, we use the concept of the LOCAL APPROPRIATOR that serves as an engine for the transformation of relative trees. Trees are used as the major item for connectivity between the user input in natural language and its sql generation. Between this two ends, lots of transformation has to take place. Users requested are tokenized and analyzed for the purpose of generating a structure and meaning from it. The generated tree called the SySe tree using a combination of Link parser and the Wordnet to generate semantic and syntactic structure for any word. Wordnet [15] is an online lexical reference system developed at Princeton. Wordnet can be downloaded for both Unix and Window. In this paper, we use a novel approach to NLIs by generating the said tree model. Theoretically, the method is a well-grounded sql generation process for complex-quantified sentences. We combine two well accepted concepts (Wordnet and Link parser) for natural language Meaning Representation. From this representation, an appropriate semantic model can be inference. We based the generation of our model on the principles of the Earley's Algorithm since the intention is to develop an appropriate parser.

Given sentence s and parse p.
To Model: conditional probability $\Pr(p|s)$ where items $d1, \ldots, d|p|$, are in p and each item dj belongs to a possibly infinite space D.
For notational convenience, define $d0$ as s.
We then have:
$\Pr(p|s) = \Pr(d1, \ldots, d|p||d0)$ 1.1
$|p| = \Pr(dj \mid d0, \ldots, dj - 1|)$ bj $=1$

Equation 1.1 is invoked from the chain rule[16]
Note that

$Pr(dj \mid d0, \ldots, dj - 1) = 0$ if the items in the state are irregular.

We call $d0, \ldots, dj - 1$, the state (a.k.a. history) at the *jth* step in the path. For consistency, the state is a set of items whereas the history is a sequence of items. We assume that the order of the items in the history is not used in the modelling.

We will abbreviate the state

$$d0, \ldots, dj - 1 \text{ as } Sj, \text{ where } Sj \in D \quad 1.2$$

Rewriting the above, we have:

$$Pr(dj \mid d0, \ldots, dj - 1) = Pr(dj \mid Sj) \quad 1.3$$

We abbreviate $D(Sj) \subseteq D$ to mean all items at state *Sj* that are permitted by the logic,

i.e. $d \in D(Sj)$ if and only if $Pr(d|Sj) = 0$.

Independence assumptions can be introduced into the model. For example, we can make the Markov assumption that only the input and the last item affect the probability of the current item:

$$Pr(dj \mid Sj) \Rightarrow Pr(dj \mid d0, dj - 1) \quad 1.4$$

Independent assumptions are considered because it supports the use of polynomial-time algorithms (e.g. Dynamic programming) to find the maximum probability parse for a given sentence, or at least to reduce the size of the search space.

For a truly usable probability distribution, we impose a restriction that the sum of the probabilities of all items at a given state sum to 1:

$$Pr(d|Sj)=1, d \in D(Sj) \quad 1.5$$

this restriction is implemented by normalization (locally-normalized conditional models) however such can cause label bias. The Earley's algorithm is suitable for parsing strings that belong to a given context free language [17]. Context free language and grammars (CFGs) cannot refer to any text around the production, or rule for context. Chomsky introduced four types of formal grammars in terms of their generative power known as Chomsky hierarchy. A hotly contested issue over several decades has been the question where natural languages are located within this hierarchy. Chomsky showed that NLs are not regular and he also presumed that NLs are not context free. Unlike most approaches in NLP, the proposed method makes no independence assumptions.

V. PARSING MODEL

The probability recursive model similar to [18]is used. The probability of a parse tree T given a word string W is rewritten using the bayes rule as:

P(T/W)= P(T)P(W/T)/P(W)

where P(W) is a constant for any word string.

The benchmark of the parser is the context free parsing that satisfies the Earley's algorithm when implemented in JAVA. Parse tree represents the syntactic structure of the sentence according to some formal grammar approved by the network model. Root node, non terminal branches and terminal nodes (leaf) exist.

VI. REPRESENTATION OF QUANTIFIERS

Since the intention is to handle all natural language on the interface, we consider some complexity associated with quantifiers. Fuzzy representation has been used as a major tool for handling quantifiers. We present a mathematical model for linguistic hedge, it is considered as a very essential part of the process of generating appropriate meaning or semantic for any given query. The semantic of any query is dependent on the level of hedges present in the query. Using the linguistic hedges **very**, positively, negatively, slightly more, slightly less etc has been an area in fuzzy representation in Natural Language. Exponents are used as factors for dilation. Here, we consider some representation for hedges:

Very. The very modifier returns the expanded fuzzy value passed as its argument, having raised all the membership values of the fuzzy value by a factor of two. Extremely modifier returns the expanded fuzzy value passed as its argument, having raised all the membership values of fuzzy value by a factor of eight or three. But the new definitions proposed here are slightly different and yet more meaningful.

These new definitions are formulated as follows.

$$very^n = \sqrt{\frac{2^{n+1}}{2}} = F \text{ to the power } \sqrt{2^n},$$

$$very^n = \sqrt{\frac{2^{n+1}}{2}} = F \text{ divided by } \sqrt{2^n},$$

When n = 1,

very = F to the power $\sqrt{2^{1,}}$

very = F divided by $\sqrt{2^{1,}}$

When n = 2,

very very = F to the power $\sqrt{2^{2,}}$

very very = F divided by $\sqrt{2^{2,}}$

When n = 3,

very very very = F to the power $\sqrt{2^{3,}}$

very, very very = F divided by $\sqrt{2^{3,}}$

Using exponential functions as a factor for concentrators, the following ensure

very ^{n}F = F to the power $e\sqrt{1/e^{-n}} = \sqrt{e^n},$

very ^{n}F = F divided by $e\sqrt{1/e^{-n}} = \sqrt{e^n},$ when n = 1,

very F = F to the power $\sqrt{e^1},$

very F = F divided by $\sqrt{e^1},$

For the above formal plan and representation, the problem of users unsatisfied still exist as users will have to choose the boolean result. Introducing fuzzy boundaries into the above means that query Q on D would be relaxed in order to accommodate the non-crisp (fuzzy) ends [19]. The representation can now be deduced to:

Let Q be the fuzzy query:

(P_1, P_2, ………………………… P_k E Q) (P_i = fuzzy predicates)

Weakening can be carried out here by applying basic uniform transformation to each production.

P: The transformation process can be applied literally.

Applying fuzzy K-NN (nearest neighbour) to this, we have that Vpi) p;K-NN (P_i with a near neighbour), P_i will be divided into classes by a function of the nearest neighbour such that; X's membership in class:

$$\bigcup i(x) = \frac{\sum_{j=1}^{k} uij \left[\frac{1}{IIx - x_j II^{\frac{2}{m-1}}}\right]}{\sum_{j=1}^{k} uij \left[\frac{1}{IIx - x_j II^{\frac{2}{m-1}}}\right]} \text{ for all}$$

Where, similar to Baye's Rule, the denominator

$$\sum_{j=1}^{k} \left[\frac{1}{IIx - x_j II^{\frac{2}{m-1}}}\right] \text{ is a normalising factor}$$

This will produce a more accurate result as each query resolves with it NN.

VII. Meaning Generation Pre and Post-processing

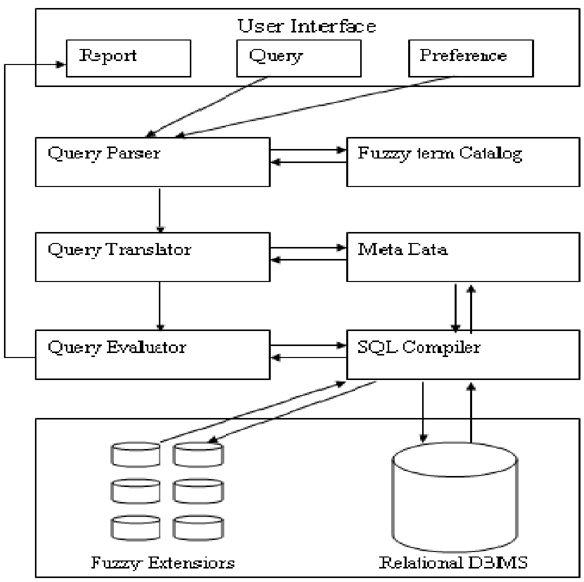
The database schema is scanned to identify table name and attributes which is stored in knowledge like file. These names are sent to Wordnet for meanings. This occurs after the syntactic tree has been generated. Semantic grammar deals with meaning of tokenized word however it differs from English meaning slightly. As example, employee in English grammar is a noun but semantically it is a database table name. Rules are generated based on database schema and attribute. We used Wordnet to find synonyms, hyperyms and hyponyms of token either single lexicon or composite lexicon. Directly mapping NL sentences to logical form using string-to-tree transduction rules. Once the SySe tree is generated, traversal of bottom-up model takes place. At this point, we identify tokens are placed in the LA post processor which already contain the database details such as table name. The advantage of using Wordnet is to allow pre processor to be more flexible than other systems such as SQ-HAL[20]. The mapping of the syntactic and semantic tree is formulated to form the SySe tree which forms the benchmark of our sql development. The LA has thus generated a new format of the user's request which can now be transformed into SQL.

VIII. GENERATING SEMANTIC TREE

The LA pre processor builds the results of wordnet and uses it for query generation since the task is to map the NL into actual element of the SQL, the generator uses four routines map the SELECT (DML), FROM (TABLE NAME) where (CONDITION) and ORDER BY. LA has been enhanced to interact with user through the interface by semantically matching patterns based on meaning. RDBMS has rich semantic feature- Glory to engineers that developed it, for example, the table Stud_dept will be understood semantically as student table. We use the SQL SELECT statement to generate a general SQL transformation template. We develop it such as one table could be created at a time. Our model is implemented in a like interface that accepts English as the language of input. It is implemented. The program is placed on PHP and java for the development of the adjoining interface. The interface takes in the request and prints out the required search result.

IX. IMPLEMENTATION

In the implementation of this work, a form is required to be filled depending on the type of web application that is being run. It's just a simply HTML form that collets search terms and form a query from it. When the Submit button is pressed, the query is then sent to the server. The query is sent to the CGI script that invokes a stand-alone php/Java application running on our server. This application parses the query, and if no errors are found, hands it in to the query execution engine which produces the result as a list of tuples that gets formatted into an HTML table and is shipped back to the user. This interface, fast and with limited user interaction, has the advantage such that it can be used with any browser, database or data repository. Due to security restrictions in accessing remote servers, a query result cannot be accessed directly by a script, therefore a client-server paradigm is deployed such that the query script itself carries the user interaction, and a server process does the actual processing. The script communicates with the server through a custom designed protocol that allows the user total control over the query execution depending on the access level since not all information on the database or on a document are classified public. The LANLI compiler parses the query and translates it into a nested loop program in a custom-designed object language. The build is developed using php. The object program is executed by an interpreter that implements a stack machine. Its stack is heterogeneous, that is, it is able to store any type of object, from integers and strings to whole vectors of Document, image, data and its tuples. An important addition to knowledge is the inclusion of of a linguistic table as an unseen table in all databases. This will be implemented by the server. The query format generated above is first checked for its type. That is whether it is deletion, update, selection or insertion query. In case it is an insertion query, the server goes directly to implement the query as no action is needed. In case it contains linguistic hedges, the membership function is calculated and then manipulated based on the hedges. Then the aggregated result is compared (similarities) and displayed as ranked result thereby satisfying the users intention to a high degree



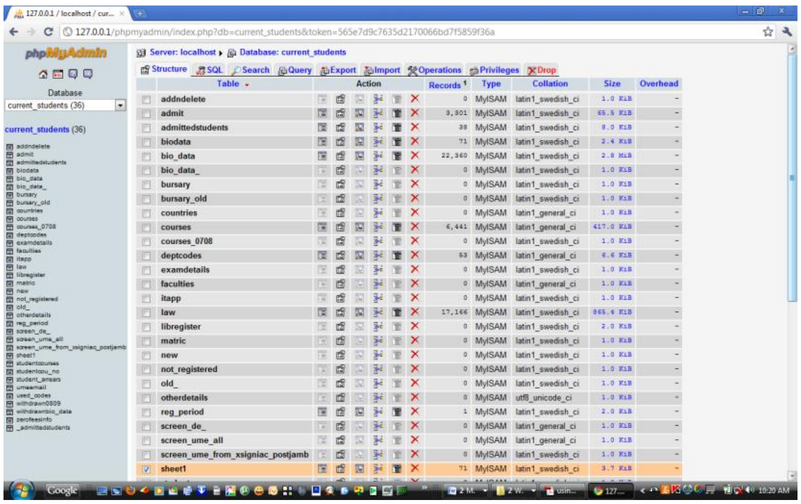
X. EXPERIMENTAL RESULT

We use the following Natural Language request to test our proposed architecture. "Show students in the department of computer science''. Here the user is interested in knowing the students that are in the department of computer science but cannot formulate a corresponding. The proposed interface handles that and transforms this request into:

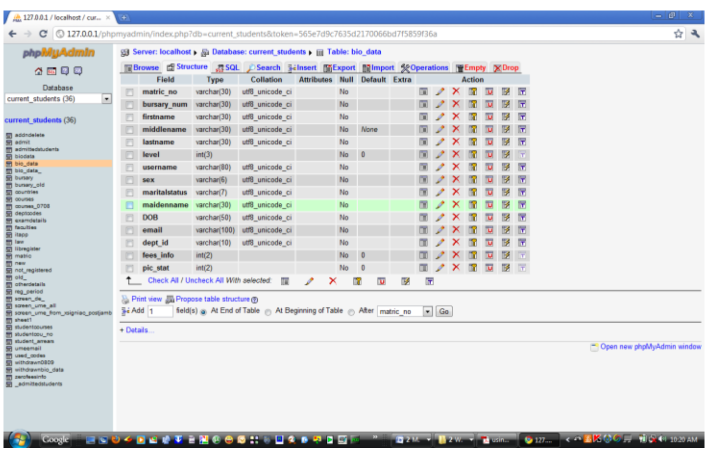
SELECT * From Students WHERE student. department= 'computer science',

To implement this model, the database used is shown below. It is student database that contains 36 table (excluding the meta data table and linguistic hedge table that are automated by the query server). Each table is interrelated by a common database attribute.

The attributes having values ranging from varchar, int to full int.

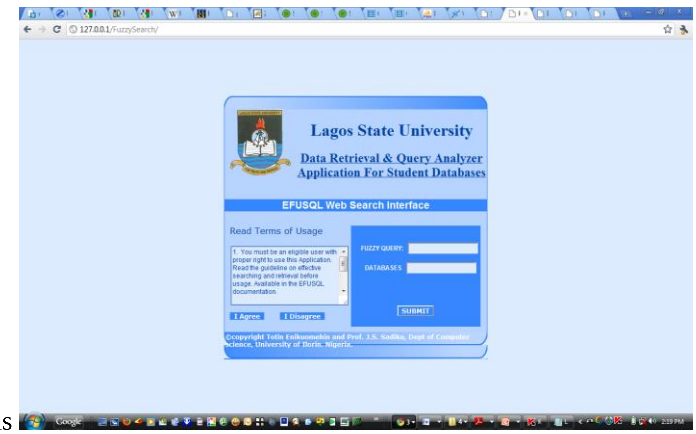


For any table, the tuples are inherent, see the table bio_data below;

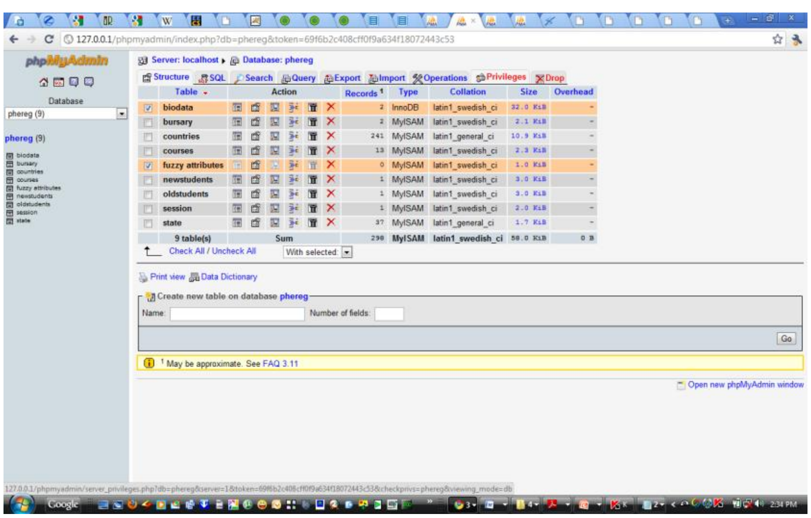


The interface as shown below is designed using php. This interface provides text area where query is to be written, list of databases, text area to accept attribute threshold along with the term command radio buttons as shown in below

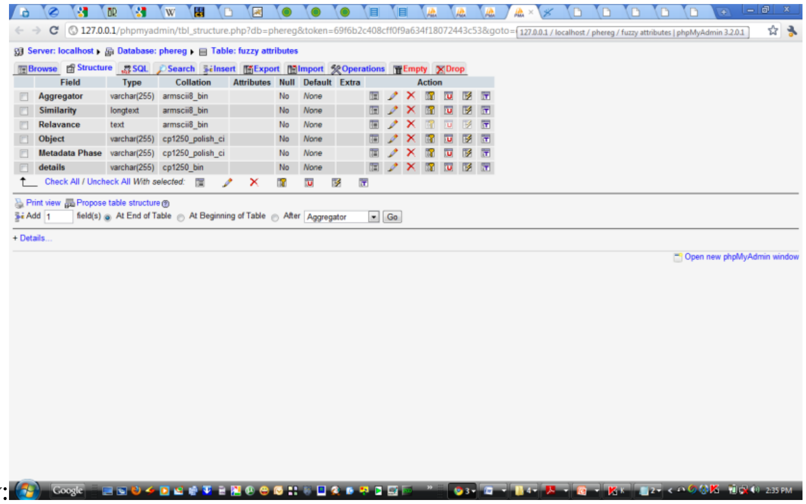
An earlier version developed with fewer text field is



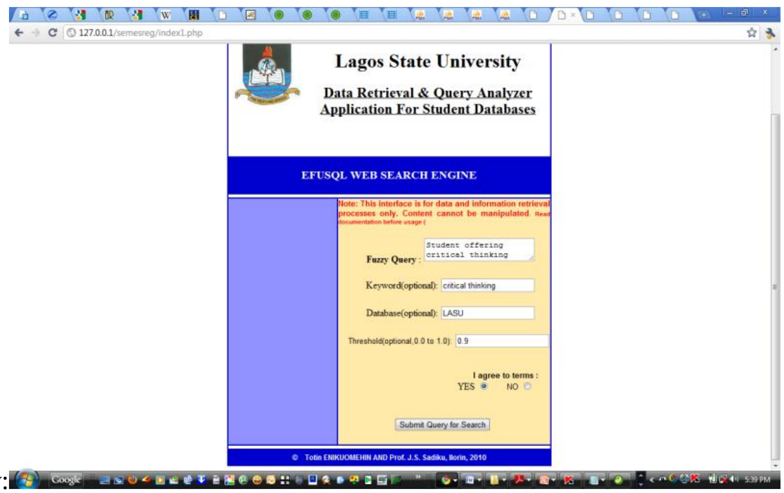
Also found below the backend that the above connects to after running the parser process described above.



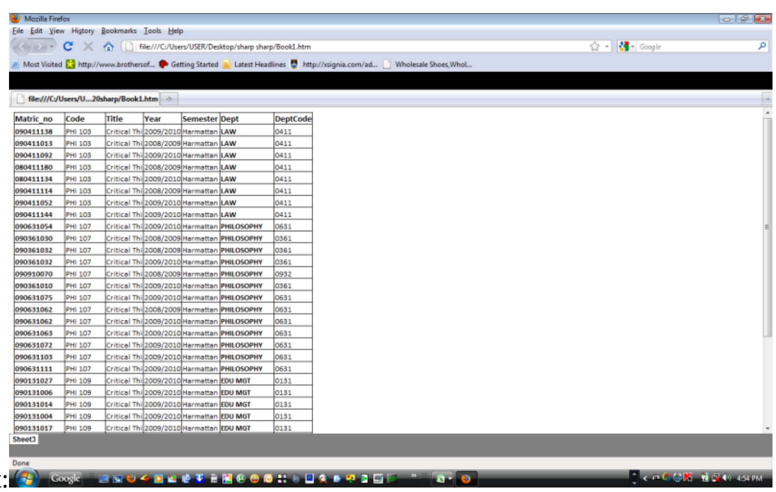
The table that makes the difference is the automatically generated fuzzy attribute table that compares the linguistic representation of the query and its seatic with other content of the database thereby returning result based on rank of similarities and the database now contains these componetnts. The fields in the table are as shown below:



Example query is the request of all student taking a course titled critical thinking as in the interface below:



This query runs against 230,000 record contained in the database and yield the following result:

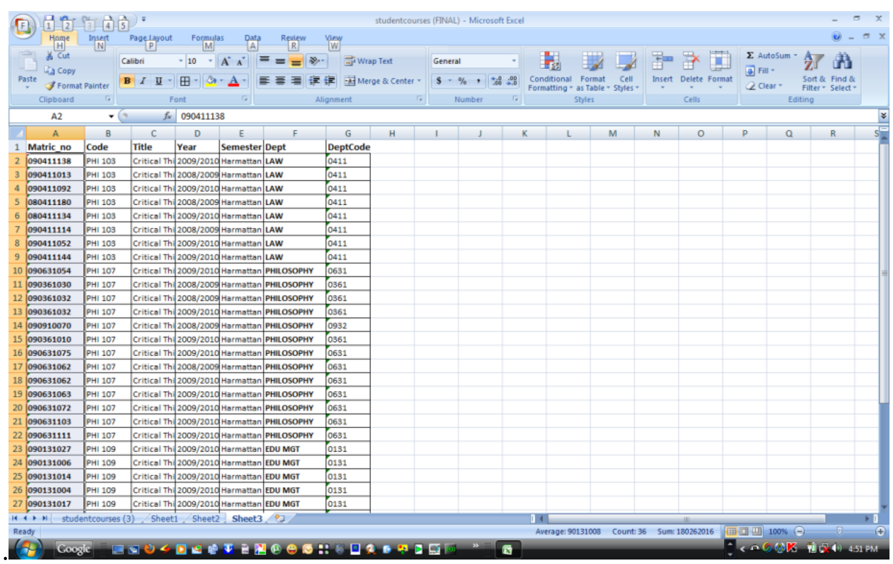


Matric_no	Code	Title	Year	Semester	Dept	DeptCode
090411138	PHI 103	Critical Thi	2009/2010	Harmattan	LAW	0411
090411013	PHI 103	Critical Thi	2008/2009	Harmattan	LAW	0411
090411092	PHI 103	Critical Thi	2009/2010	Harmattan	LAW	0411
080411180	PHI 103	Critical Thi	2008/2009	Harmattan	LAW	0411
080411134	PHI 103	Critical Thi	2009/2010	Harmattan	LAW	0411
090411114	PHI 103	Critical Thi	2008/2009	Harmattan	LAW	0411
090411052	PHI 103	Critical Thi	2009/2010	Harmattan	LAW	0411
090411144	PHI 103	Critical Thi	2009/2010	Harmattan	LAW	0411
090631054	PHI 107	Critical Thi	2009/2010	Harmattan	PHILOSOPHY	0631
090361030	PHI 107	Critical Thi	2008/2009	Harmattan	PHILOSOPHY	0361
090361032	PHI 107	Critical Thi	2008/2009	Harmattan	PHILOSOPHY	0361
090361032	PHI 107	Critical Thi	2009/2010	Harmattan	PHILOSOPHY	0361
090910070	PHI 107	Critical Thi	2008/2009	Harmattan	PHILOSOPHY	0932
090361010	PHI 107	Critical Thi	2009/2010	Harmattan	PHILOSOPHY	0361
090631075	PHI 107	Critical Thi	2009/2010	Harmattan	PHILOSOPHY	0631
090631062	PHI 107	Critical Thi	2008/2009	Harmattan	PHILOSOPHY	0631
090631062	PHI 107	Critical Thi	2009/2010	Harmattan	PHILOSOPHY	0631
090631063	PHI 107	Critical Thi	2009/2010	Harmattan	PHILOSOPHY	0631
090631072	PHI 107	Critical Thi	2009/2010	Harmattan	PHILOSOPHY	0631
090631103	PHI 107	Critical Thi	2009/2010	Harmattan	PHILOSOPHY	0631
090631111	PHI 107	Critical Thi	2009/2010	Harmattan	PHILOSOPHY	0631
090131027	PHI 109	Critical Thi	2009/2010	Harmattan	EDU MGT	0131
090131006	PHI 109	Critical Thi	2009/2010	Harmattan	EDU MGT	0131
090131014	PHI 109	Critical Thi	2009/2010	Harmattan	EDU MGT	0131
090131004	PHI 109	Critical Thi	2009/2010	Harmattan	EDU MGT	0131
090131017	PHI 109	Critical Thi	2009/2010	Harmattan	EDU MGT	0131

The result of the fuzzy query "list of all student taking critical thinking " .The result produced by the system is a total of 1622 records from three distinct departments which are LAW, PHILOSOPHY and EDUCATION MANAGENT. This result is shortened to allow screen-print. The output will be displaced as a web page and a copy downloaded into the system if the result is fetched from more than one table. The downloaded copy can be any spreadsheets format such as excel, Google doc etc. Ordinary query will just match terms that is, critical and thinking and populate result but the new query system now display the result in a useful format showing intelligence. The output can be analyzed as follows:

1. critical thinking is a compulsory course in law. It must be passed be before graduation. In fact, it is the only Junior level (100L or 200L) course taken again at LAW School after graduation.
2. Students of Philosophy take critical thinking as an addendum course and could be waved even if not taken before graduation.
3. Educational Management Student takes the same course as a elective, definitely a student may or may not take it.

Back to the results, it was displayed with LAW first followed by Philosophy and then education Management, without doubt the system, new query format, has brought great human intelligence and perception to querying as we derive from above that the course is more principal and key for those in LAW than any other department and the result is displayed in that level of relevance. Its also good to note that this will not be appreciated due to the limited number of results (1000+) however if we imagine the same query to be run to be a Centralized database of Nigeria student where the result set will be 100s of 1000s, then the ranking and intelligence will be appreciated because due to the large dataset user will have to continue to visit page by page before getting result but with the inclusion of thresholds the query target has been brought closer to their intention



	Matric_no	Code	Title	Year	Semester	Dept	DeptCode
2	090411138	PHI 103	Critical Thi	2009/2010	Harmattan	LAW	0411
3	090411013	PHI 103	Critical Thi	2008/2009	Harmattan	LAW	0411
4	090411092	PHI 103	Critical Thi	2009/2010	Harmattan	LAW	0411
5	080411180	PHI 103	Critical Thi	2008/2009	Harmattan	LAW	0411
6	080411134	PHI 103	Critical Thi	2009/2010	Harmattan	LAW	0411
7	090411114	PHI 103	Critical Thi	2008/2009	Harmattan	LAW	0411
8	090411052	PHI 103	Critical Thi	2009/2010	Harmattan	LAW	0411
9	090411144	PHI 103	Critical Thi	2009/2010	Harmattan	LAW	0411
10	090631054	PHI 107	Critical Thi	2009/2010	Harmattan	PHILOSOPHY	0631
11	090361030	PHI 107	Critical Thi	2008/2009	Harmattan	PHILOSOPHY	0361
12	090361032	PHI 107	Critical Thi	2008/2009	Harmattan	PHILOSOPHY	0361
13	090361032	PHI 107	Critical Thi	2009/2010	Harmattan	PHILOSOPHY	0361
14	090910070	PHI 107	Critical Thi	2008/2009	Harmattan	PHILOSOPHY	0932
15	090361010	PHI 107	Critical Thi	2009/2010	Harmattan	PHILOSOPHY	0361
16	090631075	PHI 107	Critical Thi	2009/2010	Harmattan	PHILOSOPHY	0631
17	090631062	PHI 107	Critical Thi	2008/2009	Harmattan	PHILOSOPHY	0631
18	090631062	PHI 107	Critical Thi	2009/2010	Harmattan	PHILOSOPHY	0631
19	090631063	PHI 107	Critical Thi	2009/2010	Harmattan	PHILOSOPHY	0631
20	090631072	PHI 107	Critical Thi	2009/2010	Harmattan	PHILOSOPHY	0631
21	090631103	PHI 107	Critical Thi	2009/2010	Harmattan	PHILOSOPHY	0631
22	090631111	PHI 107	Critical Thi	2009/2010	Harmattan	PHILOSOPHY	0631
23	090131027	PHI 109	Critical Thi	2009/2010	Harmattan	EDU MGT	0131
24	090131006	PHI 109	Critical Thi	2009/2010	Harmattan	EDU MGT	0131
25	090131014	PHI 109	Critical Thi	2009/2010	Harmattan	EDU MGT	0131
26	090131004	PHI 109	Critical Thi	2009/2010	Harmattan	EDU MGT	0131
27	090131017	PHI 109	Critical Thi	2009/2010	Harmattan	EDU MGT	0131

Actually, between the input query on the interface form and output above is the server side implementation of the developed query. On clicking submit, he application calls the server parser parameter and the query is retranslated following this format below.

SELECT (attributes "predicates")
WITH relation attributes {optional}
FROM (LANLI relation)
WHERE(Conditions "fuzzy conditions" ≥α)
// note the inclusion of α-cut as threshhold
HAVING (F Quantified THOLDT)
ON (FUZZY [P] (<fuzzy-condition>)
where ("attribute" Relaxed Normalized Conditions "fuzzyR conditions ") [D] = (SELECT A
FROM R (Table clauses)
WHERE (condition) & F QUANTIFIER T HOLD)

This simple query called O-level query resolves the fuzzy condition with the help of the fuzzy extensions then the A-level query fetches the qualifying record from the target tables.

XI. CONCLUSIONS

The Model presented in this work is simulated by using php as front end for embedded in a school application and Mysql as back end with wamp as implementation server. We have been able to develop a novel syntactic-semantic parser using the concept of LOCAL APPROPRIATOR that involves the development a disambiguated parse tree that can be mapped into an already proven semantic concept: Wordnet, which is based on semantic- relation for efficient SQL generation especially for long sentences where existing tree models fail. We further show the efficient use of an early algorithm, the Earley's algorithm with the Chomsky normal form for grammar representation. We showed that the result could be modeled with some characteristics of the Bayes pattern of event representation. Semantic structure was extended to quantification where hedges were shown to change the semantic formation of the SySe tree, once the SySe tree is altered, then the corresponding sql will be altered. We summarized by presenting the result of our experiment where the proposed model was implemented using php on mysql backend. Some features were implemented with java script for easy integration into web based applications.

XII. FUTURE WORK

Extend the current model to non crisp database like fuzzy databases. Syntatic parsing for menu based NLIDB will be considered. [21]. Application to health for an automated drug prescription expert will also be proposed[22], the results will be measured against complex sentences involving more than 4 tables

REFERENCES:

1. Ribeiro, R. A., &. Moreira, A. M. "Intelligent query model for business characteristics". Proceedings of the IEEE/WSES/IMACS CSCC'99 Conference, 1999.
2. Code E,F, A relational model for large shared data banks communications of ACM; 13, 6 (1970. 377-387
3. B. Bhuniya, P. Niyogi, Lossless join property in fuzzy relational databases. Data Knowledge Engrg, 11:109-124, 1993
4. Date, C. J, An introduction to database systems (7th ed.)". Reading: Addison-Wesley, 2000.
5. Galindo, J. "Tratamiento de la imprecisión en bases de datos relacionales: Extensión del modelo y adaptación de los SGBD actuales". Doctoral thesis, University of Granada, Spain. (Retrieved February7, 2008), from http://www.lcc.uma.es, 1999.
6. Galindo, J. "FSQL (fuzzy SQL): A fuzzy query language". (Retrieved February 7, 2008), from http://www.lcc.uma.es/~ppgg/FSQL , 2007
7. (Belkin, 2000) A. Mendelzon, G. Mihaila, and T. Milo. Querying the World Wide Web. In Proc. of First Int. Conf. on Parallel and Distributed Information Systems (PDIS'96), pp. 80-91, Dec. 1996.ftp://db.toronto.edu/pub/papers/pdis96.ps.gz
8. Taylor, Robert S. (1968). "Question negotiation and information seeking in libraries". Journal of College and Research Libraries 29 (3), 178-94.

9. Kuhlthau, C. C. Accommodating the user's information search process: Challenges for information retrieval system designers. Bulletin of the American Society for Information Science 1999, 25, 12-16.

10. Belkin, N. J. (1980). Anomalous states of knowledge as a basis for information retrieval. The Canadian Journal of Information Science, 5, 133-14

11. M ¨uller, A., and Thiel, U. (1994). Query expansion in an abductive information retrieval system. In Proceedings of the Conference on Intelligent Multimedia Information Retrieval Systems and Management (RIAO ' 94), Vol. 1, 461–480

12. Zadeh, L., Fuzzy Sets, Information and Control, Vol. 8, (1965), Pp. 338-353

13. Codd, E.F. (1970). "A Relational Model of Data for Large Shared Data Banks". Communications of the ACM 13 (6): 377–387

14. Spink, A., Wolfram, D., Jansen, B. J., & Saracevic, T. (2001). Searching the Web: The public and their queries. Journal of the American Society for Information Science, 53(2), 226- 234.

15. Wordnet WordNet, http://wordnet.princeton.edu/

16. E. Black, S. Abney, D. Flickinger, C. Gdaniec, R. Grishman, P. Harrison, D. Hindle, R. Ingria, F. Jelinek, J. Klavans, M. Liberman, M. Marcus, R. Roukos, B. Santorini & T. Strazalkowski (1991). A procedure for quantitatively comparing the syntactic coverage of English grammars. : Proceedings of the DARPA Workshop on Speech and Natural Language, 306-311, Pacific Grove, CA, February.

17. Earley, Jay (1970), "An efficient context-free parsing algorithm", Communications of the ACM 13 (2): 94–102

18. Pfeffer, A. 2000. Probabilistic Reasoning for Complex Systems. Ph.D. Dissertation, Stanford University.

19. Medina, J., Pons, O., & Vila, M. (1994). GEFRED: A Generalized Model for Fuzzy Relational Databases. Information Sciences , 76 (1), 87-109

20. http://www.csse.monash.edu.au/hons/projects/2000/Supun.Ruwanpura/

21. Belkin, N. J., Cool, C., Stein, A., and Thiel, U. (1995). Cases, scripts, and information seeking strategies: On the
design of interactive information retrieval systems. Expert Systems and Applications 9(3):379–395

22. Waldinger, R. et al. "Accessing structured health information through English queries and automatic deduction," in Proc. of AAAI Spring Symposium on Health Communications, 2011