

Bilingual Dialogs with a Network Operating System

Emad Al-Shawakfa,
Computer Science Department,
Illinois Institute of Technology,
10 W. 31st Street, room 236,
Chicago, IL 60616
emad@mail.iit.edu

Martha Evens
Computer Science Department,
Illinois Institute of Technology,
10 W. 31st Street, room 236,
Chicago, IL 60616
mwe@math.nwu.edu

Abstract

We are building a natural language interface to the Novell NetWare Network Operating System that can carry on dialogs in Arabic or English or, sometimes, a mixture of the two. The parser is based on extensive case frame tables. We have developed a set of thirty target commands that we feel are appropriate for novice users. Our response generation capabilities are minimal at this moment. We are collecting and studying user dialogs to provide better response generation capabilities. We are also investigating the different ways that novice users express commands to the Novell Network Operating System to cover as many cases as possible.

Introduction

Learning to use an operating system is a major barrier in starting to use computers. There is no single standard for operating systems, so users who are familiar with one operating system become novices again when faced with a new one. The proliferation of client-server networks means that many new users must deal with a network operating system from the beginning.

In his experience as the laboratory manager at Computing and Network Services (CNS) at IIT, the first author has come across many situations where a student got frustrated just by dealing with the Novell Network Operating System that we have installed in our laboratories. To make life easier for novice users, we have started building a natural language interface.

We are building a system that can enable a Network Operating System to carry on a bilingual dialog in both Arabic and English. Users of this environment will be able to communicate with the operating system in both languages.

Since natural languages provide the basis for our everyday communications, most people find it easier to communicate with a computer system in natural language than using technical commands. In fact, there is a growing need for an interface that will allow users to express their problems in their own terms to the computer (Buchheit 1995a). For this reason, it is important to learn more about dialog structures and the way man-machine dialogs work.

Making a computer operating system take part in a dialog is a big problem as it requires a huge effort to analyze different plans the user wants the operating system to carry out. Also it requires a natural language understanding component and a natural language generator component. Most of the systems we have found so far function in English like UC (Wilensky et al. 1984, 1988), INFANT (Buchheit 1991, 1995a,b, 1996), NALIGE (Manaris, Pritchard, and Dominick 1994), and NLDOS (Al Daimi and Sinan 1993), or in Chinese like THJ (Li and Xing 1992).

There are two novel aspects to this research. As far as we know, no one has tried to build an interface to a network operating system. Also, all other dialog systems that we are aware of are monolingual and there have been no attempts so far to establish a bilingual dialog with an operating system.

Although a few systems have been built that can establish a natural language dialog with some operating systems, there has been no effort to establish such dialogs in Arabic with any operating systems. In fact, there has been little involvement of Arabic in the field of natural language processing, perhaps because the language is so different from European languages. Since computer science is often taught in English in the Arab countries, many Arabic speaking users need a bilingual environment.

In comparing Direct Manipulation and Natural Languages as interfacing technologies, Cohen (1992) has listed the strengths and weaknesses in Table 1 for the use of natural languages to create interfaces.

To build the system, we collected a number of user queries. Different actions that can be executed by novice users were determined. Associated commands were explored. For each operating system command, different ways of expression were explored.

This paper is organized as follows: a section on the bilingual issues that talks about different problems associated with a bilingual environment; a user dialog section, a section on case frames, a section on the architecture of the system in progress, and our conclusion.

Strengths of natural languages	Weaknesses of natural languages
1. Intuitive	1. Coverage is opaque.
2. Descriptive, including: <ul style="list-style-type: none"> a. Quantification. b. Negation. c. Temporal Information. 	2. "Overkill" for short or frequent queries.
3. Context.	3. Difficulty of establishing and navigating context.
4. Anaphora (e.g. pronouns)	4. Anaphora are problematic.
5. Delayed action possible.	5. Error prone.
	6. Ambiguous.

Table 1: Strengths/Weaknesses of Natural Language Interfaces (Cohen, 1992, p. 145).

Bilingual Issues

Arabic is oriented from right to left while English is oriented from left to right. If we have a dialog that has both, then in this case, we are faced with the difficulty of understanding each utterance. We need to determine how to deal with such input.

To solve this problem, based on the orientation of the first word, we are going to consider the language to which the first word belongs as the host language. At the time any words written in the second language will be inserted in the input of the host language. What we mean here is that, assuming the first word was written in Arabic, then if we are going to have an English word, then this word will be inserted in the current line from right to left. In other words, the first letter will be typed, the second will push the first to the left, and the third letter will push both letters to the left and so on until the word is completely typed. Although it is complicated, mixed orientation is not a strange thing to the speakers of the Arabic language. Arabic is oriented from right to left while Arabic numerals are written and read from left to right.

User Dialogs

During the work done in this research so far, we have collected two sets of samples of dialog from two different sources. The first set was obtained at IIT where we have asked the laboratory staff to write down the different dialogs they have with the users of the Computing and Network Services Laboratories in Stuart Building as well as the Residence Halls. The laboratory staff came back to us with a set of written questions pertaining to the interactions of the users with the Novell NOS we have on our machines.

The second set of language samples was obtained while the first author was teaching a Novell course at Robert Morris College to some novice users. In this course, he gave his students an extra credit homework assignment to

write down a dialog with the Novell operating system. The homework assignment is shown in Figure 1.

Furthermore, we have obtained a list of Novell commands that a user might be able to execute in a Novell Network Operating System environment. Some of these commands are given in Table 2 with different ways of saying them. So far, the only language samples that we have collected are in English. Arabic samples and bilingual samples need to be collected and analyzed. A scenario of a possible dialog between the users and the system is given in Figure 2.

Case Frames

Linguists have known the concept of *case* for at least 2000 years, but, modern work with case frames originated in the work of Fillmore (1968). He has introduced six cases and since then many other cases have been introduced into research. Cases are often referred to as Thematic Roles or Semantic Roles or Participant Roles (Wendlandt and Driscoll 1991). Allen (1995) has refined the cases and added few more to them. Some of the cases defined by Allen are: **AGENT, CO-AGENT, THEME, CO-THEME, INSTRUMENT, EXPERIENCER, BENEFICIARY, AT-POSS, TO-POSS, FROM-POSS, AT-LOC, TO-LOC, FROM-LOC, AT-VALUE, TO-VALUE, FROM-VALUE, AT-TIME, TO-TIME, FROM-TIME, and PATH**

The concept of English Case Frames was used at IIT by Dardaine (1995) and Pin-Ngern (1990) in the IITLEX lexicon for the CIRCSIM-Tutor Tutoring System being implemented at IIT. We have an Arabic lexical database designed by Al-Khrisat (1992) and Alsamara (1996).

We are using a case frame approach to parsing in our system for both Arabic and English, furthermore, we are using the same set of roles for both. An example of an English Case Frame is given in Figure 3.

CIS 230 LOCAL AREA NETWORKS I.

HOMEWORK FOR AN EXTRA CREDIT

Assuming that the computer in front of you is another person, using your conversational English, what do you think should be the command issued to the Novell Network Operating System to achieve the following tasks:

1. Print a file called **myfile.txt** on the first printer of the Network.
2. Connect you to a different server on the network called Server2.
3. Change your password.
4. Copy a set of files from your home directory to your floppy.
5. Move some files from the directory **Apps** to the directory **OFFICE**.
6. Protect your files against accidental deletion.
7. Restore your deleted files.
8. Make your directory accessible to other users.
9. Select a printer queue on the Network.
10. Send a message to a friend on the Network.
11. Check who is logged on to the Network.
12. Copy a directory from your account to another account on Server2.
13. To see what your directory has.

Figure 1: Extra Credit Homework Assignment to Elicit Sample Input.

Case Frames usually hold information about syntax, semantics and pragmatics related to any verb that will help us in carrying on the dialog better. For instance, by looking at the first line in Figure 3, we notice that the verb **Print** requires a subject that is an Agent that must be the System. The second line indicates a Theme as its Direct object that must be a file, or a list of files. So, one cannot ask to print the whole directory or the whole disk.

The System.

To enable bilingual interaction with the Novell Network Operating System, we are implementing a system that will take the user's input (either in Arabic or English), analyze it, extract the semantics from it, and then based on that representation execute the request. The system has the following modules:

- 1) The Tokenizer.
- 2) Utterance Type Determiner module.
- 3) The Parser.
 - Arabic Grammar.
 - English Grammar.
 - Arabic Lexicon.
 - English Lexicon.
 - Arabic Case Frames.
 - English Case Frames.
- 4) The Input Understander.
- 5) The Command Translator, and

- 6) The Response Generator and Dialog Planner.

The layout of this system is given in Figure 4.

Two types of knowledge will be kept in the system: application-nonspecific (general) knowledge and application-specific knowledge. The general knowledge will be mainly used to figure out the interaction with the system and mainly has to do with the natural language being used during the dialog process, and more specifically, how sentences can be formed, and how verbs can be used, etc.

The application-specific knowledge on the other hand, will be that knowledge that has to do with our application, in our case the Novell Network Operating System Commands. The knowledge will be kept in a database of different commands with all the possible arguments and options a command may take.

Conclusion

Carrying on dialogs with a computer operating system has always been a problem for novice users. This inspired us to try to establish a natural language interface.

Most of the dialog systems we have seen so far function in English. Furthermore, they are monolingual. Since not everyone on the face of this planet is fluent with English; especially in the Arab world, this has encouraged us to start working on a bilingual dialog that will enable users of

User: Print my file to the first printer.
System: What is the name of your file?
User: test.dat
System: I cannot find test.dat. Do you know where it is?
User: On my floppy.

Figure 2: A Scenario of a Dialog with a User.

Word	Syn-Role	Case	Occurrence	Sel-Rest
Print	Subject	Agent	Obligatory	System
	Obj-Direct	Theme	Obligatory	file(s)
	On-NP	From-LOC	Optional	Storage Media
	On-NP	At-LOC	Optional	Printer
	From-NP	From-LOC	Optional	Storage Media
	To-NP	At-LOC	Optional	Printer
e.g. Print file sample.txt on the Printer.				

Figure 3: An Example English Case Frame.

Command	Purpose	Different Ways of saying the command
Revoke	To remove rights from a user	1. Extract 2. Strip 3. Remove
Nprint	Network printing	1. Print 2. Copy 3. Send
Setpass	To change the password	1. Change 2. Rename 3. Set password

Table 2: Some Novell Commands with Different Ways of Saying Them.

the Arab world to interact with computer operating systems in both Arabic and English.

References

Al Daimi, K., and Sinan, N. 1993. 'Communicating with Operating Systems in Natural Language'. In *International Journal of Intelligent Systems*, 8(6):671-684.

Al-Khrisat, M. 1992. Structuring the Arabic Lexicon and Thesaurus with Lexical-Semantic Relations to Support Information Retrieval. Ph.D. Dissertation, Computer Science Department, Illinois Institute of Technology, Chicago, IL.

Allen, J. 1995. *Natural Language Understanding, Second Edition*. Redwood City, CA: The Benjamin/Cummings Publishing Company, Inc.

Alsamara, K. 1996. An Arabic Lexicon to Support Information Retrieval, Text Generation, and Parsing. Ph.D. Dissertation, Computer Science Department, Illinois Institute of Technology, Chicago, IL.

Buchheit, P. 1991. INFANT: A Connectionist Like Knowledge Base and Natural Language Processing System. Ph.D. dissertation, University of Illinois at Chicago.

Buchheit, P. 1995a. A Conversational Interface for File Management in an Operating System. In *Proceedings of*

the 1995 Midwest Artificial Intelligence and Cognitive Science Society Conference, 43-47.

Buchheit, P. 1995b. A Natural Language Interface for File Management in an Operating system. In *The 1995 Annual Meeting of the American Society for Information Science (SIG/ALP: The Automated Language Processing Special Interest Group)*, Chicago, IL.

Buchheit, P. 1996. A General-Purpose Natural Language Interface to an Operating System. In *The International Journal of Intelligent Systems*, 11(7): 409-427.

Cohen, Philip. 1992. 'The Role of Natural Language in a Multimodel Interface'. In *Proceedings of the Fifth Annual Symposium on User Interface Software and Technology*, 143-149.

Dardaine, J. 1995. Towards the Semiautomatic Generation of IITROLE: A Case Model Incorporating Syntactic, Semantic and Pragmatic Information. Ph.D. Dissertation, Computer Science Department, Illinois Institute of Technology, Chicago, IL.

Fillmore, C. J. 1968. *The Case for Case..* In E. Bach and Robert Harms (eds.) *Universals in Linguistic Theory*, Holt, Rinehart, and Winston, New York: 1-88.

Li, X. and Xing, D. 1992. General Natural Language for Operating System. *SIGART Bulletin*, 3(4): 28-32.

Manaris, B., Pritchard, J. and Dominick, W. 1994. Developing a Natural Language Interface for the Unix Operating System. In *SIGCHI Bulletin*, 26(2): 34-40.

Pin-Ngern, S. 1990. A Lexical Database for English to Support Information Retrieval, Parsing and Text Generation. Ph.D. Dissertation, Computer Science Department, Illinois Institute of Technology, Chicago, IL.

Wendlandt, E. and Driscoll, J. 1991. Incorporating a Semantic Analysis into a Document Retrieval Strategy. In *Proceedings of the Fourteenth Annual International ACM/SIGIR Conference on Research and Development in Information Retrieval*, 270-279.

Wilensky, R., Chin, D. and Arens, Y. 1984. Talking to Unix in English: An Overview of UC. *Communications of the ACM*, 27: 574-593.

Wilensky, R., Chin, D., Luria, M., Martin, J., Mayfield, J. and Wu, D. 1988. The Berkeley Unix Consultant Project. *Computational Linguistics*, 14 (1): 35-84.

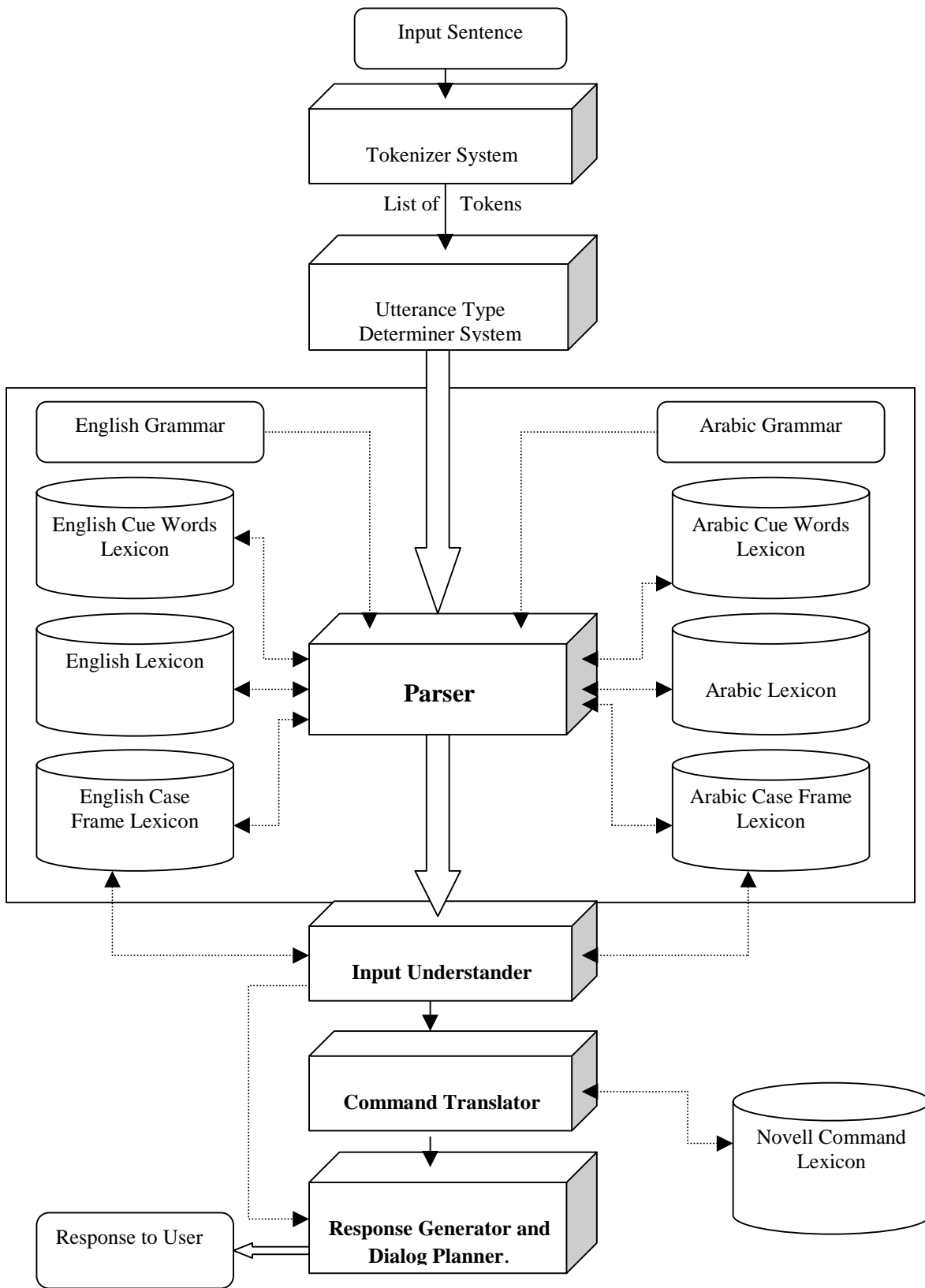


Figure 4: The Architecture of the Proposed System.