I N T E R F A C I L E : Linguistic Coverage and Query Reformulation

Yvette Mathieu   Paul Sabatier

CNRS - LADL
Université Paris 7
Tour Centrale 9 E
2 Place Jussieu
75005 Paris

1. INTRODUCTION

The experience we have gained in designing [3,6,7] and using natural language interfaces has led us to develop a general natural language system, INTERFACILE, involving the following principles :

- the linguistic coverage must be elementary but must include phenomena that allow a rapid, concise and spontaneous interaction, such as anaphora (ellipsis, pronouns, etc.)
- the linguistic competence and limits of the interface must be easily and rapidly perceived by the user.
- the interface must be equipped with strategies and procedures for leading the user to adjust his linguistic competence to the capacities of the system.

We have illustrated these principles in an application : a natural language (French) interface for acquiring the formal commands of some operating system languages. (The examples given here concern DCL of Digital Equipment Corporation).

2. INTERFACILE's ARCHITECTURE

The system is composed of :

- a portable syntactic grammar of French written in the formalism of Metamorphosis Grammars [2].
- a domain dependent lexicon-grammar [4].
- a domain dependent knowledge base written in a subset of predicate logic (Horn Clauses).

Operating on these linguistic data and domain knowledge :

- a syntactic parser coupled with the grammar and the lexicon-grammar (1) produces intermediate analyses of questions, and (2) generates some messages in case of errors.
- a semantic parser transforms the syntactic analysis into a semantic representation by consulting the knowledge base.
- an evaluator consults the knowledge base and produces answers to questions given their semantic representation.
- a general system guides the user and helps him to formulate and reformulate his queries.

The system is entirely programmed in PROLOG II and runs on various computers (VAX, SPS and micro-computers). Responses are given instantaneously.

Here is an example of a dialogue with INTERFACILE :

```
> comment créer un fichier ?

      - utilisez la commande :
      CREATE  nomfichier

> comment l'imprimer ?

      - utilisez la commande :
      PRINT  nomfichier

> comment modifier son contenu ?

      - utilisez l'éditeur avec la
        commande :
      EDIT  nomfichier

> comment copier un fichier dans un
  répertoire ?

      - utilisez la commande :
      COPY  nomfichier  [nomrépertoire]

> comment renommer ce dernier ?
      - utilisez la commande :
      RENAME/DIRECTORY  [ancienrépertoire]
                        [nouveaurépertoire]
```

<Dialogue 1>

(The prompt ">" precedes the user's queries, the prompt "-" precedes INTERFACILE's answers. English translations of the dialogues are given in the appendix).

3. LINGUISTIC COVERAGE

INTERFACILE handles only elementary syntactic structures. In order to ensure a concise and spontaneous dialogue, proforms have been introduced such as :

- pronouns (le, la, les, en, y, lui, leur, ...)
- possessives (son, sa, ses, ...)
- demonstratives (celui-ci, ceux-ci, ce dernier, ...)

If an ambiguity is detected during the resolution of a proform, INTERFACILE asks a question, as in the following dialogue :

```
> comment copier un fichier dans un
  répertoire ?

     - utilisez la commande :
       COPY  nomfichier  [nomrépertoire]

> comment l'effacer ?

     - "l'" renvoie à :
                          fichier ?      (1)
                          répertoire ?  (2)
       tapez le numéro de votre choix.

                 <Dialogue 2>
```

Proform resolution is based solely on lexical and syntactic information. This strategy is vital in the framework of a didactic system oriented towards beginners, as the following dialogue shows :

```
> comment envoyer un message à un
  utilisateur ?

     - utilisez la commande :
       MAIL

> comment le supprimer ?

    - "le" renvoie à :
                      message ?        (1)
                      utilisateur ?    (2)
      tapez le numéro de votre choix.

> 2

    - INTERFACILE n'en sait rien, mais
      sait comment :

                 envoyer un message
                 écrire un message

                 supprimer un fichier
                 supprimer un répertoire
                 supprimer un message

                 <Dialogue 3>
```

If the proform resolution had been based on the domain semantics, the system would have produced the answer for suppressing a message.

4. QUERY REFORMULATION

Given the plethora of syntactic structures and the complexity of mental processes involved in natural language understanding, the linguistic competence of interfaces yet remains limited. If we resort to such systems to request a service or obtain information that we lack, we generally know much more than the interface itself on the various ways for formulating our query.

We argue that the quality of a natural language interface must be estimated not only in terms of its linguistic coverage but also in terms of the procedures for dealing with incorrect, extragrammatical (i.e. correct but not expected by the interface) or semantically deviant sentences.

Knowing that error recovery is a complex task in the "restricted" framework of programming languages (limited syntax and rigorously defined semantics), one can appreciate the difficulty of endowing natural language interfaces with such capabilities.

One approach to the problem of error recovering is to "accept" user queries that deviate from the system's expectations and (unbeknown to the user) manipulate them into a form that can be analysed. For an overview of this approach, see [1]. This method is costly to implement, and moreover has two theoretical faults :

(1) There is no guarantee that the system's interpretation of a "deviant" query will correspond to the user's intention in posing the question.

(2) There will always be a limit to the degree of deviation the system can tolerate before it gives up trying to analyse a user query. But if the transformation of deviant queries into recognizable structures is performed automatically, the user will never know where this limit lies, and may believe that the system can make sense of anything he enters. The distinction between the sentences the system interprets correctly and those which it entirely rejects will seem arbitrary to him.

The strategy developped and implemented in INTERFACILE is quite different. Our philosophy is not to mislead the user concerning the system's real capabilities, but rather to lead his linguistic competence to that of the interface.

Here is an example illustrating such a strategy :

```
> comment ajouter à un fichier le contenu
  d'un fichier ?

     - pour "ajouter", utilisez la
       construction :
        ajouter    [quelque chose]
                 à [quelque chose]

      complétez (ou tapez / pour annuler)

   > comment ajouter ...
```

<Dialogue 4>

Here is another example :

```
   > comment envoyer un message dans un
     terminal ?

      - pour "envoyer", utilisez les
        constructions :

        envoyer     [quelque chose]
                  à [quelqu'un]

        envoyer       [quelque chose]
                  sur [quelque chose]

        complétez (ou tapez / pour annuler)

   > comment envoyer ...
```

<Dialogue 5>

INTERFACILE explains exactly which part of the question it doesn't understand : an unknown word, a verb with an incorrect preposition or unexpected order of complements, etc. In case of spelling errors, solutions for correction are given to the user [8].

To guide the user in reformulating his query, the grammar and the lexicon-grammar are consulted as knowledge bases. The user may reformulate his query with the information given by INTERFACILE.

(At any moment, the user can also obtain the lists of verbs, nouns, prepositions, pronouns, etc.).

When the user has an erroneous or incomplete view of the application domain, INTERFACILE points out his misunderstanding, as in the following dialogue :

```
   > comment imprimer le fichier d'une
     version ?

      - fichier de quoi ?  INTERFACILE parle
        plutôt de :

           fichier (tout simplement)
           fichier d'un répertoire
           version d'un fichier
```

<Dialogue 6>

Another example :

```
   > comment initialiser un fichier ?

      - INTERFACILE n'en sait rien, mais
        sait comment :

           créer un fichier
           éditer un fichier
           effacer un fichier

           initialiser un disque
           initialiser une disquette
```

<Dialogue 7>

The guidance given here goes beyond linguistic reformulation. It concerns the semantics of the application domain. Note that the information given to the user is not preprogrammed, but is (1) deduced from the knowledge base and (2) generated by the linguistic component (grammar, lexicon-grammar and parsers).

5. CONCLUSION

As far as the domain of application is concerned, INTERFACILE has much in common with systems like UC [9] or INTERIX [5].

The originality of our system lies in the effort we have devoted to :

- linguistic phenomena such as anaphora that ensure a natural dialogue.
- strategies and procedures for reformulation in order to live with the unavoidable linguistic limitations of any interface.

Our research and experience in developing applications in these two domains have led us to believe that it is not practical to add procedures to an already existing system to make it user-friendly. On the contrary, these problems must be taken into account in the design phase, as we have done in designing INTERFACILE.

6. ACKNOWLEDGEMENTS

We are grateful to Hannah Blau for her fruitful comments on a preliminary version of this paper.

We thank the Centre Mondial d'Informatique (CMI, Paris) for giving us the opportunity to develop a version of INTERFACILE on a VAX 785.

7. REFERENCES

[1] Carbonell J., Hayes P., "Recovery Strategies for Parsing Extragrammatical Language", American Journal of Computational Linguistics, 9, 3-4, 1983.

[2] Colmerauer A., Les Grammaires de Métamorphoses, GIA, Luminy, Université Aix-Marseille II, 1975, and "Metamorphosis Grammars" in Natural Language Communication With Computers, Bolc L. ed., Springer Verlag 1978.

[3] Duchier D., Sabatier P., INTERFRANCE: Un prototype d'interface en langue française, Rapport de recherche, CGE, Marcoussis, 1982.

[4] Gross M., "Lexicon-Grammars and the syntactic analysis of French", Proc. of COLING, Karlsruhe, 1984.

[5] Guez S., Sabbagh S., "INTERIX: Un système d'aide à l'utilisation d'Unix", Conférence AFCET, Paris, 1984.

[6] Oliveira E., Pereira L., Sabatier P., "ORBI: An expert system for environmental resources evaluation through natural language, Proc. of First International Logic Programming Conference, Marseille, 1982.

[7] Pique J.F., Sabatier P., "An informative, adaptable and efficient natural language consultable database system", Proc. of ECAI Conference, 1982.

[8] Sabatier P., A Prolog program for spelling correction, Note technique, LADL, 1985.

[9] Wilensky R., "Talking to UNIX in English: An overview of UC" Proc. of AAAI Conference, 1982.

APPENDIX : Translation of sample dialogues

Dialogue 1 :

```
  > how can I create a file ?

       - use the command :
         CREATE  filename

  > how can I print it ?

       - use the command :
         PRINT  filename

  > how can I modify its contents ?

       - use the editor, by the
         command :
         EDIT  filename

  > how can I copy a file into a
    directory ?

       - use the command :
       COPY  filename  [directoryname]

  > how can I rename the latter ?

      - use the command :
        RENAME/DIRECTORY  [olddirectory]
                          [newdirectory]
```

Dialogue 2 :

```
  > how can I copy a file into a directory?

       - use the command :
         COPY  filename [directoryname]

  > how can I delete it ?

       - "it"  refers to :
                      file ?        (1)
                      directory ?   (2)
         enter the number of your choice.
```

Dialogue 3 :

```
  > how can I send a message to a user ?

       - use the command :
         MAIL

  > how can I suppress it ?

       - "it" refers to :
                           message ?    (1)
                           user ?       (2)
         enter the number of your choice.
  > 2
     - INTERFACILE doesn't know, but he
       knows how to :

                    send a message
                    write a message
                    suppress a file
                    suppress a directory
                    suppress a message
```

Dialogue 4 :

```
  > how can I append to a file the content
    of a file ?

       - for "append" use the structure :
         append [something] to [something]

         complete (or type / to cancel)

  > how can I append ...
```

Dialogue 5 :

```
  > how can I send a message in a terminal?

       - for "send", use the structures :

         send [something] to [somebody]
         send [something] on [something]

       complete (or type / to cancel)

   > how can I send ...
```

Dialogue 6 :

```
  > how can I print a file of a version ?

       - file of what ?  INTERFACILE talks
         about :
                     a file
                     a file of a directory
                     a version of a file
```

Dialogue 7 :

```
  > how can I initialize a file ?

       - INTERFACILE doesn't know, but he
         knows how to :
                  create a file
                  edit a file
                  delete a file
                  initialize a floppy disk
                  initialize a disk
```