An Approach for Building Natural Language Database using Pattern Matching Technique

Felix Nartey
School of Technology
Blue Crest University College
P.O. Box AN 18392, Accra

Francis Amavi
School of Computer Science
Data link Institute
P.O.BOX 2481, Tema

Isaac Nyameamah
School of Technology
Blue Crest University College
P.O.Box AN 18392, Accra

ABSTRACT
Information assumes a vital aspect of our lives. One of the main supplies of information is databases. There is a blooming interest in databases by researchers and its innovation is growing rapidly. This has subsequently resulted in an influence on the utilization of computers assiduously. Furthermore, most Information Technology (IT) related applications stores, retrieves, organizes, accesses and analyzes information from databases. For instance, retrieval of information from database involves the understanding of database languages like Structured Query Language (SQL). Nonetheless, not everyone is capable of scripting SQL queries, as they may not know of the structure of SQL makeup of the database. The aforementioned issues triggered the building of systems where non-expert users compose their questions in their natural language and obtain the results in the form of a database. The natural language interface to the database (NLIDB) was developed to query relational databases in their natural language instead of working with SQL, an idea provoked to form a new kind of management system. The existing approaches to NLIDBs are challenged with some weaknesses, which include non-intelligent, slow response time and inability to interact with user queries. This paper presents an intelligent NLIDB that uses a chatbot as the natural language interface and a Synthetic Intelligence Markup language (SIML) as the Knowledge-base. The Chatbot is utilized to capture keywords in the user's utterances stored in the knowledge base. The proposed NLIDB structure is based on pattern matching technique employed to communicate with users, manages complications and uncertainties for building natural language queries associated with SQL. Experimental results show that the proposed method gave a promising result on user satisfaction and task completion as compared with existing approaches.

General Terms
Natural Language Processing, Pattern Matching Technique.

Keywords
Natural Language Database, Knowledge Base, Database, Synthetic Intelligent Markup Language.

1. INTRODUCTION
In this contemporary world, computer-based information systems have been extensively used to help many organizations including private, academic and education establishments in handling and processing of their data [1]. Database frameworks are used to control data and the software that permits a computer to implement database roles such as recovering, tallying, removing and altering data is known as Database Management System (DBMS) [2]. These systems are modelled to maintain a large quantum of information. Fetching a bulk quantity of the same type of data is very effective in databases [3]. However, the user needs to master the database architecture completely in building the queries. To access this information, the user is expected to have knowledge in SQL. SQL is comprehensively employed in industry and is supported by major DBMS. In addition, most of the languages used in accessing relational database systems [3] are formed because of SQL models. Only people who have the technical understanding or knowledge of these languages can have access to the information [4] in the database. A naïve end user is usually ignorant in the usage of SQL. Bearing in mind the purpose to retrieve the information, a graphical User Interface must be utilized. This graphical UI requires some essential preparation for utilizing the framework. With the help of this interface, the end-user can question the framework in natural dialect like English, Hindi, French, Chinese etc., and the outcome is shown in a similar dialect [5]. This gives the idea of NLIDB.

In this paper, an intelligent, interactive and robust NLIDB is proposed, to help users retrieve vital data stored in a database (Employee Database) capable of matching to the user requests, aid the user to projected goals and create appropriate information from the underlying relational database by utilizing Pattern Matching technique.

The contributions of the paper are summarized as follows;

1. The Chatbot is used to capture keywords in the user's utterances stored in the knowledge base
2. NLIDB structure is based on pattern matching technique employed to communicate with users, manages complications and uncertainties for building natural language queries associated with SQL.

The rest of the paper is organized into the following sections; Section 2 reviews some related works. Section 3 delves into the proposed system. Section 4 presents the experimental results and finally, section 5 concludes the paper with some recommendations

2. RELATED WORK
For the past thirty years, numerous efforts have been fashioned to build useful natural language interface databases. However, it ended up to be substantially more difficult than originally expected. There have been many studies introducing the hypothesis and applications of NLIDBs, which has been categorized into four kinds of frameworks namely; Pattern Matching system, Syntax based system, Semantic-based system and Intermediate language representation [8].

Among this pattern, matching was most widely used technique. The core benefit of the pattern-matching method is

its simplicity. In such systems, no sophisticated parsing and comprehension of modules are needed, and the systems are simple to execute. One of the best natural language processing frameworks that utilize this method is ELIZA [9], which functions by processing users' response to the scripts. It naturally says differently, rearticulates the statements of the users as questions, and produce appropriate response of the answers to the user. In syntax based and semantic grammar systems, a parse tree is generated for user's queries. The parse tree is directly mapped to database query language. The drawback with these approaches is difficulty in mapping the parse tree to query language [7]. In order to overcome this drawback, intermediate representation languages were proposed. In this, the sentence is first mapped to logic query language followed by the translation of the logical query into general database query [6]. The benefit of this technique is that the framework generating the logic queries is independent from the database and therefore, it is very easy in domain replacements. Despite the achievements accomplished in this area, existing NLIDBs do not assure precise interpretation of queries in natural language to database languages. Motivated by the aforementioned challenges, this paper proposed to build a natural language database using pattern-matching techniques.

3. PROPOSED SYSTEM

It is acknowledged that databases react only to standard SQL queries and it is not easy for an ordinary person to be knowledgeable in SQL querying. In addition, people might be unacquainted with the database structures namely table formats, their fields with matching types, primary keys and more. Furthermore, not all NLIDBs are interactive based. Because of this, an intelligent, interactive and robust natural language interface database to provide a friendly environment to aid users to retrieve relevant and required data kept in a database was designed. The proposed architecture takes natural language in the English language and was established by adopting the Pattern Matching (PM) technique. The primary benefit of the framework is that it conceals the inbuilt complication embroiled in information recovery due to ambiguity. It also enhances quick response to user queries.

3.1 Proposed System Processes

The following process outlines how the intended framework works:

I. A user first enters a query in its natural language. As per this paper, the proposed system only accepts inputs in the English Language.

II. The sentence undergoes word processing to ensure that the words are broken into tokens and into pre-establish formats to suit the user's input.

III. After preprocessing of the queries, they are sent to the Chatbot (Synbot), which is the interactive agent. It receives the queries and forwards them to the SIML interpreter, which is the knowledge base of the system.

IV. The SIML interpreter will check and inspect the tokens of words of the user input at run time in the chatbot to see if there is a pattern match with the specific series of words in its knowledge base. If there is no match, the framework is terminated and the query is discarded and returned to the user through the chatbot as an invalid query by the response element. However, if there is a match between both queries, the SIML (Interpreter) will then further proceed and forward the query to the Database (SQLITE)

V. The database (SQLITE) will then check the information in its table (Employee table) to see if there is a pattern match between its table and that of the tokens of natural language queries as well as the Database scripts (SQL Statements) in the Knowledge base (SIML). If there are no patterns matching, the system is aborted and the query is discarded and sent to the user as an invalid query. However, if there is a pattern matching, the response part of the framework is evaluated and the appropriate natural language query is sent to the user as the final output.

3.2 System Algorithm

Pattern matching technique was adopted and utilized in building the natural language interface database. This technique was employed in developing the recommended system because of its simplicity. For the technique to work, the user must submit a query in its natural language. The natural language query utilized in this paper is English. The English language query is broken down into tokens and then each token is matched with keywords or patterns in the knowledge base and the database. If a match is found, the resultant natural language query answer is produced to the user otherwise, an invalid response is sent to the user.

In this paper, Employee database is used as a case study. To achieve this, the following algorithm is developed to generate the appropriate natural language query to the user

3.2.1Algorithm: Pattern Matching

STEP 1: Input the query in English language

STEP 2: Tokenize the query

STEP 3: If the token (natural language) == Pattern matching (keywords)

3.1 Equivalent predefined Natural language query is forwarded to the SIML interpreter.

Else, go to STEP 6

STEP 4: If pattern==Response in the SIML interpreter (knowledge base)

4.1 SIML interpreter forward the natural language query to database

Else, go to STEP 6

STEP 5 if Response==information in Database table

5.1 SQL is produced and implemented

5.2 corresponding natural language query answer is retrieved and sent to the user

STEP 6: Invalid query generated to the user.

3.3 System Implementation

The proposed system was implemented using Microsoft visual studio, C# programming language, SQLite, SIML and Chabot. The system answers most of the questions posed to it by the users in the English language. The system enables a user to get information about the subject of interest by typing the text in its natural language form. In this paper, the domain is limited to an employee database. Below are snapshots of the performance of the recommended system by showing some inputs with their associated outcomes.

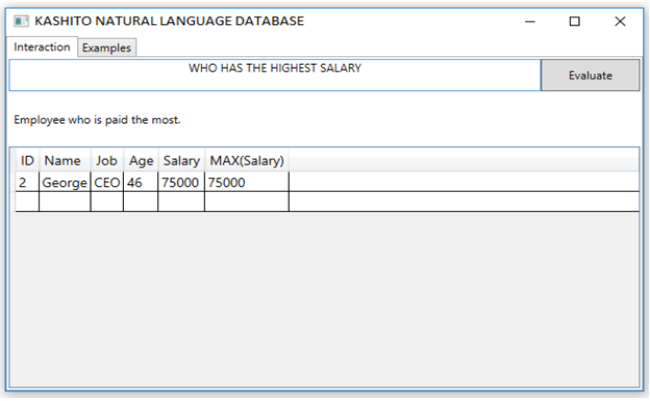

Figure 1: Proposed system in action

From the Figure 1 above, when the user enters an English query, the system checks with the patterns of words in its knowledge base and database by employing the concept of pattern matching technique to generate the appropriate natural language query to the user.

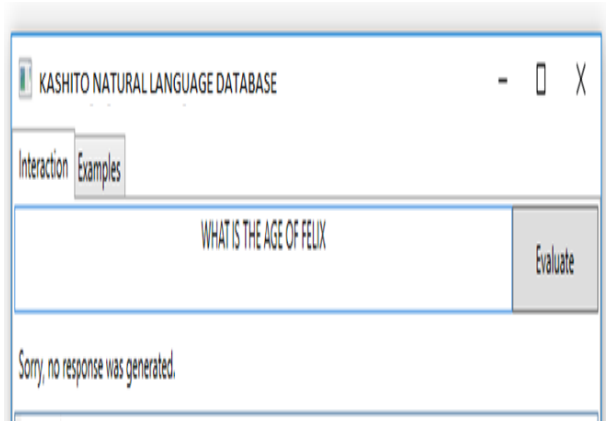

Figure 2: Proposed system in action

In a case where a user enters a query that do not match or exist within the patterns of words in the knowledge base of the NLIDB system or with the information in the Database, the

system should neatly handle this error and show an appropriate message by flagging off an error message as shown in Figure 2.

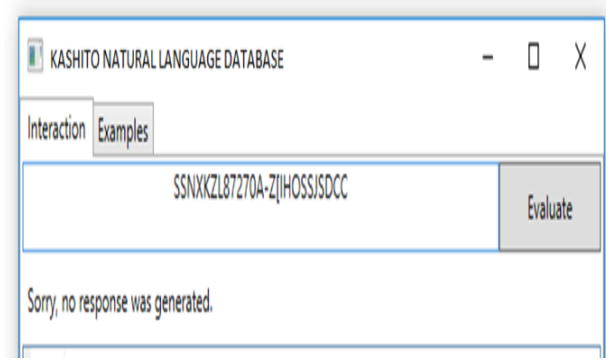

Figure 3: Proposed system in action

In figure 3 above, when the user enters an arbitrarily random query, which does not make any sense in the English language and does not match with tokens of words, it generates an error response.

4. EXPERIMENTAL RESULTS

To evaluate the NLIDB, a list of 50 questions with varying structures on the domain of the Employee Database was created. Out of these questions, 45 questions were correctly processed by the proposed system resulting in 90% accuracy. The 5% errors are due to the lack of coverage of our grammars in the pattern-matching module. Among the 50 questions correctly analyzed, we choose 45 questions with different query-tuples and take their corresponding query-tuples as the set of input to evaluate the Proposed System. The result is shown in Table 1 below

Table 1. Questions with Successful Answers

Type	Number of Questions	Percentage
No Interaction With users	20	40%
Interaction with Users	10	20%
Number of Questions Successfully answered	30	60%

Out of those, 20 questions can be answered automatically without interaction with the user. Most of the errors in the proposed system are pattern-matching errors and answer extraction errors. It is mainly because specific tokens and words from the user in the intermediate representation cannot match or tally with the patterns of sentences in the Knowledge base or with the database information. This is illustrated in Table 2 below.

Table 2. Questions with Unsuccessful Answers

Type	Number of Questions	Percentage
Pattern Matching Errors	10	25%
Answer Extraction Errors	5	15%
Number of Questions Unsuccessfully answered	15	40%

5. CONCLUSIONS

The proposed framework introduces an easy way of developing a natural language interface to database to users especially those with limited knowledge in SQL language using pattern-matching technique. The presented system accepts an English Language request or query by translating it to a standard SQL query and back to the user in a natural language query utilizing a knowledge base and a relational database. Patterns and tokens of words are all contained as a corpus in the knowledge base. The pattern matching technique approach solves the language complexities by simply matching the user requests against scripted patterns in the knowledge base and information from the database. The intended system was implemented using an Employee

Database. The efficacy of the NLIDB system has been successful and satisfactory with the aid of experimental results. The presented system however is only capable of answering queries based on the appropriate database and knowledge base. It is therefore tailor made for only a particular database. Because of this, it can only deal with a limited domain and only a small set of queries can be answered.

In future work, more rigorous techniques like semantic and syntax-based approaches can be utilized and extended to the system in order to widen the knowledge base and cover more queries for the database.

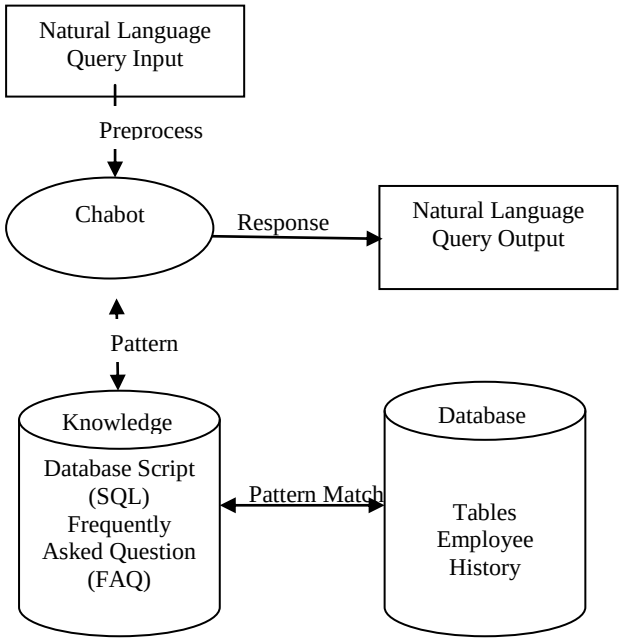

Figure 4: Architecture of proposed system

6. ACKNOWLEDGEMENTS

I wish to thank God for his grace and mercies. My next gratitude goes to my wife Racheal Nartey for her encouragement and support. The paper would not have been a success without the contributions made by Patrick, Ebenezer and my co-authors.

7. REFERENCES

[1] N Nihalani et al, "An Intelligent Interface for Relational Databases", IJSSST, Vol. 11, No. 1, ISSN: 1473-804x online, 1473-8031 print, p30.

[2] Zongmin Ma, "Intelligent Databases: Technologies and Applications", IGI publishing, 320 pages, 2007

[3] Dietmar Wolfram, "Applications of SQL for Informetric Data Processing", Proceedings of the 33rd conference of the Canadian Association for Information Science, 2005.

[4] Siasar djahantighi F, Norouzifard M, Davarpanah S H, Shenassa M H. Using natural language processing in order to create SQL queries. In: IEEE International Conference on Computer and Communication Engineering (ICCCE); 13-15 May 2008; Kuala Lumpur, Malaysia: IEEE. pp. 600 - 604.

[5] Li H, Shi Y. A WordNet -based natural language interface to relational databases. In: IEEE 2nd International Conference on Computer and Automation Engineering (ICCAE); 26-28 Feb. 2010; Singapore: IEEE. pp. 514 – 518

[6] Ashish kumar,Kummar singh vaisha, "Natural Lnguage Interface to Databases: Development Techniques," Elixir Computer Science and Engg Article.*May*.2013.

[7] AmandeepKaur, Parteek Bhatia, "Punjabi Language Interface to Database" communicated to International journal of computer science, WASET (World Academy of Science and Technology).

[8] I. Androutsopoulos, G. Ritchie, P. Thanisch, Natural language interfaces to databases – an introduction, Journal of Natural Language Engineering 1 (1) (1995) 2981

[9] Weizenbaum, J. (1966). ELIZA-A computer program for the study of natural language communication between man and machine, Communications of the ACM, Vol. 10, No. 8, pp36 45.