Sri Lanka Association for Artificial Intelligence (SLAAI)
Proceedings of the fourth Annual Sessions
31st October 2007 - Colombo

NLP - Based Expert System for Database Design and Development

U. Leelarathna [1], G. Ranasinghe [1], N. Wimalasena [1], D. Weerasinghe[1], A. Karunananda [2]

Faculty of Information Technology, University of Moratuwa, Sri Lanka.

cinco.publish@gmail.com[1] , asoka@itfac.mrt.ac.lk[2]

Abstract

Database designing and development involves a sequence of tasks including extracting the requirements, identifying the entities, their attributes, relationships between the entities, constraints, drawing a conceptual schema, mapping the ER diagrams to the database schema, and eventually developing the database. As such database design and development has become a tedious task for novice person. In addressing the above issue, we propose a Natural Language Processing enabled Expert Systems, which accepts textual domain descriptions and generate a relational database schema followed by a database. The entire NLP enabled system can be customized to link up as a front-end for any database management system. The system has been developed using Prolog, Flex and C#.net.

1. Introduction

Database and Database systems have become an essential component in modern day life. Most of the applications that we come across today involve large amount of data, making database systems the first choice in storing, processing, and disseminating data.

Designing a database requires a good knowledge and skill on database concepts. After a thorough requirement analysis, the conceptual schema design should be done in a high level data model such as Entity Relationship Model. Then this should be mapped in to a schema using a data model such as Relational Data Model.

Hence the user must have some idea of ER designing and mapping it to Relational schema in order to design and implement a database for an application. This makes designing and implementing a database a difficult task and complicated process for a user who has less knowledge on databases. The expert knowledge and skills is always preferred in this context to develop a superior database. However, consulting and depending on expert knowledge all the time, brings out certain issues, since expert knowledge is expensive and not always available.
If there's a possibility that users can represent the requirements in their natural language rather than as ER diagrams or Relational schema and the system can extract the user requirements itself, draw ER diagrams and implement the database automatically, then life will be easier for the user. Even a novice user with less knowledge on database design will be able to develop a database easily using such a system.

Based on this idea we have been working on developing an intelligent system to design and develop a database using expert knowledge with minimum user interaction. The proposed system is a **Natural Language Processing enabled Expert system.** It takes the requirements specified in natural language as the input, process it and identify entities, attributes, relationships and constraints using an expert system, draw the Relational schema and develop the database for the user. This paper presents our approach to develop this system.

The rest of this paper is organized as follows. Section 2 provides an overview of the current approaches to database development. Section 3 describes the design of the system. Section 4 presents how the system works and finally Section 5 presents a discussion with a note on further work.

2. DB Systems-Design & development

Many tools have been introduced to simplify the process of database design and development. Most of these tools facilitate the conversion of ER diagrams to database. These designing tools take the user drawn database designs (e.g. ER diagram), and generates the SQL code appropriate for the database.

Most tools for database design and development facilitate the conversion of ER diagrams to database. These designing tools accept the user drawn database designs (e.g. ER diagram), and generates the SQL code appropriate for the database. For example, Clay [10] is a modeling tool of this type that acts as a plug-

in for Eclipse. DBVA for JBuilder for Windows 2.0 [11], DBVA for IntelliJ IDEA for Windows 2.0 [10] are some of the other similar tools to build database systems.

There are also several tools that use Expert systems for database design and development. Among others, Generalized Expert System for Database Design (GESDD) [3] is one such system. It is made up of two parts:

1. An expert system for generating methodologies for database design, called ESGM

2. An expert system for database design, called ESDD.

Using ESGM, database design experts can specify different design methodologies or modify existing ones. The database designer uses ESDD to design a database. It supports several well-known data models, namely, the hierarchical data model, the network data model, or the relational data model. However, GESDD is a menu-driven system and no support for interaction though natural languages.

SECSI[2] is yet another software system, which uses an expert system for database design. This system generates a specific semantic network representing the application from an application description given with either a subset of the natural language, or a formal language, or a graphical interface. It uses a set of design rules to complete and simplify the semantic network to reach flat normalized relations.

It is understood that although experts systems technology has already been used to develop software tools for design and development of databases, such systems are mainly targeted for experts in database design, but not for novice designers with little experience. Further, although SECS like systems enables NLP supports, such systems do not provide adequate facilities for novice database designers to use the system.

In view of that we propose to design and develop NLP enabled expert system that can be used by a novice database designer.

3. Proposed NLP enabled ES

We have designed the proposed system with 5 modules, namely, NLP Parser, English Morphological Analyzer, Dictionary, Knowledge base and Output generator. Figure 1 shows the top-level design of the Natural Language Processing enabled Expert system**.** In broader sense, NLP enabled expert system comprise two major modules, namely, experts system module and output generator module. The role of each module is described below.

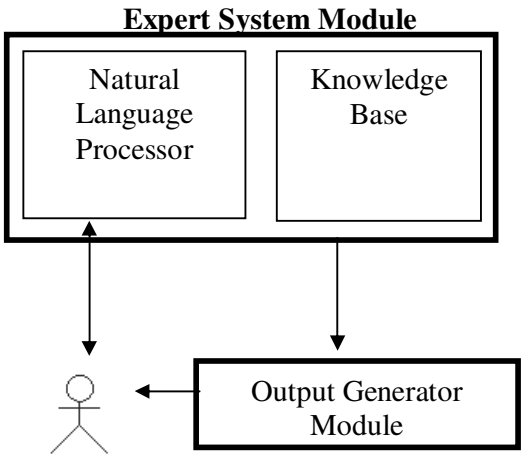

Figure 1: Top Level design of NLP enabled ES

3.1 Expert System Module

The role of the Expert System is two fold:

- Use a Natural Language Processing parser to obtain user information in the best possible way.

- Use the Knowledge Base of expert's knowledge to develop the relational schema of the database.

It is a known fact that the domain descriptions provided by the users are generally incomplete and rather ambiguous. In such a case, with the use of features of the expert system technology, our system is able to handle incomplete information by asking questions from the user. Also after the system design and develop the database, user might want to know reasons or explanations on certain system decisions. In this case system supports reasoning and explaining.

3.1.1 Natural Language Processor

The Natural Language processor in the Expert System Module collectively comprises a tokenizer, a parser, an English Morphological Analyzer and English language dictionary. Figure 2 visualize the Natural Language Processor.

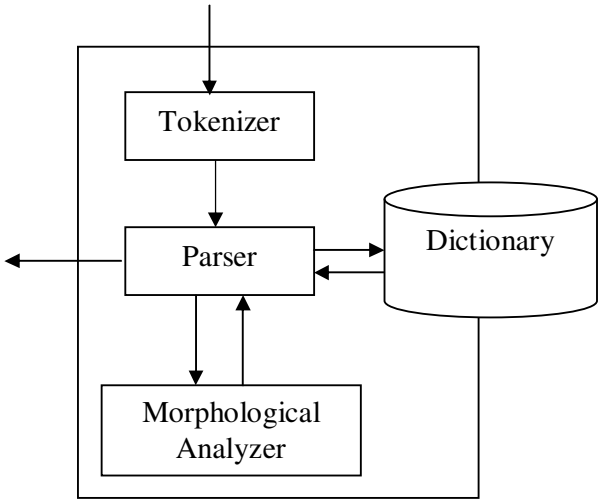

Figure 2: Natural Language Processor

3.1.1.1 Tokenizer

The natural language processor contains an efficient tokenizer written in ISO standard Prolog. Tokenization is the process of breaking a text file up into words and/or other significant units. The tokenizer takes each sentence in the text file one-by-one and breaks them in to separate tokens For example; it breaks the input string "this is an example" into the series of tokens:

[this,is,an,example]

3.1.1.2 English Parser

The English Parser embedded in the expert system module receives tokens for each sentence of the domain description, and identify the parts of speech in the sentence [9].

Each sentence of the file is tokenized and stored into a variable. The value of the variable of a given sentence is passed to the parser. The parser contains the Definite Clause Grammar (DCG) rules [4, 5] that identify the parts of speech. At this stage of the project, we assume that the sentences input to the system, has only present tense and no spelling and grammatical mistakes. The parser in this project is linked up with an English dictionary to get parts of speech [1] information of the words in the input sentence. Win-Prolog is used to develop the parser in Flex environment.

3.1.1.3 English Morphological Analyzer

The natural language processor also deals with an English Morphological Analyzer that identifies the different inflection of the English words [8]. The inflections of each word are attached to the list which is sent to the Expert System. This is also developed using Win-Prolog in Flex environment.

3.1.1.4 Dictionary

NLP system uses a dictionary such as English word dictionary to identify the parts-of-speech of a given tokenized sentence. English word dictionary contains English words and the lexical information. English concept dictionary contains synonyms, anti-synonyms and general knowledge about English words. To identify the parts-of-speech of each tokenized sentences our system uses the WordNet dictionary [6, 7].

WordNet is a lexical reference system, developed by the University of Princeton. Its design makes the use of dictionaries more convenient. Data from WordNet can be used as input for various applications. It provides a database, written in Prolog.

3.1.2 Knowledge Base

The knowledge base consists of the rules that are used to identify database relations and relationships. This is the reservoir of expert knowledge to design and develop databases. It identifies the database entities, relationships, relations and attributes by applying the rules and designs the database relational schema.

The knowledge base has been implemented as a production system that uses rule base knowledge representation. The inference engine of the expert system uses the forward chaining strategy to discover the appropriate design of a database for a given problem. The inference engine also uses the priority base conflict resolution for exploring the knowledge base. The output generated by the expert system is written in a special format to a text file. The Expert System module has been developed using Knowledge Specification Language in Flex environment.

3.2 Output Generator

The output generator reads the output of the expert system and graphically represents the relational database schema to the user. At present output generator provides graphical output separately from the NLP interface. The output generator module also reads relational scheme and generate SQL commands, which will be executed by a respective

database management system. The output generator module can be customized to link up the expert system module with an available database management system. This module has been implemented using C#.NET in Visual Studio.Net framework.

4. How System Works

In this section we describe how the system works for a given input paragraph. Assume that the user has input the sentences **'Department has employees. Employee has id, name, department, age, address'** through NLP interface.

Then the expert system accepts the input stream and sends through the parser and generates the following output.

Sentence 1- [Department/NN, has/VB,
employees/NNS/.]

Sentence 2- [Employee/NN, has/VB, ID/NN,
name/NN, department/NN,
age/NN, address/NN/.]

Sentence 3- [Department/NN,has/VB
ID/NN,name/NN,ID/NN,name/NN,
noofemployees/NN].

The above output has the semantics as
Sentence 1- Department is a noun, has is a verb, employees is a plural noun.

Sentence2 - employee is a noun, has is a verb, ID is a noun, name is a noun, department is a noun, age is a noun, address is a noun.

Sentence 3 -department is a noun, has is a verb, ID is a noun, name is a noun, noofemployees is a noun

The expert system reads the input list and generates the output by identifying the relations, attributes, relationships etc. According to the example the system identifies the first sentence as a description of a relationship and the second sentence as a description of relation. Figure 3 shows the output generated by the expert system which is a text file.

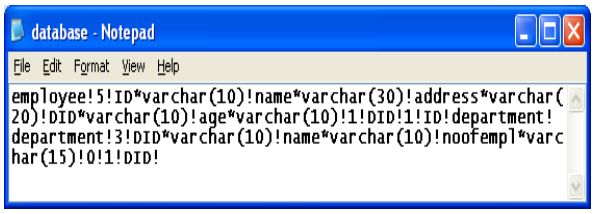

Figure 3: An output generated by the ES

According to the above information the output generator of the module draws the relational schema and develops the database with two tables, namely, department and employee. Figure 4 shows the relational schema generated by the output generator module to indicate the relation via department ID (DID).

.

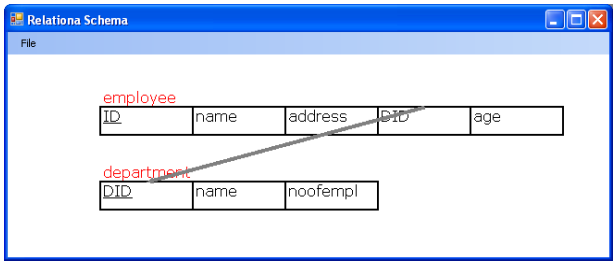

Figure 4: Relational schema by output generator

5. Discussion

The objective of our project was to develop a Natural Language Processing enabled Expert System for database design and development. The proposed system simplifies the database design and development process and unlike many existing systems for database development, it can be used by a novice database designer. The system accepts textual domain descriptions and generates a relational database schema followed by a database. At present the system can handle domain descriptions written in simple present tense. Since the system has been developed as an expert system, it is possible for the system to communicate with the user so as to receive complex versions of descriptions in a simpler form. In addition, since the system is able to provide reasons for the generated designs, the user will be in position to receive justifications for proposed designs. As such NLP technology together with the Expert system technology has made the database design and development process more interactive and more flexible even for the use by novice persons.

This system can be further improved by integrating the output generator and the expert system interface in to a single user interface, so that the user is able to view the relational schema and do modifications

when necessary. We can also provide the user with the facility to customize the output generator module to be linked up with a chosen database management system. Incorporation of more experts knowledge for the knowledge base and additional knowledge for improving the NLP module will improve the effectiveness of the system.

6. References

[1] Aravind K. Joshi, B. Srinivas "Disambiguation of Super Parts of Speech (or Supertags): Almost Parsing", 15th International Conference on Computational Linguistics, Kyoto, Japan, August 1994.

[2] Bouzeghoub M., Gardarin G., Métais E., "Database design tools: an expert system approach", Proceedings of the 11th international conference on Very Large Data Bases - Volume 11, Stockholm, Sweden, 1985 pp 82 – 95

[3] Dogac A., Yürüten B., Spaccapietra S. "A Generalized Expert System for Database Design", http://portal.acm.org/citation.cfm?id=63389

[4] Doran C., Egedi D., Hockey B. A., Srinivas B., Zaidel M., "XTAG System – A wide coverage grammar for English.", 15th International Conference on Computational Linguistics, Kyoto, Japan, August 1994.

[5] Egedi D, Martin P., "A Freely Available Syntactic Lexicon for English.", 15th International Conference on Computational Linguistics, Kyoto, Japan, August 1994.

[6] Gangemi A., Guarino N., Oltramari A., "Conceptual Analysis of Lexical Taxonomies: The Case of WordNet Top-Level", 15th International Conference on Computational Linguistics, Kyoto, Japan, August 1994.

[7] Hristea F., "On the Semiautomatic Generation of WordNet Type Synsets and Clusters", Journal of Universal Computer Science, (12/4/02)

[8] Schmid H., "Part-Of-Speech Tagging with Neural Networks.", 15th International Conference on Computational Linguistics, Kyoto, Japan, August 1994.

[9] Turian J.,Wellington B., Melamed I. D., "Scalable Discriminative Learningfor Natural Language Parsing and Translation".

[10] Azzurri Ltd. Japan, http://www.azzurri.jp/en/software/clay/index.jsp, Down loaded on 20th May 2007

[11] http://www.surfpack.com/downloads/DBVA-for-JBuilder-for-Windows/23404.html, Downloaded on 20th May 2007

[12]http://pcwin.com/Software_Development/DBVA_for_IntelliJ_IDEA_for_Windows/index.htm, Down loaded on 20th May 2007