A Natural Language Interface to a Theater Information Database

Margus Treumuth

Institute of Computer Science
University of Tartu, Tartu, Estonia
treumuth@ut.ee

Abstract

The development of a natural language dialogue system as an interface to a theater information database is a joint research project of University of Tartu (Estonia) and Tallinn University of Technology (Estonia). The underlying database contains information about theater performances in a certain theater or city. The dialogue system can be used to ask information about performances using either spoken or typewritten natural language in Estonian. The dialogue management module was developed by the author at the University of Tartu. The speech recognition and speech generation modules were added by Tallinn University of Technology. This article discusses the development of the dialogue module but not the speech recognition/generation modules.

Keywords: dialogue system, natural language interface, question answering system, closed-domain.

1. Introduction

As computers continue to become more affordable and accessible, the importance of user interfaces that are effective, robust, unobtrusive, and user-friendly – regardless of user expertise or impediments – becomes more pronounced. The goal of natural language interfaces is to bridge the gap between the linguistic performance of the user and the linguistic "competence" of the underlying computer system [1]. The performance of speech recognition components has significantly improved: only within ten years we have passed from systems able to recognize isolated words to systems able to recognize continuous speech with an unlimited vocabulary uttered by any speaker; or to systems able to carry a spontaneous dialogue over a telephone, on a well-defined topic (e.g. information on train or airplane schedules). The expanding computer and telephone networks allow the general public an immediate location-independent access to large databases [2].

The paper is organized as follows. In section 2 we briefly draw parallels with early question answering systems and our current development. Section 3 describes the system components. Section 4 is concerned with experiments. Section 5 describes the software environment of the implementation. Finally we will make our conclusions in section 6.

2. History and motivation

One of the famous early question answering systems was ELIZA [3] that detected important words in the person's input. The same approach is also used as a secondary parsing technique in our dialogue system. The primary parser in our system uses basically the same technique with a slight difference by using a domain specific relational database (theater information).

The dialogue system developed in this project operates in a constrained linguistic domain – theater information. The underlying database contains information about theater performances in a certain theater or city. The dialogue system can be used to ask information about performances. As for now, the system does not contain price or booking information, also the names of actors and authors are not included at this time.

The system can deal with either typewritten or spoken language. The typewritten interface is accessible at http://www.dialoogid.ee/. The language used by the dialogue system is Estonian.

The research groups involved are the Institute of Cybernetics: Laboratory of Phonetics and Speech Technology of Tallinn University of Technology and the Institute of Computer Science: Research Group of Computational Linguistics at Tartu University.

As Estonian speech recognition and speech generation modules are available, our goal is to build a system that could be used over a telephone.

3. System components

The dialogue system consists of modules for speech recognition, dialogue management, morphological analysis, query generation and speech generation. This article discusses the dialogue management module and query generation module. We do not discuss the speech recognition and speech generation modules as these are standalone modules developed by collaborating researchers in Tallinn University of Technology [4].

3.1. Dialogue Management Module

The dialogue management module allows mixed-initiative interaction (computer or user controlled) and employs a simplistic keyword matching strategy that integrates Estonian morphological analyzer [5]. The Estonian language is an agglutinative language that is rich in morphology. Therefore, the parsing technique involves automatic generation of lemmas or base forms using morphological analyzer. This way the system can handle minor deviations of the input. As a weakness – the morphological base form generation could also trigger ambiguity leading to unexpected results.

Some deviations still cause problems and in the near future we will use the spell checking functions of the morphological analyzer to handle typing errors. We also plan to use the Levenshtein algorithm [6] to calculate the distance between strings. This way we can guess which word (from a

dictionary or database) is meant when an unknown word is encountered.

The system also uses examples of linguistic phenomena from a dialogue corpus [7], yet it is not a stochastic approach as probabilistic techniques are not used at the moment. We examined the corpus to see how the users phrase their questions and how they express dates and time.

3.2. Knowledge Base

The knowledge base of the dialogue management module consists of a primary database and a secondary database. The primary database contains the theater information (city, theater, performance, date) and the secondary database contains some simple linguistic facts (domain specific words by attribute-value pairs) e.g.

Keyword (the knowledge base is in Estonian): *ticket*
Values for keyword *ticket* (the number of values per keyword is not limited):

- Sorry, we have no ticket prices.
- Sorry, we have no information about the availability of tickets.
- We only know the dates of performances.

The primary database – the database of theaters and performances – is gathered manually from several online databases and is also updatable over a web interface. We also plan to arrange the database to run automatic updates daily.

The secondary database is also directly visible and adjustable in a knowledge base settings file. Various forms of greeting expressions and some domain specific phrases can be modified by a system administrator.

3.3. Query Generation Module

The query generation component converts user input to SQL (structured query language) queries [8] - commands to be passed to the underlying database.

3.3.1. Primary Parser

The primary parser detects proper names that occur in the primary relational database (names of performances and theaters), date and time phrases. Once a certain keyword has been recognized, the system may retrieve the answer from the database having the power of SQL available for quick definition and manipulation of data.

Let us consider a dialogue taking place between a human and a computer.

- <Human>: I would like to see the musical ***Cats*** on ***March 17***.
- <Computer>: The Cats is not played on March 17, yet it is played on March 19.

In this example, the highlighted words (Cats and March 17) are the actual keywords which convey the most important information and are recognized by the system. All the other words are semantically irrelevant and can be ignored.

The previous example might seem simple, yet the recognition of dates is not a simple task. There are many ways users can express dates and time. Therefore, we have created a separate module in our system for the date recognition.

The reaction to a user utterance depends on the state of the dialogue (*dialogue context*). That is, the choice of answer is based on previously acquired knowledge. Users can continue to ask queries about the previous topic. The system can remember facts the user has asked before. For example, if the user has mentioned a theater by name, all further references to some certain dates are handled in the context of the theater mentioned previously, e.g.

- <Human>: What is going on in the Theater Royal?
- <Computer>: The Producers is played today.
- <Human>: What about tomorrow?
- <Computer>: There are no events in the Theater Royal tomorrow.

3.3.2. Secondary Parser

The secondary parser is used if the primary parser gave no results. It can recognize only predefined words and/or phrases described in the secondary database and can only respond using a number of predefined sentence patterns also described in the secondary database.

Randomization is used in choosing the answer from the secondary database to provide the effect of non-linear transformations between inputs and outputs. Users tend to like slight unexpectedness and surprises (e.g. various expressions of greetings). The users will get bored if the system is too predictable and they find out the limits of the system too quickly.

It is essential to keep the users trying to get their answers. This will provide the developers with valuable chat logs as the system stores all conversations. These chat logs are later used as training data to manually improve the performance of the linguistic model that relies on collection of predefined keywords and expressions to represent semantic notions.

4. Results of Experiments

The system was tested with 150 conversations, some typewritten and some spoken. There were differences in typewritten and spoken conversations, yet the distinction between those is not important at this stage of development.

The subjects failed to communicate successfully with the system 5% of the time. The system failed 25% percent of attempts to successfully answer a question. This shows that the system can make mistakes yet the users are still able to get their answers by rephrasing their questions.

The main problems are:

- There are some mistakes in pattern matching where minor deviations of the input take place. Mainly while matching the names of performances or names of theatres. As stated above, in the near future we will apply the Levenshtein algorithm to calculate the distance between strings. This way we can guess which word (from a dictionary or database) is meant when an unknown word is encountered. We also plan to use the

spell checking functions of the morphological analyzer to handle typing errors.

- The users quickly discover the limits of the system knowledge – there is no information about ticket prices, no booking. So the user is unable to get the information from the system. We plan to expand the knowledge base and include the ticket prices and booking information.

5. Implementation effort

The dialogue module is a web enabled system. Therefore, it is easy for the developer to add, modify and deploy new functionality. The web enabled system also provides an easy way to collect chat logs that can be used as training data to manually improve the performance of the linguistic model.

The dialogue module was developed in PHP using MySQL as the database server and Apache HTTP Server as the web server.

6. Conclusions

The development of a natural language dialogue system as an interface to a theater information database is a joint research project of University of Tartu (Estonia) and Tallinn University of Technology (Estonia). The underlying database contains information about theater performances in a certain theater or city. The dialogue system can be used to ask information about performances using either spoken or typewritten natural language in Estonian. The dialogue management module was developed by the author at the University of Tartu. The speech recognition and speech generation modules were added by Tallinn University of Technology.

The dialogue system developed in this project demonstrates that we are capable of designing systems which can understand a small subset of natural language in a constrained linguistic domain. The conducted experiments show that our methods are satisfactory yet need some further improvements.

In the future, we hope to release a dialog based speech-understanding system that could be used over a telephone.

7. Acknowledgements

The project has support from the Estonian Ministry of Culture, Estonian Information Technology Foundation, Estonian Science Foundation (grant no 5685) and Elion Enterprises Ltd.

8. References

[1] B. Manaris, "Natural Language Processing: A Human-Computer Interaction Perspective," In Advances in Computers (Marvin V. Zelkowitz, ed.), vol. 47, pp. 1-66, Academic Press, New York, 1998.

[2] Minker, W., Bennacef S. Speech and Human-Machine Dialog. Boston/Dordrecht/London: Kluwer Academic Publishers, 2004.

[3] Weizenbaum, J. ELIZA – A Computer Program for the Study of Natural Language Communication Between Man and Machine. Commun. ACM 9(7), 36-43, 1966.

[4] Tanel Alumäe: Large Vocabulary Continuous Speech Recognition for Estonian Using Morphemes and Classes. TSD 2004: 245-252. 2004.

[5] Kaalep, Heiki-Jaan. An Estonian Morphological Analyser and the Impact of a Corpus on Its Development. Computers and the Humanities 31: 115-133, 1997.

[6] Vladimir I. Levenshtein, Binary codes capable of correcting deletions, insertions, and reversals, Doklady Akademii Nauk SSSR, 163(4):845-848, 1965 (Russian). English translation in Soviet Physics Doklady, 10(8):707-710, 1966.

[7] Gerassimenko, Olga; Hennoste, Tiit; Koit, Mare; Rääbis, Andriela; Strandson, Krista; Valdisoo, Maret; Vutt, Evely. Annotated dialogue corpus as a language resource: an experience of building the Estonian dialogue corpus. In: The first Baltic conference “Human language technologies. The Baltic perspective”. Commission of the official language at the chancellery of the president of Latvia. Riga. 150–155. 2004

[8] Donald D. Chamberlin and Raymond F. Boyce. "SEQUEL: A structured English query language", International Conference on Management of Data, Proceedings of the 1974 ACM SIGFIDET (now SIGMOD) workshop on Data description, access and control, Ann Arbor, Michigan, pp. 249-264, 1974.