SEMANTIC BASED NL FRONT-END FOR DB ACCESS: FRAMEWORK ADOPTATION FOR SWAHILI LANGUAGE

Narin'yani Alexander [1], Muchemi Lawrence[2]

[1]Russian Research Institute for Artificial Intelligence
P.B. 85, 125190, Moscow, Russia, Tel (495) 158 9430
[2]Lecturer, School of Computing and Informatics
University of Nairobi
Tel. No: 254-2-4444919 ext 103
Location: Chiromo Campus, ICT/SCI Bldg, Rm 103
e-mail: [1]narin@aha.ru, [2]**lmuchemi@uonbi.ac.ke**

ABSRACT: The forty years of Natural Language (NL) processing research have not reached the necessary threshold to move results from laboratories to practical realities of commercial world. This unsatisfactory state in NL processing has been a result of an erroneously chosen philosophy and methodology that considers human language understanding as a syntactic problem while significantly ignoring semantics and pragmatics aspects of NL. These have a key role in comprehension of NL texts within narrow subject domains such as natural language queries (NLQ) to databases.

In this paper we consider some NLQ paradigm which integrates the following principles:

A. Semantically-oriented approach to NL analysis.

B. Efficient use of a problem-oriented knowledge within the NL analysis/understanding process

C. A bottom-up distributed self-organizing parsing which would be based on data-driven type of control.

The NLQ paradigm is presented as a framework. Evaluation has been done by development and testing of a prototype resulting from the above framework. The prototype has been successfully tested for English and Russian NLQ. Through examples we show that this framework can successfully be adopted for Swahili NLQ as well.

Keywords: natural language querying, pragmatics, semantics, parsing

1 INTRODUCTION

The forty years of the Natural Language Processing (NLP) research, including as well research into Natural Language (NL) front-end to Databases, makes one wonder whether the current unsatisfactory state-of-art in NLP reflects our insufficient knowledge and practical abilities or is a result of erroneous conceptual paradigms and methodologies. While the former is an inherent shortcoming, the latter seems to be the more probable reason. It therefore necessitates to be considered afresh with respect to the problems of NL understanding in general and of NL front-end design in particular.

In this paper we consider some paradigm which integrates the following principles:

A. Semantically-oriented approach to the NL analysis.

B. Efficient use of the knowledge extracted from the problem area within the NL analysis/ understanding process

C. Bottom-up distributed self-organizing parsing which is based on the data-driven type of control.

The integration and implementation of these principles form a sound basis for NLQ processing and is envisaged to overcome the crucial threshold between experimental

systems and practicable applications. This paper discusses these principles in detail and shows evaluation of the results by highlighting successes of several NLP projects carried out at the Russian Research Institute for Artificial Intelligence [4 - 11]. We also demonstrate that the principles can be adopted for Swahili NLQ parsing.

2 NL-INTERFACE TO DATABASES, WHAT IS IT?

Any NL-interface works as a mediator between the user and the database. It translates the query formulated in NL into a formal representation, matches it to the database and passes the appropriate information from the database to the user. See Fig. 1 below.

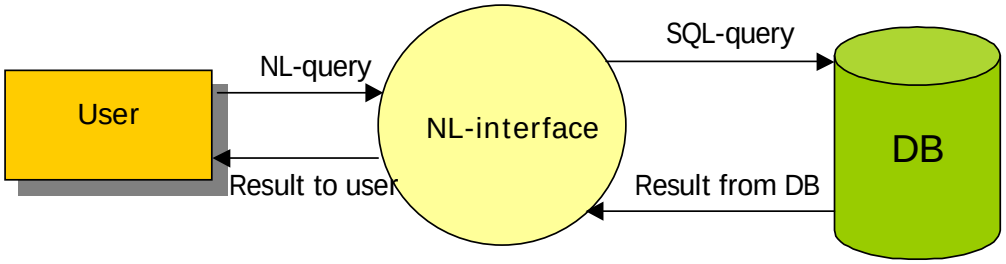

Fig. 1 Context Diagram of a NL Interface

It's important that the query is not restricted in any manner. For example one can ask for a price list in an e-store:

Query in English

"What is the cost of a red Samsung or Roventa Vacuum cleaner?"

Same Query in Swahili

" Nipe bei ya mashine ya kufyonza vumbi aina ya Samsung au Roventa yenye rangi nyekundu?"

Same Query in non-standard (broken)Swahili

"Saidia bei ya vakiumu Kilina nyekundu aina ya Samsung ama Roventa"

(Help me with price of red vacuum cleaner of type Samsung or Roventa)

In terms of SQL, the query would be as follows:

SELECT COST

FROM VACUUM_CLEANER_TABLE

WHERE (MAKE = "SAMSUNG") OR (MAKE = "ROVENTA")

3 THE CHALLENGES

This section addresses some of the key problems for a number of research projects. A substantial percentage of world's native languages have had little or no research in the area of natural language database access. Gathering from the reviewed literature, there are still some outstanding knowledge gaps that hinder development of the NL DB access systems. Key among them includes computerized phrase identification mechanisms from texts, conversion of the phrases to Structured Query Language (SQL), unsatisfactory

parsing efficiencies and issues of disambiguation and pragmatics among others. Swahili is not an exception to these and this paper attempts to address some of these issues.

The knowledge gaps stated above are further complicated by presence of diverse speakers from different backgrounds in East Africa which makes use of non-standard Swahili very common. Attempts to syntactically parse unrestrained text will fail miserably because syntactic parsing assumes a language that is strictly following some rules. Results from 'broken' language will be unsatisfactory. As shall be argued below a semantic approach is superior here to a syntactic one and is therefore employed in our experiments.

To analyze means to understand

To understand a person **A** the following conditions must be met:

- to know sufficiently enough the language **A** uses and
- to orient oneself sufficiently in the subject domain that **A** is communicating about.

One may be a native Swahili, English or Russian speaker but may never understand the text if he has not heard the subject or the domain before. In the above two principle conditions, the word "sufficiently" means "just enough but not necessarily perfect", i.e. some vicinity of "perfection". It's interesting that these two vicinities are inversely proportional: *the better you orient in the subject domain* (the narrower the vicinity of "complete knowledge" is), *the more imperfect your experience in the language of **A** may be* (the more wide the vicinity of "complete knowledge" of **A**'s tongue) and vice versa.

These intervals of vicinity resemble to an extent, the intervals of uncertainty from Heisenberg uncertainty principle in the quantum mechanics. This relation works only to some degree because it looses meaning when one of these two estimations is close to zero. Thus even best knowledge of a subject cannot ensure comprehension of a phrase in a language one knows nothing about. At the current state a computer can be an expert only in a very narrow subject domain where it is specially prepared to orient within corresponding semantics and pragmatics by the problem-oriented knowledge based apparata.

It means that out of such subject domain it is impossible meanwhile to get a sufficient level of understanding to convert a text into its meaning which may allow computer to use it as input information, for example as a concrete query to a concrete database.

Semantically- vs. Syntactically- oriented Processing

The above argument leads to a conclusion that our main instrument in the NL text understanding process should be semantics and pragmatics of the subject domain that forms a frame within which we are communicating with our computer.

The next natural conclusion is the following: the traditional syntactically-oriented approach is an absolute dead-end for the NL understanding process. This has been proved by complete failure of all research which has used it as a basis within a wide subject domain. It should be replaced with the semantically-oriented philosophy which can be stated as follows:

> *"When analyzing a NL text input, it is necessary to use its lexical semantics within the subject domain to reconstruct its probable meaning. Only if this meaning has several variants then it would be useful (as local as possible) to turn to syntactic aspects of the text to resolve this ambiguity."*

For nearly thirty years one of the authors of the paper has discussed the conflict of these two paradigms with many colleagues from the traditional syntactic-based NL projects in various countries. The main point of argument has been that the English we speak is not native to most of us and therefore is occasionally grammatically incorrect. We make a lot of grammatical errors but nevertheless we are able to understand each other quite well. It wouldn't be possible if our inner comprehension process would be syntactically-oriented. We simply comprehend the meaning basing on semantics and pragmatics of the topic. If processing was syntactic no one would comprehend broken language.

At least during the last decade the global syntactically-oriented project has discovered the paradigm's ultimate inadequacy.

An example

For an illustration we may consider NL query to some personnel database and its component representing the condition (wage = $4,000). This component may be expressed with a spectrum of NL textual form, for example:

1. *... with **wages** equal $4,000... Swahili{ ...na **ujira** takribani dola 4000…}*
2. *... whose **salary** is $4,000 ... Swahili{ ... **mshahara** wake ni dola 4000…}*
3. ***... earning** $4,000 ... Swahili{ ...**mapato** ya dola 4000…}*
4. *... being **paid** $4,000 ... Swahili{ ...**analipwa** dola 4000…}*
5. *... having $4,000 **salary** ... Swahili{ ...ako na **mshahara** wa dola 4000…},* etc.

In spite of difference in their syntactical forms these expression include:

a. two obligatory components which represent the meanings of '**wage'** in various forms and **$4,000** (the latter, of course, may be represented also with lexical expressions)

b. an optional component representing the meaning of the '=' relation (in the last three examples it is present by default, i.e. by absence of other relations - '>', '<=', etc.).

All other words carry no semantic information in our examples and can be excluded.

If we take into account the pragmatics of the database query, the 'wage' semantics relates to an attribute name (let us denote it A_w) and $4,000 relates to a value of this attribute (V_w). Obviously, the index of attribute (A) or value (V) may relate to not one but to several attributes - for example, $4,000 may relate to different attributes with the values in dollars and the range covering this sum. It leads us to the following fragments of a formal representation being obtained after lexical analysis and replacement the input words with vocabulary information projected onto the given application:

1. $A_w = V_w$
2. $A_w\ V_w$
3. $V_w\ A_w$

That means that only three simple rules (the 'i' indexes should relate to the same attribute)

$$... A_i R V_i ...$$

... A_i V_i ...

... V_i A_i ...

cover majority of possible text expressions of a simple predicate of the *Relation (Attribute, Constant)* form in any database query.

We should add here the rule ... A_i ... where 'i' relates uniquely to one attribute (for example, 'red sedan' for a database including only one attribute allowing the 'red' value and one with the 'sedan' value) to make this set complete to understand ($A_{color} = V_{color}$) & ($A_{type} = V_{type}$).

This is just a simplified illustration of the principle. Real query analysis rules are more complex than those. But many times easier, more reliable and domain-independent than any syntactically-oriented set of rules which may be proposed for the similar function. Of critical importance is that the rules are practically the same for Swahili, English or Russian.

Bottom-up vs. Top-down

The key argument about efficiency of parsing usually is "Bottom-up or Top-down?" i.e. "Direct or reverse" organization of logic inference, planning and parsing? In the 70s, 80s and 90s this choice had been made in favor of the second alternative (reverse). At that period it seemed obvious that the direct process would lead to a combinatorial explosion. Backtracking appeared to be a magic key to any inference problem, or at least to many of them. Later it became clearer that parsing can not be a source of combinatorial explosion. A limited number of elements in the input text and highly selective principles of combining them do not permit the process to produce too many variants of the target structure and imbedded constituents.

The bottom-up organization of the analysis ensures much higher efficiency at the expense of elimination of the backtracking problem. It also allows ultimate decentralization of the process control. The latter makes possible to localize each action of parsing and permits constructing the "lingware" with minimum interdependencies restricting the non-deterministic flow of such actions.

4. THE FRAMEWORK

We have considered above three principles which define the paradigm discussed:

- Semantically-oriented approach to the NL analysis.
- Efficient use of a problem-oriented knowledge within the NL analysis / understanding process (pragmatics approach)
- A bottom-up distributed self-organizing parsing which would be based on data-driven type of control.

The above principles were put into a framework of a technology whose structure is represented by the system architecture of the prototype discussed below.

The integration and implementation of the above principles were evaluated for a wide spectrum of languages. The project was carried out at the Russian Research Institute for Artificial Intelligence under a project called 'InBASE'. The technology is presented in

the sections below. It is apparent that if these experiments are applied to Swahili, the results would not be different.

System Architecture

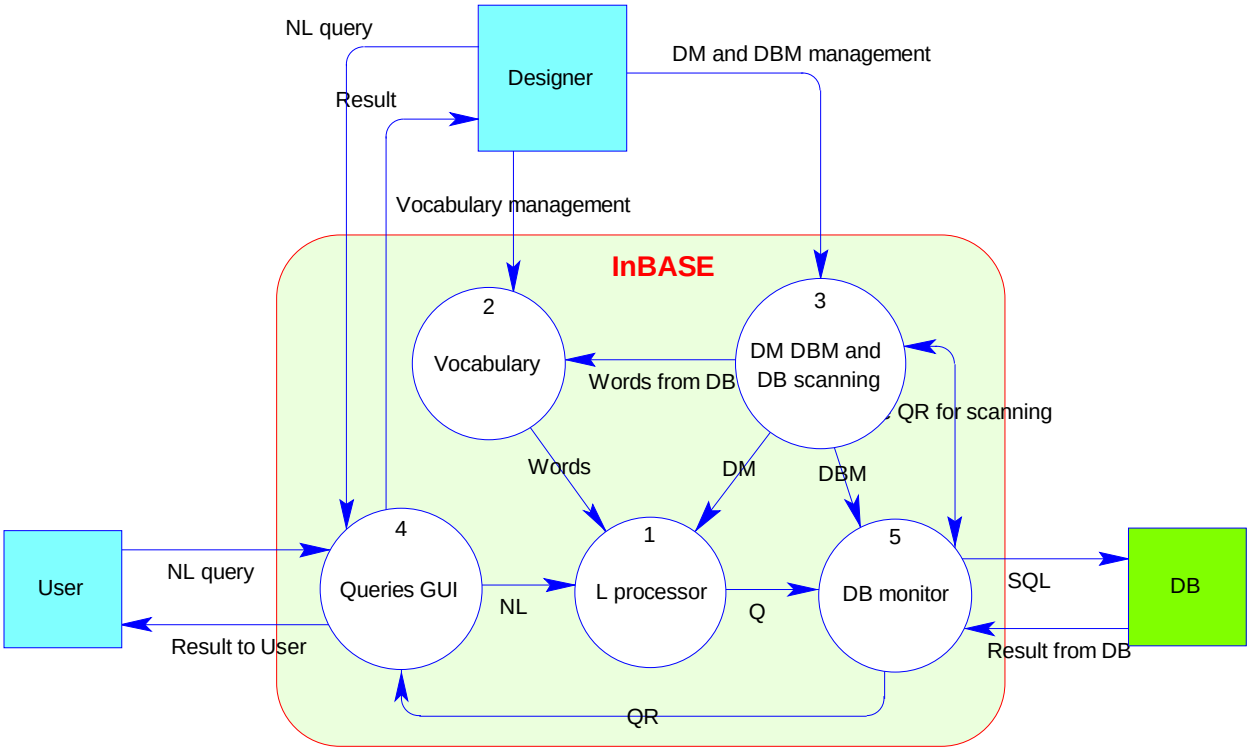

Fig 2. The general Framework of the System.

The system consists of the following components (see fig. 2):

§ A linguistic processor (L-processor), which translates NL-queries to their intermediate formal representation in Q-language form (1)

§ A design monitor for adaptation of the general-purpose NL shell to a particular database (2, 3)

§ A query-answer monitor, that sends queries to the L-processor and displays answers (4).

§ DB-monitor (5), which translates Q into SQL and processes result from DB.

The system builds an intermediate formal query representation expressed in Q-language - a special language similar to the simplified object query language OQL. To build the Q-query representation the system uses the Domain Model (DM). The Q-query is converted into SQL or into any other query representation language of concrete relational model (RM) of the data base.

The answer to the query from DB can be formed as QR-representation which is just the result of the information retrieval in terms of DM. All these representations - Q-query, DM, RM and QR have their XML representations.

The system operates in two modes:

- ð Constructing Natural Language Front-end (NLF) to a concrete database (the designer's mode);
- ð Query processing (end-user's mode).

With some experience, a designer creates a basic NLF for correspondingly simple DB during several hours of work and to turn it into reliable NLF of the full capacity in a few days.

The second mode is intended for an ordinary user and is based on NLF designed at the first mode for processing of NL queries to the database and presenting results of the database search.

Designer's Mode

To design an NLF means to adapt the general purpose L-processor to the target database structure and its content. The main steps of the NLF design process are as follows:

- ð to create the specifications of Domain Model (DM) and Database Model (DBM)
- ð to form the vocabulary
- ð To debug the NLF.

The DM and DBM reflects the database tables structure and links between the tables as well as a set of other database parameters (semantic characteristics) being important for creation of the NLF and its performance. Necessary information concerning the content and individual features of the database is provided by the designer.

The content of the vocabulary of the constructed NLF consists of three parts.

The first one includes a general-purpose lexicon - it forms the initial vocabulary of any L-processor (*list, find, more than, not, except, average,* etc.), about one thousand words.

The second - the largest - part of the vocabulary content is provided by scanning the database and added to the vocabulary as *values* of corresponding DB *attributes.*

The third part of the lexicon consists of words specific to the database problem domain. This includes words and phrases referring to attributes (for example, *salary, wage, date of issue*), synonyms to values taken from the database (for example, *woman* for *female* denoting sex in the database).

Before the L-processor becomes ready to analyze NL queries under the User's Mode, the designer has to check whether the lexicon is complete. This should be done on the basis of a set of typical queries to the database, which are chosen by the designer himself and/or provided by the customer to test the NLF under construction. The debugging makes it possible to extend the lexicon and to evaluate a reliability of the created NLF.

User's Mode

Since the goal of NLF is free and unrestricted NL access to the content of user's database, the screen in this mode is split into three major areas: an area for typing query, an area for displaying the SQL statement corresponding to the query meaning, and the response structured as data from the database. For example, all the synonymous queries to our sample personnel database:

- *Salary of single mothers*
- *single mothers, their salaries*
- *How large are wages of mothers which are unmarried or divorced?*
- *What is the salary of single women with children?*
- *How much earn single mothers?*

are translated to an SQL statement, which is interpreted, and its results are presented to the user (fig. 3). Note that the database response contains more information than has been requested, since the designer of the interface provided Name and Post of Employee as default identification data.

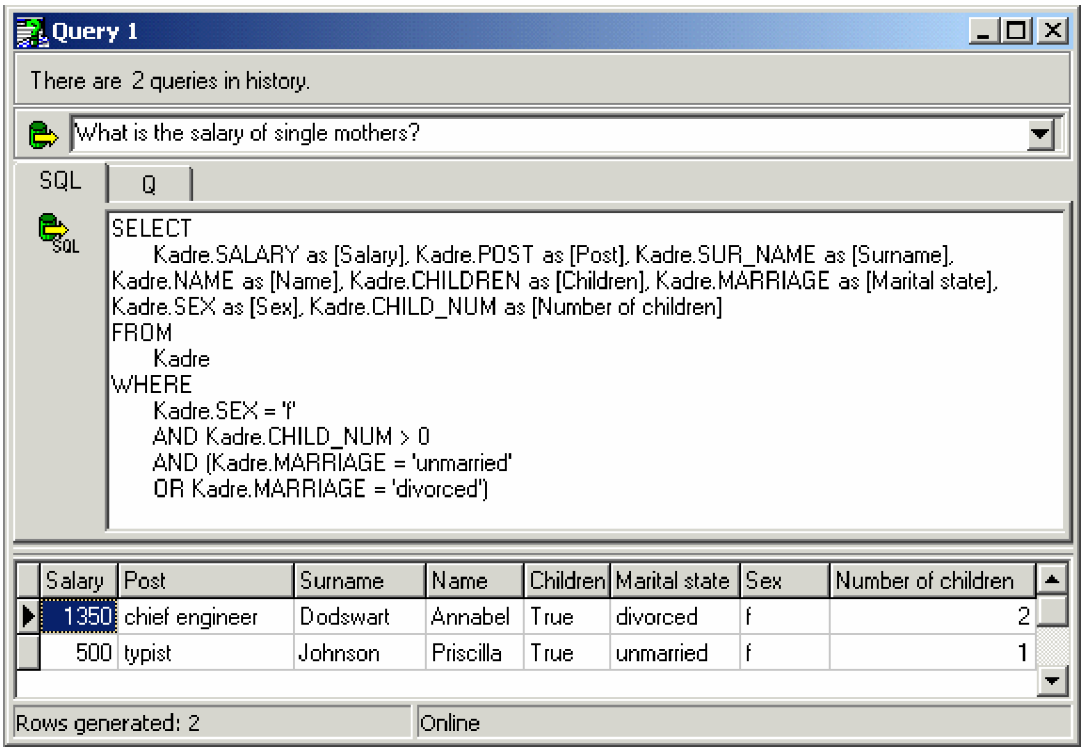

Fig 3. SQL representation of a Natural Language query

Even after careful designing and debugging, an incidental unknown word may appear in some query under the user's mode. The new word can be added by the user to the vocabulary on-line:

- ð by specifying its semantics in terms of attributes of the database;
- ð as a synonym to some other word in the vocabulary.

The experiments explained above and the results obtained are satisfactory for all languages tested. These however can be tested for any other language by researchers. Testing for Swahili language can specifically be interesting as no other languages originating from African continent have been tested up to now. Research is still going on for Swahili NL.

CONCLUSION

Natural language understanding is an area of research with a long history of a slow and steady progress. Experience has taught us that the NLQ processing paradigm that has relied on syntax parsing has not been adequate. Researchers should shift focus from the syntax orientation to semantic orientation with only a limited syntax processing for disambiguation purposes only [2, 3, 12].

This paper has underlined the importance of semantic processing for better natural language understanding within limited subject domains. It has also demonstrated that taking pragmatics into account is also necessary within the NL analysis through the use of problem oriented knowledge [4 - 12].

The paper has stressed necessity of improving parsing efficiency by a bottom-up distributed self-organizing which would be based on data-driven type of process control. Finally the paper has demonstrated an integration of these principles into a framework which has been evaluated through a successful prototype.

BIBLIOGRAPHY

1. Genereux M., (2005). Efficient Semantic Parsing Of Conversational Speech. Proceeds of 2nd Baltic Conference on Human Language Technologies, April 4-5, 2005,
2. Jung, H. and Lee. G., (2002). Multilingual question answering with high portability on relational databases. Proceedings of the 2002 conference on multilingual summarization and question answering - Volume 19. Pages: 1 – 8. Published by Association for Computational Linguistics, Morristown, NJ, USA.
3. Jurafsky D., and Martin J.H., (2003). Speech and Language Processing - An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition. Pearson ed. Series, New Delhi, India.
4. Narin'yani A.S. AI work in the Computer Center of the Siberian Branch of the USSR Academy of sciences. - In: "Machine Intelligence", n.9, Ellis Horwood Ltd., Publishers, Chichester, 1979.
5. Narin'yani A.S. Interaction with a limited object domain - ZAPSIB Project. Proc. of the Int.conf. "Computational Linguistics- 1980", Tokyo, 1980
6. Narin'yani A.S. T.M.Yahno. A "bottom-up" procedure as non-deterministic parallel process. - Computers and Artificial Intelligence, Bratislava, 1982, vol. 1, n.4.
7. Narin'yani A.S. Verso un modello integrato del linguaggio naturale. - Ricerche di Psicologia, Milano, 1983. V.11, n.25.
8. Narin'yani A.S. ZAPSIB Language Processors// T.A.Information, v. 24, n.1, Paris, 1983.
9. Narin'yani A.S. Intelligent Software Technology for the New Decade. - Communications of the ACM, vol.34, 1991, n 6, p.60-67.
10. Narin'yani A.S. The problem of NL query to database is solved // Proc. of DIALOGUE'95 Intern. Workshop, Kazan, Russia, 1995, (in Russian)

11. Narin'yani A.S To interact means to understand each other. In: Proc. of 9-th International Conference «Speech and Computer (SPECOM'2004)» & INTAS Strategic Scientific Workshop. SPb 2004.

12. Rojas J., Torres J. A Survey in Natural Language Databases Interfaces, Proceedings of 8th International Congress on Computer Science Research. Instituto Tecnologico de Colima, Colima, Mexico, Nov. 2001, pp 63-70.