Parsing Strategies for a Spoken Language Processing System

T. L. Soto, J. G. Amores
{teresa,gaby}@babel.us.es
JULIETTA, Universidad de Sevilla, Spain

J. F. Quesada
josefran@cica.es
CICA, Sevilla, Spain
jfq@babel.us.es
JULIETTA, Univ. de Sevilla, Spain

Abstract

This paper describes the parsing strategies developed for Iris, an NLP system integrated with a speech recogniser and a dialogue manager. We will describe the syntactic notation which accounts for phenomena such as free sentence order in Spanish and in spoken language, as well as parsing strategies for sentence fragmentation into autonomous blocks. The grammar rule notation includes a novel typified semantic nuclei specification (CTAC) which simplifies the interface between the parser and the dialogue manager. In addition, the parsing module includes several strategies for the detection and correction (whenever possible) of the deficiencies originated at the speech recognition stage (over-recognition, under-recognition, close-recognition and mis-recognition). Finally, we present some provisional results obtained with the first prototype designed with the ideas described in this paper.

1 Introduction

This paper describes the parsing module of Iris, an NLP system which receives the string generated by a speech recogniser as its input and passes its output to a Dialogue Manager (DM) module. The paper is organised as follows. Sections 2 and 3 discuss how the characteristics of the spoken language affect the design and implementation of NLP systems. Section 4 outlines the parsing strategies we have implemented in Iris in order to solve the problems posed by spoken language. Section 5 describes the CTAC protocol, a process which ensures a smooth interface between the parser and the dialogue manager. Finally, section 6 concludes with an evaluation of the overall system at the current stage of development.[1]

2 Spoken Language and Recognised Language

The design, implementation and evaluation of any parsing system is conditioned by the characteristics of spoken natural language, as well as by the inaccuracies originated during recognition. Usually, the main problems that an NLP module has to face when analysing spoken input are the following: how to cope with relaxed grammatical structures typical of the spoken language and with the errors pro-

[1]This work has been partially funded by the Spanish CICYT under grant 95-0214-OP.

duced at the recognition stage.

Spoken language utterances usually contain grammatically incorrect constructions, syntactic free order, words which are not present in the lexicon, incomplete constructions due to elision, intratextual constructions or anaphora, and speech disfluencies, including repetitions, stutters and speech repairs (Heeman 1998). In this paper we focus on the characteristics of spoken language as syntactic constructions. Syntactic constructions in Spanish are characterised by exhibiting a relatively free order. This general characteristic of Spanish is even more unpredictable in instances of spoken language.

In addition to the notion of spoken language, we should introduce the concept of "recognised language", i.e., the statements actually recognised by the speech recogniser and which have to be processed by the NLP module. At present, even the most robust speech recognisers show a high error rate when applied to continuous speech and speaker independent contexts. Our study of the interface between the Speech Recognition (SR) and parsing modules has led us to classify recognition inaccuracies into four groups: *Over-Recognition*, *Under-Recognition*, *Close-Recognition* and *Mis-Recognition*. These will be described later. It is, thus, necessary to design parsing strategies which detect and solve these problems, whenever possible.

Besides, since we are dealing with a conversational system, the parsing module has to be able to analyse speech samples which are typical of a conversation, such as: doubt and hesitation expressions, colloquial or idiomatic expressions, deictic constructions, intratextual or anaphoric constructions, pronominal constructions, elision, etc. (Allen et al. 1996; Bear et al. 1992; Heeman 1998; Siu & Ostendorf 1996).

3 Parsing Spoken Language

The domain for which Iris has been developed is characterised, at the level of the parsing module, as follows:

- Natural spoken language.
- Integration with a speech recogniser in the input.
- Integration with a dialogue manager in the output.

The parser has as its input an instance of spoken language which has been generated by a speech recogniser. Because of this, some parsing mechanisms must be developed so that the parser is capable of dealing with phenomena such as incomplete grammatical constructions, free order, etc.

This is a serious source of problems for unidirectional parsing algorithms, such as Earley (Earley 1970), those based on chart (Kay 1980) or the GLR algorithm (Tomita 1987; Tomita 1991).

Basically, these algorithms would have to keep a set of all possible analyses during the parsing process as a result of the syntactic free order which is characteristic of spoken language and because of the speech disfluencies produced at the recognition stage. The algorithms should be relaxed so that they could permit the incomplete application of grammar productions (in order to allow for incomplete grammatical constructions and Mis-Recognition deficiencies). However, all these strategies would be computationally very expensive.

Therefore, a natural spoken language domain will benefit from a bidirectional algorithm and the use of a semantic–driven grammar. With this goal in mind, we have used the SCP algorithm (Quesada 1997) as a central parsing strategy. This algorithm has been augmented with a set of strategies which are described below.

Next, we present the main characteristics of the SCP (Syntactic Constraint Propagation) algorithm (Quesada 1996; Quesada 1997; Quesada 1998). If we bear in mind the threefold level in the analysis of the problem of parsing (computational, linguistic and formal), the SCP algorithm can be considered as a very efficient algorithm from the computational viewpoint, as it shows a performance of around 10,000 to 20,000 words per second for natural language grammars.[2] It also guarantees the maximum linguistic coverage for natural languages (it is complemented with a unification algorithm, thus allowing the direct treatment of unication–based grammars). It is also mathematically robust: the algorithm has been demonstrated as formally correct and sound.

The computational layer includes a specific memory management model and a strategy for grammar compilation. This module has been designed with the goal of efficiency. The linguistic layer is in charge of general applicability, and basically includes a mechanism for the integration of the algorithm with unification grammars. Finally, at the formal level, the mathematical kernel proposed permits the demostration of the correctness and soundness of the algorithm (Quesada 1997).

[2]This performance refers to the original parsing algorithm only. That is, without having been adapted to the Iris application, which incorporates some pre–processing, post–processing and unification.

4 Parsing Strategies in Iris

In this section we describe the different mechanisms which have been implemented in Iris so that the system can deal with instances of spoken language and recognised language.

4.1 VOID Category: Words without Relevant Semantic Information

Every entry in the lexicon is associated with a semantic category. Thus, for a sentence such as:

(1) **Muestreme los vuelos de la tarde de Boston a New York** (Show me the afternoon flights from Boston to New York)

from the ATIS domain and in Spanish, the system generates the following lexical layer:

INFO FLIGHT TIMEDAY CITY CITY

As we can infer from the example, not every lexical entry has been passed on to the parser. Ignored entries correspond to grammatical words, such as determiners, prepositions, conjunctions, etc. In order to deal with words of this sort, the system incorporates the lexical category VOID, which merely means that a lexical entry does not provide any relevant semantic information for the domain, and, therefore, it is dispensable. This strategy may also be applied to colloquial and idiomatic expressions since they only provide pragmatic and discourse–related information. If a lexical entry has been identified as VOID, it is not even treated by the parser. The system may parse or may not parse this entry according to the specific domain and corpus by means of activating or de-activating the corresponding command (**VoidWordsIgnore**).

4.2 Not Found Words: Words not Included in the Lexicon

It may be the case that the lexical databases recognisable by the speech recognition module is larger than the lexicon in the NLP module. This happens when the same recognition system is being used for different applications. In the example below "abuelos" (grandparents) has been recognised instead of "vuelos" (flights), but this word does not belong to the domain.

(2) **Muestreme por favor los abuelos de la tarde de Boston a New York** (Please show me the afternoon grandparents from Boston to New York)

In most cases, these words are unpredictable. That is to say, they may be considered as irrelevant in the domain and just be ignored, as in the VOID case. However, we could also assume that those words have some semantic content and should be dealt with by the parser somehow. To solve this problem, Iris has been equipped with the **NotFoundWordIgnore** and the **NotFoundWordDefault** commands. When the former is activated, any word which is not in the lexicon is treated as a VOID category. Otherwise, the user may configure the system so that it assigns one or more semantic categories to any word not found in its lexicon. In either case the analysis process will never be interrupted and Iris will be able to generate an output.

In future research work, we are considering the possibility of provisionally incorporating the unkown entry into the lexicon and match it with a specific category. Provided all the information required by the DM is complete once the assignment of the new category to the unknown word has been done, an estimation calculus on the number of successful analyses with the new word would determine its definitive incorporation into the lexicon.

4.3 The Optional (?) and Random_Order (&) Operators: Free Word Order

As we described in the introduction, natural spoken language is characterised by having a very free syntactic order. Furthermore, Spanish may be considered a free word-order language. For NLP applications where the input is a sample of spoken language, Spanish may show a great deal of syntactic free order. To cope with this problem, Iris incorporates two syntactic operators at the grammatical level: the optionality (?) and the random_order (&) operators. Consider for example the following grammatical rule:

(1:statement1 → INFO? [FLIGHT TIME_DAY? ORIGIN? DEST?]&)

This rule is a good instance of the two operators mentioned above. With the ? operator, one rule only suffices to analyse a wide variety of statements, such as:

(a) **Muestreme los vuelos de Boston a New York** (Show me the flights from Boston to New York)

(b) **Muestreme los vuelos de la tarde** (Show me the afternoon flights)

With the & operator, Iris allows for at least 22 different phrase combinations of the string which appears between square brackets.

4.4 Speech Recognition Deficiencies

We are now going to describe the mechanisms developed in Iris to deal with the recognised language. First of all, we will describe the phenomena of *Under–Recognition*, *Over–Recognition*, *Close–Recognition* and *Mis–Recognition*.

- **Under–Recognition:**

  Under–Recognition has to do with instances of recognised language where words which were originally uttered by the speaker have not been recognised. For example:

  (3) **original:** *muestreme los vuelos de la tarde de Boston a New York*
  (Show me the afternoon flights from Boston to New York)
  recognised: *muestreme vuelos tarde de Boston a New York*
  (Show me afternoon flights from Boston to New York.)

  Where the determiner has not been recognised.

- **Over–Recognition:**

  Over–Recognition has to do with instances of recognised language where words which were not originally uttered by the speaker have been recognised. For example:

  (4) **original:** *muestreme los vuelos de la tarde de Boston a New York*
  (Show me the afternoon flights from Boston to New York)
  recognised: *muestreme los vuelos de la tarde cabina de Boston a New York*
  (Show me the afternoon flights cabin from Boston to New York)

- **Close–Recognition:**

  Close–Recognition has to do with instances of recognised language where the recogniser generates words similar to the ones originally uttered by the speaker. In a high percentage, the words which have been closely recognised belong to the same morphological category and have a very strong semantic similarity. This is the case of the word *muestra* in the following example, which has been closely recognised from the original *muestrame*:

  (5) **original:** *muestrame los vuelos de la tarde de Boston a New York*
  (Show me the afternoon flights from Boston to New York)
  recognised: *muestra los vuelos de la tarde de Boston a New York*
  (Show the afternoon flights from Boston to New York)

- **Mis–Recognition:**

  Mis–Recognition has to do with instances of recognised language where the speech recogniser generates words which have no similarity with those originally uttered by the speaker. Under this term we group all those cases where recognition is seriously defective, as in the following example:

  (6) **original:** *Ayuda*
  (Help)
  recognised: *Ocho*
  (Eight)

4.5 Semantic–Driven Grammar: Close–Recognition

Iris is capable of dealing with some cases of recognition deficiencies of the Close–Recognition type. As we have defined it before, Close–Recognition happens whenever a similar word to the one uttered by the speaker is generated at the recognition stage. Since Iris incorporates a semantic–driven grammar, any morphological entry which belongs to the same semantic category will convey the same meaning and thus will be analysed on equal terms by the parser. That was the case of *muestra* closely recognised from the original *muestrame*. The two words belong to the same semantic class and the analysis process will not be affected.

4.6 Parsing of Partial Strings: Over–Recognition and Under–Recognition

Iris incorporates a mechanism which is capable of dealing with recognition deficiencies of the types of Over– and Under–Recognition. If a statement has not been properly recognised, the system can extract partial substrings from the complete string and analyse them. Provided the DM gets all the information needed, the parsing process goes on with the analysis of the following statement. In case the DM requires extra information which has not been properly completed with the parsing of the partial strings, the DM can still keep the information obtained from the first analysis and generate a question to the user. At this point, the user will have to add only the information which is missing.

5 The CTAC Protocol

As stated above, the NLP module we describe in this paper must interact with a DM, which results in a different class of potential problems, since the parser output has to be adequately integrated into the protocol defined by the DM. A novel interface protocol (the CTAC protocol) has been defined so that the formal and computational properties of the interface between the parser and the DM module be as simple as possible.

The semantic categories of the grammar are associated with typified feature structures both at the lexical level and the non–terminal level. One of the novelties of our notation is that both terminal and non–terminal nodes contain the same internal backbone consisting of four features: CLASS, TYPE, ARG(uments), and CONTENT. In addition, grammar rules are augmented with functional equations which are used by the unification module to generate a parsed output which is then sent to the DM.

As an illustration, consider the following macros, grammar rules and lexical entries:

```
flight_info=(CLASS:Request,
     TYPE:Flight_Info1,
     ARG:[Origin,Dest])

DEST=(CLASS:Object,
      TYPE:Dest,
      ARG:[City]|[Airport]|
  [State]|[Country])

ORIGIN=(CLASS:Object,
        TYPE:Origin,
        ARG:[City]|[Airport]|
    [State]|[Country])

(103:flight_info -> FLIGHT ORIGIN? DEST)
          {@up.Origin = @self-2;
           @up.Dest = @self-3;
           @ctac(); }
```

```
(LU:Boston,CAT:City)
(LU:"New York",CAT:City)
```

Each non-terminal node in the grammar is typified for its class, type and arg features, as the macro definitions above illustrate. Alternate possibilities in the argument list is shown by |, and obligatoriness by a comma inside an argument list. Grammar rules may include optionality operators (?) and are augmented with functional equations in an LFG-like fashion. That is, an equation like `@up.Origin = @self-2` states that the second `@self-2` element in the right-hand side of the rule will provide the Origin information to the mother functional equation. Notice that the `FLIGHT` node in the grammar rule above does not have any functional equation associated with it. This is so because in this domain *flight* is taken as a default semantic constituent. Finally, the `@ctac()` function flattens the resulting feature-structure and normalises the output so that it may be understood by the DM. Basically, it adds a new `CONT`(tent) feature, in which it will store the actual realisation of arguments. If the input semantic categories are not matched against the appropriate argument, a `Null` realisation will appear.

For example, the resulting structure after parsing the sentence *Muestra los vuelos de Boston a New York* would look like this:

```
(ARG:[Origin,Dest],
 CLASS:Request,
 TYPE:Flight_Infoi,
 CONT:[Boston,New York],
 Origin:
    (CLASS:Object,
     TYPE:Origin,
     ARG:[City]|[Airport]|[State]|[Country],
     CONT:[[Boston]|[Null]|[Null]|[Null]],
     City:
        (CLASS:Object,
         TYPE:City)),
 Dest:
    (CLASS:Object,
     TYPE:Destination,
     ARG:[City]|[Airport]|[State]|[Country],
     CONT:[[New York]|[Null]|[Null]|[Null]],
     City:
        (CLASS:Object,
         TYPE:City)))
```

6 Results, Conclusions and Future Work

In this paper we have described a series of parsing strategies in Iris, a conversational NLP system which takes as input the (N-)best candidate(s) generated by a speech recogniser and produces as output a structure understandable by a dialogue manager. These strategies solve common problems in spoken language parsing such as over-recognition, under-recognition, close-recognition, and mis-recognition, as well as the problem of free sentence order. A novel protocol has been described which ensures a straightforward interface between the parser and the dialogue manager. This paper has shown work in progress. So far, we have tested the viability of the notation described, including the CTAC protocol, for a corpus of 7,652 words and 3,115 sentences in the application domain. The grammar contains 280 rules.

Below is a short summary of the statistics provided by the implementation of a prototype designed according to the ideas described in this paper.

```
*********************
* GLOBAL STATISTICS *
*********************
*STRUCTURES
* Input Sentences ...:    3115
* Input Words .......:    7652
* Nodes .............:   23003
* Events ............:  390098
*TIME
* Analysis ..........:111.950 s.(100.0 %)
*    Lexical Analysis: 69.370 s. (62.0 %)
```

```
*    Parsing ........: 42.580 s. (38.0 %)
* Total Time ........:111.950 s.
*PERFORMANCE
*Sentences Analyzed per Second: 27.825
*Words Analyzed per Second....: 68.352
**********************
```

During this phase, we have concentrated on the functionality of the model. Nevertheless, the overall performance (68 words/second) is well enough to use the system in real-time applications. Also, in the second stage of the project, we hope to improve the efficiency of the system taking advantage of SCP (section 3). While the efficiency of SCP is 10,000-20,000 words/second, the efficiency of Iris decreases until 68 w/s due to the not-optimized layer of processing of Iris (optional and free word order operators, partial strings parsing, void words, etc.).

References

J. F. Allen, B. W. Miller, E. K. Ringger & T. Sikorski. 1996. A Robust System for Natural Language Dialogue. In *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics*, June 1996.

J. Bear, J. Dowding & E. Shriberg. 1992. Integrating Multiple Knowledge Sources for Detection and Correction of Repairs in Human–Computer Dialog. In *Proceedings of the 30th Annual Meeting of the Association for Computational Linguistics*, 1992. 56–63.

Jay Earley. 1970. An Efficient Context–Free Parsing Algorithm. *Communications of the ACM*, 13(2), 94-102.

Peter Anthony Heeman. 1998. *Speech Repairs, Intonational Boundaries and Discourse Markers: Modeling Speakers' Utterances in Spoken Dialog.* PhD dissertation, Department of Computer Science, University of Rochester, New York.

Martin Kay. 1980. Algorithm Schemata and Data Structures in Syntactic Processing. *CSL-80-12 Xerox Palo Alto Research Center.*

José F. Quesada. 1996. Bidirectional and Event–Driven Parsing with Multi Virtual Trees. *II International Conference on Mathematical Linguistics*, Tarragona, May 1996.

José F. Quesada. 1997. *El algoritmo SCP de analisis sintactico mediante propagacion de restricciones. [The SCP parsing algorithm based on syntactic constraints propagation].* PhD Thesis. July 1997. University of Seville.

José F. Quesada. 1998. Overparsing. *International Workshop on Mathematical Linguistics*, State College, Pennsylvania State University, April 1998.

M Siu & M Ostendorf. 1996. Modeling Disfluencies in Conversational Speech. In *Proceedings of the 4rd International Conference on Spoken Language Processing (ICSLP-96).* 382-391.

Masaru Tomita. 1987. An Efficient Augmented Context–Free Parsing Algorithm. *Computational Linguistics* 13 (1-2), 31-46.

Masaru Tomita. 1991. *Generalized LR Parsing.* London: Kluwer Academic Publishers.

S. R. Young & M Matessa. 1991. Using Pragmatic and Semantic Knowledge to Correct Parsing of Spoken Language Utterances. In *Proceedings of the 2nd European Conference on Speech Communication and Technology (Eurospeech)*, Genova, Italy. 223-227.