RESEARCH ARTICLE OPEN ACCESS

Natural language Interface for a DBMS inference system

Khalid Ahmed Ibrahim*

*(Faculty of Computer Science and Information Technology, Karary University)

--************************----------------------------------

Abstract:

This paper outlines a language interface for a database. The proposed system i-kiosk uses corpora-based techniques to map the input query of an user in natural language (English) format and generates a corresponding SQL statement, wont to retrieve results from the database. This minimizes user prompts and aims for an automatic response generation from the system without the necessity for extensive GUI menus thus making it user friendly and more practical. It sketches the methodology, proposed architecture and advantages and constraints and concludes with implications and suggestions for future research

Keywords **—Database, Language Interface, i-kiosk, Natural Language, GUI.**

--************************----------------------------------

I. INTRODUCTION

Purpose of the system

This system purports to simply accept user queries in an exceedingly language English during this example and converts it to a SQL statement. This is then passed on to the backend database and therefore the results output to screen. This helps any lay user with no prior knowledge of SQL to retrieve information from the database.

Scope of the system

The System focuses totally on mapping a user West Germanic language input String into a SQL query. this is often not yet one more system for language translation nor does it offer benign support for database modifications. It is a database inference system.

Objectives and success criteria of the project

The proposed system is corresponding to an information kiosk. A superlative example would be an outsized store, having thousands of various products on sale. It's an infatuated task to look for a specific product within the target counter, thereby seeking help from the salespersons. Now this method obviates the requirement for manual help and instead allows the users to input their tongue query for an answer. The success largely depends on what percentage words are mapped and recognized by the system and also the character of the database and also the user profiles. The objectives also include minimizing GUI prompts and menus and reducing dependency on user to settle on and choose his choices instead of allow the system to automate its responses, reducing the quantity of iterations required before a user can fixate his enquiry satisfactorily.

Definitions, acronyms, and abbreviations

A language interface to a database (NLIDB) may be a system that enables the user to access information stored in a very database by typing requests expressed in some language e.g. English an example would be: what's the time of arrival of Chennai Express?

IIMETHODOLOGY

The system initializes the JDBC connectivity to the database and stores the meta-data. country language syntax is already defined and stored for reference mapping within the system. The methodology used is corpora-based techniques, which uses collection of words to research and process an announcement and map it accordingly. this technique minimizes user interaction to guide it towards its goals, instead it relies on the corpora-based technique and painstakingly builds up the SQL statement using the input words keyed in by the user in his linguistic communication format. It compares the tokens generated with each of the various data dictionaries and meta-data repositories and reaches a conclusion about user statement objectives. It then eliminates those components of the statement that it concludes aren't required and retains people who have a sway on the ultimate SQL statement string. It does this by string comparison methods and has an inbuilt logic to realize this goal. The SQL statement generator then literally puts together or generates the suitable SQL statement to retrieve results.

A simple case in point would be a railway information kiosk.
The kiosk contains details of trains arrival and departures.
The platforms on which they arrive
The time and date
The destination and source

The user keys in his information request utilizing his natural (English during this case) language construct. No elaborate and annoying crop up menus or motivational menus are proposed because the primary injection to obliterate fabricated queries which are system driven. The system pardons spelling errors and typos.

How it works

The system initializes with the database on take off. It then collects the meta-data and processes them in appropriate formats. The user inputs his tongue query which is passed to a tokenizer that parses the user input and generates tokens which are then passed to an indoor mapper which interacts with a pre-defined data dictionary and semantic rules specific to English people language to assist in matching and mapping the string tokens. there's a database specific rule generator which has the Metadata and people rules also act in unison to help the mapper. The mapper also interacts with the meta-data repository and uses a way of selection and elimination to process the statement whereby those strings specific and required by the SQL generator is allowed while the remainder are filtered. The mapper outputs the processed request to the SQL statement generator which uses this filtered string in unison with the meta-data repository to get the suitable SQL statement to question the database which returns the results to an output generator which formats it and outputs the results to the GUI interface.

What this technique isn't:

Does not portend to develop another language to access the database and focuses to reduce user interaction and output ambiguity and does not concern itself with database updating and alterations. Minimal use of prompts and menu driven approach.

III. Comparisons with existing systems

Prototype NLDIBS like LUNAR, RENDEZVOUS and LADDER had already appeared within the late sixties and early seventies. By the late seventies more NLDIBS gave the impression to engage users in dialogues to assist him/her formulate queries, although semantic grammars helped systems with impressive characteristics the resulting systems proved difficult to port to different application domains. Indeed, grammar had to be developed whenever LADDER was configured for a brand-new application. As researchers began to specialize in portable NLIDBS semantic grammars were gradually abandoned. PLANES and PHILIQA 1 were a number of the opposite NLIDBS that appeared within the late seventies.

CHAT-80 is one in all the simplest known NLIDBS of the first eighties. it had been implemented entirely in prolog. It transformed English questions into prolog expressions, which were evaluated against the prolog database. ASK allowed end users to show the system new words and ideas at any point of your time during the interaction. ASK was a whole information management system providing its own inbuilt databases and also the ability to interact with multiple external databases, email programs and other computer applications. JANUS had similar abilities to interface to multiple underlying systems.

Since then, several commercially available NLIDBS have appeared and a few of them, claim to achieve success. However, NLIDBS are still being treated as research of exotic systems instead of a typical option for interfacing to databases and their use isn't widespread. the event of successful alternatives to NLIDBS like graphical and form-based interfaces and therefore the intrinsic problems of NLIDBS are probably the most reason for the shortage of acceptance of NLIDBS.

Generic linguistic front ends have also appeared, these are general purpose systems that map language inputs to expressions of a logical language ex CLE system. These generic front ends will be was NLIDBS by attaching additional modules that evaluate the logic expression against a database. So, we see that though heavyweight systems have appeared they need been constrained by the requirement for continuous user interaction and are also specific to the database domain which it addresses. The system isn't wholly portable and relies extensively on the users to try to to the mapping for it. this system is simply an inference system, but it demonstrates that even without extensive user interaction by means of menu selection and helping the system SELECT and eliminate choices thus understanding the user requirements, this technique uses a completely unique thanks to generate the SQL query. Furthermore, this method isn't application specific and it's portable to any database domain. The restrictions placed on that are that the database naming standards be more in tune with normal user understanding of the domain which is addressed and also the quantity of relations be limited which is why this can be a more feasible system for the smaller databases and reason behind why it absolutely was named I-Kiosk.

CONCLUSION

The nuances of accessing and retrieving information from a database are critical and therefore the wherewithal’s with respect to those has not been fully met even with advances in database integration and connectivity with front applications. there's a slew of research well into menu and user driven systems that elicit information. The so-called intelligent retrieval is yet to emerge thanks to the incredible costs involved and also the complexities of the language constructs. there is an essence of human intelligence to grasp logical and semantic parts of speech and map it accordingly. No single system has been able to replicate this map accurately in software terms. The more ambiguous a language and more complex it becomes. There are attempts at learning systems that learn from experiences, but the brilliant side is that usage of common words and terms will be filtered to some hundreds and if an elaborate mapping of the definite use of each word with grammatical constraints is evolved such a system could eventually bride the gap. this method tries out possibilities by assuming a restricted set of relations and granting more flexibility from user interaction. Here the onus is on lowering the user help or interaction with the system. The system is inspired to create its own assumptions using corpora-based techniques.

The shortcomings will be there but within the long term the optimizations would negate the inherent constraints and would offer a far better alternative that seek to beat a number of the constraints posed by present systems.

REFERENCES

[1] ACL 1990 proceedings of ACL '90 Pittsburg, Pennsylvania 1990

[2] ACL 1991 proceedings of ACL 1991 Berkley , CA 1991

[3] R.Basili , M.T.Pazienza, P.Velardi, using word association for syntactic disambiguation, 2nd congress of the Italian association for artificial intelligence

[4] B.Boguraev, Building a Lexicon, the contribution of computers , IBM Report , T.J. Watson research center,1991.

[5] M.Brent Automatic Acquisition of subcategorization frames from untagged texts , in ACL 1991

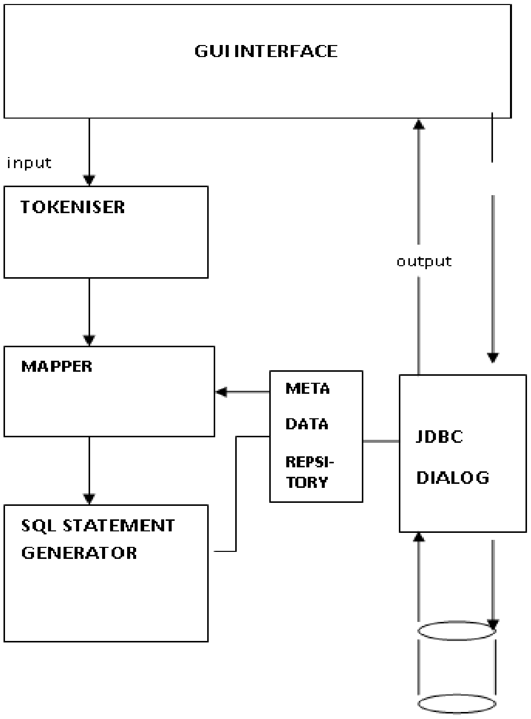
[6] N.Calzolari, R.Bindi, Acquisition of sub categorical frames from untagged texts from ACL 1991

[7] K.W.Church , P.Hanks, word association norms mutual information and lexicography

[8] lK. Church, W. Gale, P. Hanks, D. Hindle, Using lStatistics in Lexical Analysis, in (Zernik,1991).

[9] lS. Fabrizi, M.T.Pazienza, P. Velardi, A corpus-driven clustering algorithm for the acquisition of word ontologies, forthcoming.M.Fasolo, L.Garbuio, N.Guarino,Comprensioneldi descrizioni di attivita' economico-lproduttive espresse in linguaggio naturale,

System Architecture of NLIDBMS Inference System


GUI INTERFACE
input
TOKENISER
output
MAPPER
META
DATA
REPSI-
TORY
JDBC
DIALOG
SQL STATEMENT
GENERATOR