Menu Choice Translation

-a flexible menu-based controlled natural language system -

Cristina Vertan
University of Hamburg
cri@nats.informatik.uni-hamburg.de

Walther v. Hahn
University of Hamburg
vhahn@nats.informatik.uni-hamburg.de

Abstract

This paper describes an experiment in controlled natural language interface technology. The system under work ensures a syntactic and semantic correct composition of input sentences, without restricting the user too much. The lexicon as well as the syntactic and semantic restrictions are maintained in flexible structures which can be updated at any time. The application of the system at present is machine translation. The paper presents the general paradigm of menu based natural language interfaces, the architecture of the proposed system (MenuChoice), the status of implementation, as well as further work.

1 Menu-based Natural Language Interfaces

Natural language is still the most accessible modality of human computer interaction. even if it is still hard to implement it. Especially, those features which make natural language extremely flexible (spontaneous utterances, rich syntactic choice, or semantic domain independence) are the most difficult features for natural language systems. Until now, such systems require careful use, because they are strongly dependent of the correctness of their input, which, however, often contains typing errors, grammatical re-formulations, or slightly ungrammatical every-day idioms. Such systems with their deep analysis of all possible meanings and pragmatic senses may even ask back to resolve ambiguities, invisible to the user. Additionally, these system require huge resources, and thus are too expensive and too sophisticated for many applications. A

solution in between is a system which offers actively controlled language and performs in a restricted domain only. Menu-based natural language interfaces are examples of such systems . These interfaces try to overcome the restrictions imposed by the early menu-based systems [Walker89], are robust by virtue of predefined linguistic subsets, are less expensive, and avoid the resource problems stemming from the totally free use of full natural language.

Menu-based natural language interfaces propose a broad variety of natural expressions: the user has the choice among a large number of alternatives at any place of the input, but the input language is still controlled in the sense that any choice has a well-defined (syntactic) structure. Additionally, choosing from a menu eliminates typos. As the quality of a natural language processing system depends strongly already on the formal quality of the input, this kind of systems was regarded as the ideal cost/effect compromise for human-computer.

The first fully implemented interface, NLMenu, was developed in the eighties (Thompson 1984). The interface was designed according to applications (i.e., access languages of the target systems). It was based on a context-free grammar and a corresponding lexicon. It offered a number of menus from which the user could select words. The classification of the menus was application dependent. According to the specified grammar only those menus were enabled, from which a correct sentence could be composed. It its time, NLMenu was a considerable step forward. However, the language was heavily restricted. Moreover, using a stem-lexicon without morphological processes the system produced rather strange sentences, which did not look like natural communication.

ROSY(RObust SYntactic analysis) (Blum 1987) was developed at the University of Saarbrücken and was meant as an extension of NLMenu, taking into account the particularities of German language. With the addition of a morphological processor, the natural impression of the generated sentences was strongly improved.

Several other interfaces, based on the same principles were developed afterwards, mainly for database query tasks (Androutsopoulos and Ritchie 2000).

Among the disadvantageous features of this generation of interfaces, we mention:

- a heavily controlled language,
- an a priori classification of words in the menu slots according to applications (e.g., in ROSY menus for "Commands", "Objects", "Attributes" etc., in other systems a fixed order of parts of speech),
- no spontaneous insertions,
- no support for lexicon-updates,
- no semantic constraints for slots,
- a reduced size of the lexicon due to artificial space limitations,
- no intelligent ordering of items in the menus,
- no connection with other input modalities (for example images)

MenuGen (MENÜbasierte GENerierung) (Hammerich 1999) was implemented at the University of Hamburg as a prototype. It was developed as a menu-based natural language interface with the following features:

- the user can enter new words,
- the system is platform-independent (it is entirely written in Java),
- it supports multilingualism,
- graphical extension: words can be also selected by clicking on a technical drawing.

The output quality of MenuGen was better as NLMenu, because a full-form lexicon was used. This can be an inconvenient for real applications, which require bigger lexicons. A second limitation is the checking of the syntactic correctness only after the end of the complete choice. In case of an input, which has no internal interpretation, the user is asked to produce a new input, but he was not guided (e.g., by a menu-disabling process) how to produce a correct input sentence.

2 MenuChoice Translation System

In the previous section we described the principles of existent menu based natural language interfaces. However, most of these interfaces are used only as monolingual database query systems and had very restricted capabilities. In the following paragraphs we describe the architecture of a menu-based interface system which can be used also for machine translation. The language is still controlled by several restrictions (lexicon, sentence types, verb tense and roles, etc.) but the user has much more freedom in composing sentences. Additionally two interacting modules ensure syntactic *and* semantic correctness. A morphological generator is also linked to the interface. There is no default classification of words in menus.

2.1 The scenario

The scenario of the current version is the following: a tourist (English speaking) travels to a country where English is not widely known. During his/her travel he needs basic medical assistance, and therefore he explains the symptoms and other details of the illness to a local doctor (probably by using a handheld computer, PDA). This scenario has the following characteristics:

- the input sentences usually are not very complicated, but
- it is highly important that they are correctly formulated and, even more important, correctly translated.

From the description of the scenario it becomes clear, that the quality of the translation in such an application must not be endangered by input errors, dialects, typing errors, elliptical sentences, etc.

The system, which is described here, aims at the elimination of these sources of mistranslations by offering alternatives for the next word to be entered. The users do not type any longer their word but select from a list of possible continuations. The system is currently being implemented for the language pair English – Romanian but can be adapted to other languages.

2.2 Architecture of the System

The architecture of the system is presented in Figure 1. In the followings we will explain the data structures which are used, the processing modules as well as the flow of information through the system.

All possible sentences to be generated (let us call this set S) are maintained in a word graph.

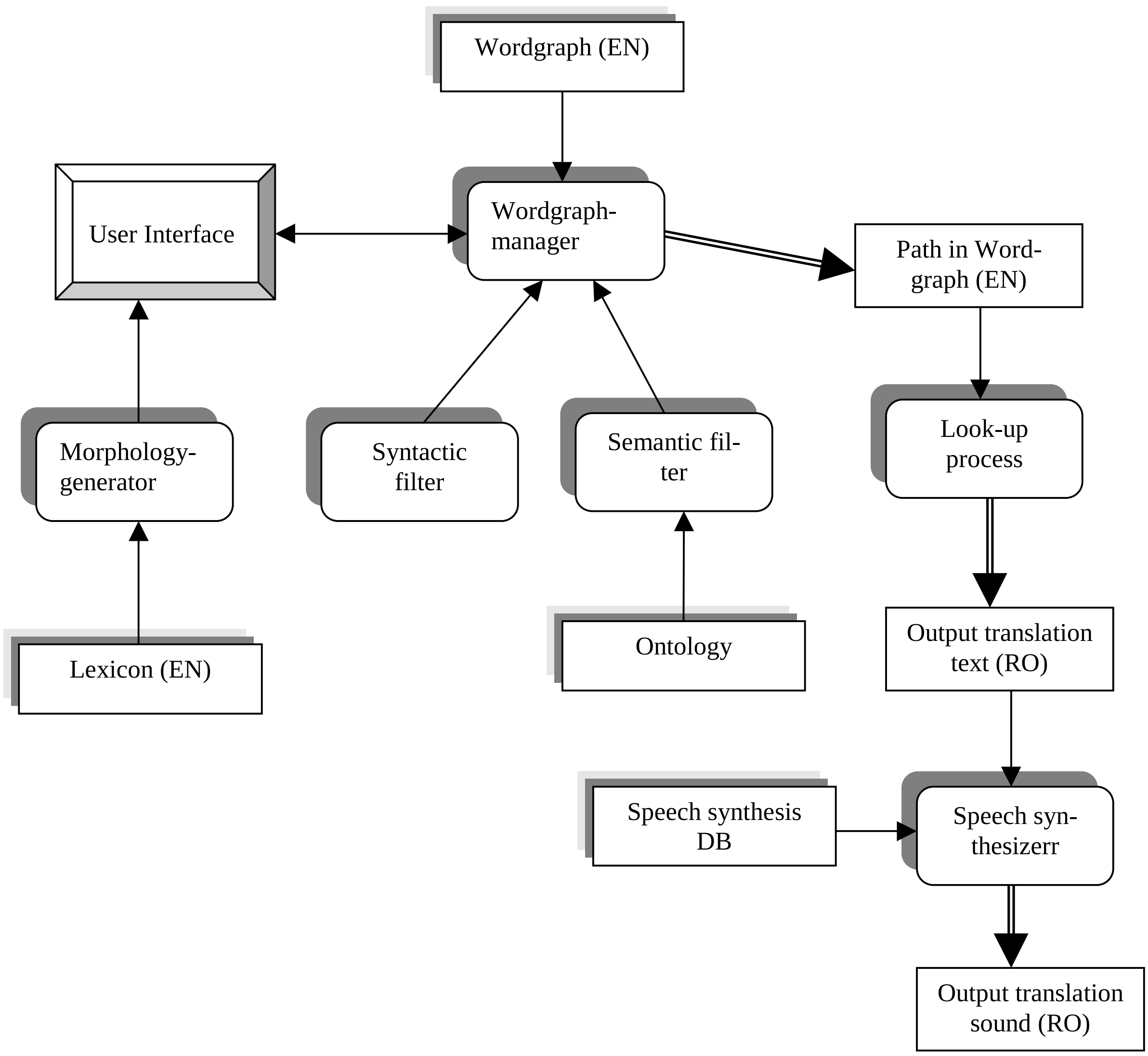

Figure 1. Architecture of the MCT System

A word graph is a directed graph G =(W, E) with the following properties:
W is the set of nodes representing all possible words to be used,
E is the set of edges joining two nodes. Between two nodes w_1 and w_2 there is one edge if and only if there is one sentence in S where w_2 is following w_1.

For example for the sentences S_1:

- `I have a very bad headache.`
- `I had pain in stomach`
- `Since three days I have fever.`
- `The pain is stronger after I am eating.`

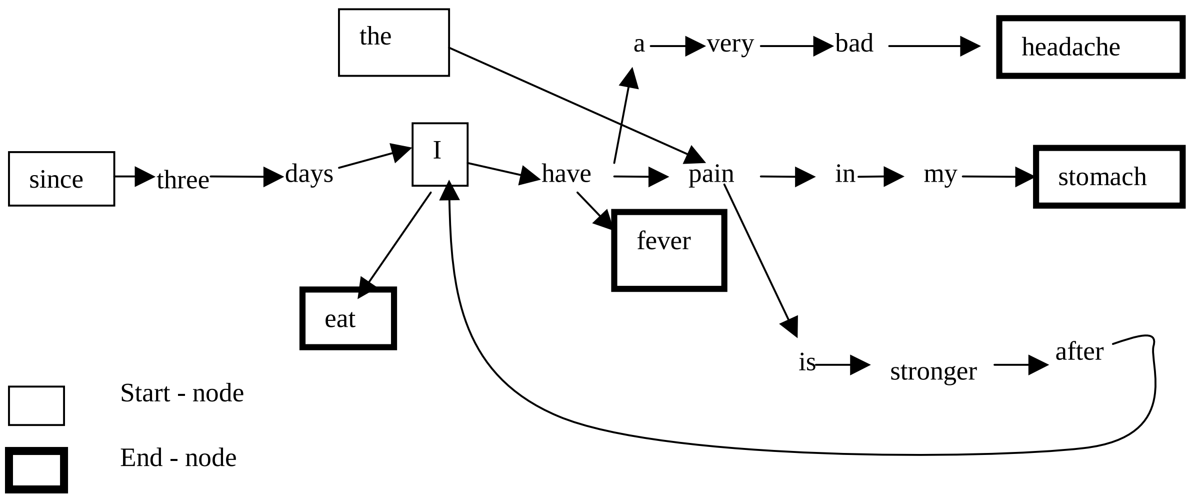

Figure 2. Word graph associated to the set of sentences S_1

From this example we can observe that :

- a sentence in S_1 represents a path in the word-graph but
- not every possible path in the word-graph is a (syntactically and/or semantically) correct sentence.
- For example a sentence like
  - `“Since three days I have pain in stomach”` is a correct sentence although not in S_1
  - `“Since three days I have pain is stronger after I have fever”` is not syntactically correct
  - `“The pain is stronger after I have a very bad headache”` is syntactically correct but semantically incorrect.
  - `“Since three days I eat”` is syntactically and semantically correct but most probably not pragmatically correct[1]

In order to ensure the syntactic and semantic correctness two other structures are used:

1. a transition network grammar (Allen 1995) which provides the rules to produce syntactically correct sentences
2. an ontology (Sowa 1999) for the semantic correctness.

As a consequence a correct sentence is a path in the word-graph:

- to which a path in the transition network corresponds, and
- whose nodes satisfy the constraints imposed by the ontology.

The interaction between the word-graph, the transition network and the ontology is ensured by the “word-graph manager”.

[1] The correct sentence is actually "Since three days I have been eating”. In our system, tenses different from past and present are only indicated by checking some controls on the user interface.

By these means the user can construct not only sentences which exist in the original set S but also variations which fulfill the constraints given below. The user interface tries to avoid the disadvantages of previous menu based input interfaces: The user can choose the next input word from a list, which is dynamically generated by the word-graph manager. As for a real system with a very large number of words, this list can be very large, therefore the word are shown in a dynamic way:

- either those words are shown dynamically, which begin by the sequence typed in so far, or
- according to the previous use, the most frequently used continuations are given.

This option can be changed at the beginning or during the use of the system.

The words in the menu are initially given in their quotation form. The morphology generator is responsible for the morphological correctness of the sentence afterwards. However, for the moment we restrict verbs to present tense. The interface, however, gives an opportunity to specify another tense by non-linguistic means..

Once the sentence is composed, the next step of the system is the translation into a target language. As already explained, our system aims at avoiding the implementation of a rule-based machine translation tool with all corresponding processes (complete analysis of the source sentence, transfer rules or interlingua representation, and generation in the target language). The solution proposed here is a database of translation patterns in the following manner:

- at the beginning these patterns are represented by the translations of the words and the sentence structures in the set S,
- two such patterns can be combined and generate a new pattern,
- the database is updated dynamically with new patterns.

A necessary feature for "natural" interaction is the option for adding new words. An acquisition tool, independent from the current language, allows for entering lexicon enhancements. However, in order to ensure semantic correctness, only new words which belong to an existing class are allowed. A later extension should include an intuitive analogy-based lexical acquisition tool.

3 Conclusions and further work

The architecture that we presented is presently under implementation. This phase will be followed by an evaluation done in two directions:

- a theoretical study of the dependency between the growth of the lexicon, the data structures associated and the speed of the system
- a practical evaluation of the correctness of the generated input sentences as well as the corresponding translations.

Speaking of user evaluations, the system inherits the well known general MT problem, that users cannot estimate the correctness of a translation to an unknown language.

We are also planning to implement (or include from external sites) two other tools:

- a tool for developing and extending the ontology
- a tool for selecting word by clicking on different parts of a technical drawing or image. This is a realistic help for non-expert speakers in the domain of distant maintenance.

In our view such systems have two main advantages for restricted domains:

- they eliminate input errors
- they can replace expensive machine translation tools.
- the engineering of such systems is much easier for non-linguists, compared to complex grammatical and semantic representations of full MT systems.

References

Jeff Allen 1995., *Natural Language Understanding*, Benjamin Cummings Publishing Company.

Ioan Androutsopoulos and George Ritchie 2000, *Database Interfaces*, in: Dale R., Moisl H. and Somers H (Eds.): Handbook of Natural Language Processing, Marcel Dekker

Edward J. Blum, E. 1987, *ROSY - Menü-basiertes Parsing natürlicher Sprache unter besonderer Berücksichtigung des Deutschen*, KI-Labor, Memo Nr. 20, FB. Informatik, University of Saarland, Saarbrücken

Walther v. Hahn, W. 1994, *Computermodelle der Sprache und des Sprechens*, in V. Braitenberg, I. Hosp (Eds.): Simulation - Computer zwischen Experiment und Theorie, Texte vom "Bozner Treffen", Reinbek bei Hamburg, S. 80-99

Stefan Hammerich, 1999, *Menübasierte Generierung natürlicher Sprache*, Studienarbeit, University of Hamburg, 1999.

James D. Moore and V. Mittal 1995, *Dynamic Generation of Follow up Question Menus: Facilitating Interactive Natural Language Dialogues*, in Proceedings of CHI'95, pages 90-97.

John F. Sowa 1995, *Knowledge Representation – Logical, Philosophical and Computational Foundations*, PWS Publishing Company.

C. W. Thompson 1989, *Using Menu-based Natural Language Understanding to Avoid Problems Associated with Traditional Interfaces to Databases*, Ph. D. Dissertation, Dept. of Computer Science, University of Texas, Austin,

M. Walker and S. *Whittaker , When Natural Language is Better than Menus: A Field Study*, Technical Report HPL-BRC-TR-89-020, HP Laboratories-Bristol.