OVERVIEW OF THE NATURAL LANGUAGE INTERFACES

Victoria LAZU, Victoria BOBICEV

Technical University of Moldova

Abstract: *The paper presents an overview of the natural language interfaces, their types and techniques of elaboration. Natural Language Interfaces (NLI) are the easiest way of communication for the users. However, there is no computer system that could process the unrestricted natural language input without errors yet. The proposed solutions for this problem used restricted vocabulary and a limited set of phrase construction rules adapted to the specific task and domain.*

Keywords: *Natural language interface (NLI), natural language processing (NLP), Computational Linguistics (CL), controlled language.*

1. Introduction

Natural language Interfaces (NLI) are an active area of study in the field of Natural Language Processing (NLP) and Computational Linguistics (CL). Natural language interface gives people the possibility to interact with any electronic device using ordinary language they use daily. In the ideal case the devices understand what was said and respond appropriately. Today we are still far from such ideal technology. Currently, human-computer natural language interaction systems have not been implemented yet for national languages (including Romanian); their development requires innovative research related to the specific features of each language in question [1]. The development of human-computer interaction in natural language for Romanian is a part of the current trends to develop human-machine interaction systems in national languages other than English.

2. General structure and types of architecture

Natural Language Interface consists of two components [6]:

1) linguistic component;
2) computer system component.

Linguistic component is responsible for the understanding and interpreting the user interrogations expressed in natural language and translating it a formal language appropriate for internal system functionality. The second function of this part is the generation of relevant answers in natural language. Computer system component is heavily dependent on its functions; it can be a database, a robot or a virtual interlocutor. Figure 1 shows the general structure of this type of the interfaces. The query usually expressed as a natural language sentence is analyzed syntactically and semantically and then converted into formal representation suitable for the system in question.

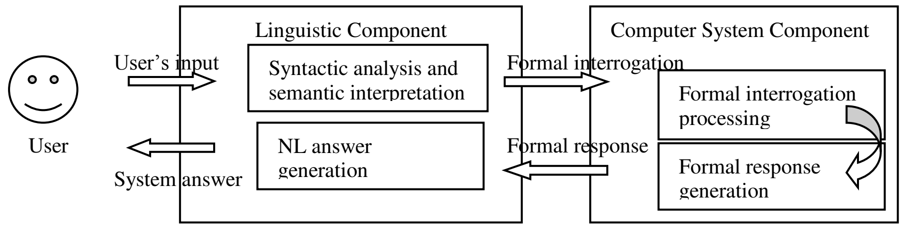


Figure 1. Main elements of the computer system with a natural language interface.

The best principle of design is to separate the interface components from the main system and to build relatively independent modules for each task. The most difficult part in such systems is the natural language input analysis. Natural language is highly ambiguous and currently there is no reliable grammar which is able to analyze any possible sentence even for English. Thus, various methods were created that allow computer systems to interpret the user's input even if it is too complicate for complete structural analysis. We list several architectures adopted for the elaboration of such systems [2]. They are presented in the increasing order of their complexity and respectively their performance.

- *Pattern Matching Systems.* The main advantage of the pattern-matching approach is its simplicity. In such systems no parsing and interpretation modules are needed, and they are easy to implement. In this kind of approach, a number of system adapted patterns are designed and these patterns are being searched in the user's input. Evidently, these systems are limited to specific task and their performance depends on number and complexity of the patterns. For example, if a database contains information about the countries, their capitals and language, the possible pattern would contain words "capital", "country" and "language". Thus, the system would search for these words and some names in a user question. An example of a question can be: "What is the capital of Canada?" which would match the pattern: " . . . capital . . . <country>". This would generate the action: "Report Capital of row where Country = Canada" in which the part "<country>" is replaced with the word from the question [13].

In the easiest cases a system can guess that "Canada" should be searched in the column "Country" but in more complex storages of information it is more complicate. An example given in [2] demonstrated that the question "Get the unit price of TOFU" cannot be answered by the system because it is not indicated what is TOFU and the correct question which could be processed by the system should be "Get the unit price of product whose name is TOFU". In this case the system matches name with the attribute *productname* and searches TOFU in the corresponding column.

In many cases these systems are designed to come up with some reasonable answer, even if the input is out of the range of the patterns they have. One of the earlier examples of natural language processing system which has been designed in this style is ELIZA[1]. ELIZA functions comparing user's input phrase with the internal patterns and typically rephrased the statements of the users as questions. There are a set of general phrases and questions which ELIZA use in case of zero matching between the user's sentence and the patterns.
in order to demonstrate some features of intelligence NLI need to be able to learn from the online ontologies. In [4] the system extracts the information about the term appeared in the question only implicitly. The system consuls the appropriate ontology and extract the necessary domain knowledge which help to interpret the user's query adequately. For example, the term '*watershed*' can be found in geographic ontology and interpreted correctly.

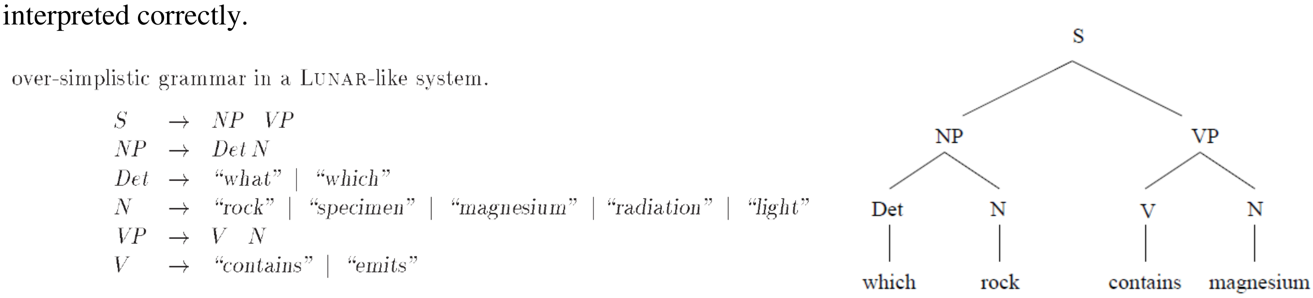


Figure 2. A simplified example of a small part of syntactic grammar [12].

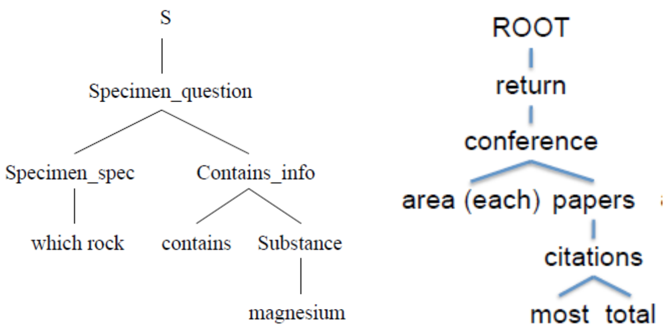
- *Syntax-Based Systems*. In these systems the user's question is parsed (i.e. analyzed syntactically) and the resulting parse tree is directly mapped to the computer system internal language. Syntax-based systems use a grammar that describes the possible syntactic structures of the user's questions. Syntax based NLIs are usually application-specific and provide a set of carefully designed structures which facilitate the mapping from the parse tree to the internal language. It is usually difficult to develop the mapping rules that can transform directly the parse tree into some expression in an internal language (e.g. SQL). The main advantage of using syntax based approaches is that they provide detailed information about the structure of a sentence. Having this information, the system can map the semantic meanings to certain production rules (or nodes in a parse tree). Figure 2 presents a simplified fragment of a syntactic grammar including lexicon rules which map the part of speech nodes to real words. Below the list of rules a parse tree of the sentence "Which rock contains magnesium?" is demonstrated.

- *Semantic Grammar Systems*. Such systems are similar to the syntax based system. They also produce a parse tree of a sentence and the result is obtained by mapping this tree to the internal representation. The basic idea of a semantic grammar system is to simplify the parse tree as much as possible, by removing unnecessary nodes or combining some nodes together. Based on this idea, the semantic grammar system can better reflect the semantic representation without having complex parse tree structures. The semantic grammar approach also provides a special way for assigning a name to a certain node in the tree, thus resulting in less ambiguity compared to the syntax based approach. Two parse trees created with semantic

[1] http://www.lpa.co.uk/pws_dem4.htm

grammar are presented in the figure 3. Unlike the syntactic grammar which contains syntactic categories such as ‘noun phrase’, ‘verb phrase’, etc. semantic grammar manipulates elements with semantic meaning. The first one (a) was designed for interrogation the database containing information about samples of rocks brought back from the moon and it contained such semantic categories as ‘specimen’, ‘substance’, etc. and this grammar impose the constrains which allowed a ‘specimen’ to ‘contain’ a ‘substance’ but did not allow a ‘substance’ to ‘contain’ a ‘specimen’. The second example (b) presents a parse tree for the interrogation “Return the conference in each area whose papers have the most total citations” to the database with conference papers. Its nodes also are the semantic categories used in the domain such as ‘conference’, ‘paper’ and so on.

- *Intermediate Representation Languages*. Most current NLIs first transform the natural language question into an intermediate logical form, expressed in some internal meaning representation language. The intermediate logical form expresses the meaning of the user’s question in terms of high level world concepts, which intend to be independent of the internal representation structure. The logical form is then translated to an expression in the internal representation language and processed by the computer system.



(a) (b)

Figure 3. Two simplified examples of semantic grammar parse tree from [12].

3. Types of interfaces

Currently, the researches in the domain of natural language interfaces development include several directions, namely:

- Interfaces to traditional relational databases.
- Intelligent interfaces for expert systems.
- Natural language interfaces to databases stored in XML.
- Intelligent interfaces for ontologies.
- Natural language interfaces in robotics.
- Natural language interfaces to production systems.
- Natural language interfaces in virtual reality.

Research work in querying data from relational databases often follows one of two paths: the structured query approach and the keyword-based approach. Both approaches have their advantages and disadvantages. The structured query approach, while expressive and powerful, is not easy for naive users. The keyword-based approach is very friendly to use, but cannot express complex query intent accurately. Natural language has both advantages to a large extent: even naive users are able to express complex query intent in natural language. A good review of the NLI to databases is given in [2].

Any expert system is usually designed for a narrow field and, respectively, it manipulates the terms of this area with a well-defined meaning. Query analysis for expert systems has similar steps as for DB, and requires a well-developed lexicon for the expert system for proper analysis of the terms. In most cases, the use of so-called narrowed language with a limited number of well-specified words is required to interact with expert systems in order to reduce ambiguity. The approach shown in [3] is designed to use a controlled language for the queries.

XML has become very popular as a form of database storage especially when it comes to a various online data. Converting a natural language query sentence into a formal database query is a major challenge. The NLI to XML (NaLIX) presented in [9] can accept an arbitrary English language sentence as a query input. This query is then translated into an XQuery expression.

With development of the Semantic Web ontologies become part of the Internet; their number is growing exponentially and most of them are accessible online. In order to demonstrate some features of intelligence NLI need to be able to learn from the online ontologies. In [4] the system extracts the information about the term appeared in the question only implicitly.

Higher intelligence and greater autonomy of more advanced robot systems increase the requirements for the design of a flexible interface to control the system. Simple instructions in natural language syntax are not sufficient for interacting with an autonomous mobile robot [10]. Flexible human-machine interaction is only possible if intelligent robots are made more responsive. A robot needs the capability to report task specific information, to explain its behavior, and to provide information about the environment. If the information about the environment and the world model is exchanged in a NLI dialogue system, the natural language commands and queries from the user are the input for the natural language access system.

The requirements for NLI to production systems are similar to the previous ones with the robots. In [6] an example of such a system is given in detail. The key point of the method presented is the use of a number of reusable application-independent components, in order to reduce the effort while adapting this system to the new environment.

Natural language interfaces in virtual reality requires a more complicated structure because in many cases the interaction is done in dialogue form. Dialogue involves linking sentences issued by the user; the system should preserve the history of dialogue and be able to connect current expression to the previous ones in order to understand the spoken input adequately. The system presented in [7] used off-the-shelf speech recognition and synthesis modules.

4. Conclusions

Natural language interfaces have been studied for several decades. Numerous attempts to create such interfaces for various applications had achieved some success. However all created systems could be used only in limited domain they were designed for. All of them worked with the restricted subset of natural language lexicon and syntactic structures. The ambiguity of free natural language is the unsolved problem for all kind of such systems. Nevertheless, computer systems are reached almost every domain of human activity and the development of more natural human-like interaction for various domains is a vital necessity.

References

[1] T. S. Topoleanu. *Sistem Inteligent de Interactiune Vocală Om-Robot Mobil Industrial*. Teza de doctor, universitatea “Transilvania”, Braşov, 2012.

[2] N. Nihalani, S. Silakari, M. Motwani. *Natural Language Interface for Database: A Brief Review*, IJCSI, Vol. 8, Issue 2, (2011).

[3] S. Mador-Haim. *Natural Language Interface for Geographical Information Systems*. Master thesis, Technion - Israel Institute of Technology, Haifa, (2006).

[4] Y. Li, I. Chaudhuri, H. Yang, S. Singh and H. V. Jagadish. *Enabling Domain-Awareness for a Generic Natural Language Interface*. AAAI, Canada, (2007).

[5] E. Kaufmann, A. Bernstein, R. Zumstein. *Querix: A Natural Language Interface to Query Ontologies Based on Clarification Dialogs*. In: 5th ISWC , (2006), pp. 980-981.

[6] S. Chong, R. Pucella. *A Framework for Creating Natural Language User Interfaces for Action-Based Applications*. Proceedings of the Third International AMAST Workshop on Algebraic Methods in Language Processing, TWLT Report 21 (AMiLP-3), (2003).

[7] S. Everett, K. Wauchope, M. A. Pérez-Quiñones. *Creating natural language interfaces to VR systems*. Virtual Reality 4(2) (1999), pp. 103-113

[8] T. M. Wasfy, A. K. Noor. *Rule-based natural-language interface for virtual environments*. Advances in Engineering Software, 33(3) (2002) DOI: 10.1016/S0965-9978(02)00004-2.

[9] Y. Li, H. Yang, H. V. Jagadish. *Constructing a Generic Natural Language Interface for an XML Database*, Advances in Database Technology - EDBT 2006, Lecture Notes in Computer Science, vol. 3896, (2006), pp 737-754.

[10] T. Laengle, T. C. Lueth, E. Stopp, G. Herzog, U. Rembold. *KANTRA: Human-Machine Interaction for Intelligent Robots Using Natural Language*, Proceedings of the 3rd International Workshop on Robot and Human Communication, (1994).

[11] H. P. Barendregt. *The Lambda Calculus, Its Syntax and Semantics.* Studies in Logic. North-Holland, Amsterdam, (1981).

[12] F. Li and H. V. Jagadish. *Constructing an Interactive Natural Language Interface for Relational Databases*. PVLDB, (2014).

[13] J. Miller, C. Krapichler, L. S. Nguyen, K.-H. Englmeier, M. Lang. *Speech Interaction In Virtual Reality Symposia Conference,* The Mobile Genome: Genetic and Physiological Impacts of Transportable Elements, Germany, (2015).