International Symposium on Reference Resolution for Natural Language Processing (RRNLP 2002), p. 101-107, Alicante (Spain), June 3-4 2002.

Automatic Evaluation for Anaphora Resolution in SUPAR system[1]

Antonio Ferrández; Jesús Peral; Sergio Luján-Mora
Dept. Languages and Information Systems
Alicante University - Apt. 99
03080 - Alicante - Spain
antonio@dlsi.ua.es, jperal@dlsi.ua.es, slujan@dlsi.ua.es

Abstract

This paper presents the evaluation module integrated in the system called *Slot Unification Parser for Anaphora Resolution (SUPAR)*. This module allows us to evaluate automatically anaphora resolution, which is a very important issue in the current state of the art of anaphora resolution. It can evaluate different kinds of anaphors (e.g. pronouns, definite descriptions, etc.), and it provides a tool that facilitates the anaphoric tagging of texts. The texts to tag anaphorically are independent from the language. Therefore, it can evaluate anaphora resolution in different languages. Presently, we have tagged 921 Spanish pronouns and 1,163 English pronouns. In the future, this module will allow different researchers to test their anaphora resolution algorithms on the same texts.

1 Introduction

Anaphora resolution is one of the most active research areas in Natural Language Processing (NLP). The comprehension of anaphora phenomenon is an important process, and given that it is crucial in any application that pretends to deal with natural language, it has been deeply studied in the last years. We have a set of Conferences and Workshops that focuses on anaphora problem such as the Discourse Anaphora and Resolution Colloquium (DAARC), the ACL/EACL Workshop on Operational Factors in Practical, Robust Anaphora Resolution (1997), or the ACL Workshop Coreference and its Applications (1999). We also have several journals that have focused on anaphora problem such as the Special Issue on Anaphora Resolution in Machine Translation (Machine Translation, 1999), or the Special Issue on Anaphora and Ellipsis Resolution (Computational Linguistics, 2001).

During the last years, there have been many proposals to resolve different kinds of anaphors. For example, those anaphora resolution implementations that rely on constraint and preference heuristics which employ information originating from morpho-syntactic, syntactic, or shallow semantic analysis. However, there is not a comparative evaluation of all these systems on the same texts and languages since co-reference evaluation that was carried out in MUC-6 and MUC-7 in 1995 and 1998 respectively. Since then, several efforts have been carried out in order to set a common evaluation measures (Barbu and Mitkov, 2001; Byron, 2001), but it is clear that a comparative evaluation between approaches on the same texts is desirable. That is to say, it is required an *evaluation workbench* that allows the comparison of different algorithms, although they use different pre-processing tools, but on the same data.

In our previous works (Ferrández and Peral, 2000; Palomar et al., 2001), we have faced up this comparison by means of implementing several baselines or well-known strategies. In this way, we could obtain a fair comparison on the same language (we have usually worked on Spanish texts) and with the same pre-processing tools.

In this paper, we present the evaluation module that has been integrated in our system, called *Slot Unification Parser for Anaphora Resolution*

[1] This paper has been partially supported by the Spanish Government (CICYT) projects numbers TIC2000-0664-C02-02 and TIC2001-3530-C02-02

(SUPAR). This module allows an automatic evaluation of different kinds of anaphors and in different languages. In the future, it will be available for everybody that wants to compare its anaphora resolution system with ours.

In the following section, SUPAR system is briefly presented. This is followed by the description of the proposed evaluation module and tagging tool. Finally, in the last section, the evaluation results are presented.

2 SUPAR system

The Slot Unification Parser for Anaphora Resolution (SUPAR) has been previously presented in Ferrández et al. (1999). It is a general-purpose computational system and a modular system that can be applied to different applications (e.g. Machine Translation or Information Retrieval). Presently, it is included in the Question Answering system, with which we have participated in the TREC-9 and TREC-10 Conferences (Vicedo and Ferrández, 2000), which can show the SUPAR's robustness.

SUPAR is described graphically in Figure 1. It can work on different languages. Currently, it can work on Spanish or English texts, although it can be easily adapted to other languages by representing the syntactic information in the *Slot Unification Grammar (SUG)* formalism, and using the proper POS-tagger.

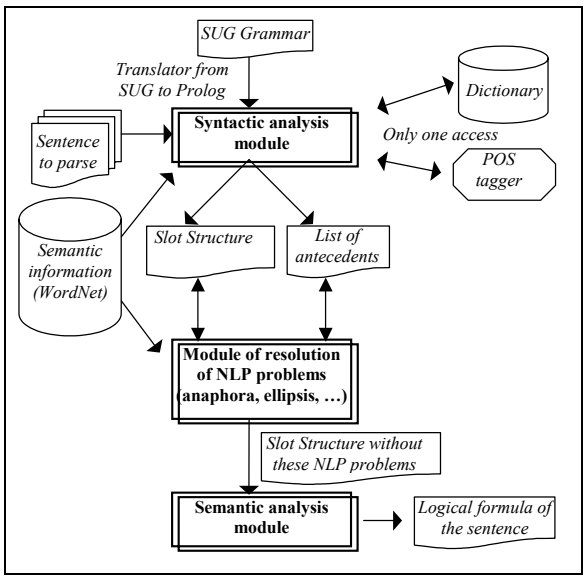

Figure 1. SUPAR's Architecture.

The **syntactic analysis module** takes as input this SUG grammar and the output of the POS tagger. It allows one to carry out either partial or full parsing of the text, by selecting the constituents we want to parse. For example, anaphora resolution is carried out by parsing coordinated prepositional phrases, coordinated noun phrases, pronouns, conjunctions and verbs in whatever order they appear in the text. In this case, NPs can include relative clauses, appositions, coordinated PPs and coordinated adjectives. Conjunctions are used to segment sentences into clauses.

The following **module of resolution of NLP problems** deals with anaphora resolution as well as other NLP problems such as extraposition, ellipsis or PP-attachment. This module builds a list of candidate antecedents from previous sentences in order to solve intersentential anaphora. This list stores knowledge obtained from previous stages, such as morphological (number or gender of the antecedent), syntactic (e.g. the head and modifiers), and knowledge about the position of the antecedent in the text (e.g. the identifier of the antecedent or the position of the antecedent with reference to the verb of the clause). In Figure 2 an example of the information stored for two antecedents is shown, and how anaphora resolution is solved.

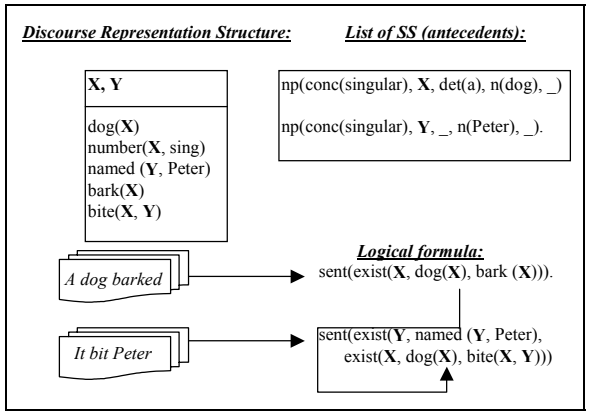

Figure 2. An example of anaphora resolution.

The **SUPAR anaphora resolution algorithm** distinguishes between constraints and preferences, and it uses as input the morphological knowledge of the POS tagger, and the partial syntactic structure of each sentence. Moreover, it carries out a clause segmentation of each sentence in order to

run c-command restrictions, in spite of the partial parsing (Ferrández et al., 1998; Palomar et al. 2001). Furthermore, it can automatically detect pleonastic pronouns, e.g. *It's tea-time*. Finally, it can deal with other kinds of anaphors as well as pronouns such as definite descriptions (Muñoz et al., 2000) or zero-pronouns in Spanish (Ferrández and Peral, 2000).

The output of this module is a structure (SS) where all the anaphors have been resolved. The anaphora resolution means that all the information about the anaphor and its antecedent is stored in the SS: entity identifier, morphological and syntactic information, and knowledge about the position of the antecedent in the text. In this way, co-reference chains are stored in the whole SS of the text, and it allows one to evaluate SUPAR with or without anaphora resolution as we have carried out in the last two TREC Question Answering tracks. This SS is then used in the last module of the system, in which the final logical formula of the sentence is obtained.

3 The evaluation module

This section describes the evaluation module proposed in this paper. This evaluation is applied on anaphora resolution process. It uses as input an anaphorically tagged text by means of the tagging tool that is described in the following subsection, and it returns several evaluation measures that are described in subsection 3.2.

3.1 The tagging tool

The text is anaphorically tagged by means of a tool that facilitates the tagging process. It receives as input texts that have been automatically POS-tagged and segmented into sentences. As well as these texts, it works on the output of SUPAR. In this way, it can correct the failures in its anaphora resolution process. Therefore, it can be considered as a semi-automatic anaphor-tagging tool. It receives the following set of text files (as it is shown in Figure 3):

- A file with the text segmented into words and sentences.
- A file with all the anaphors detected in the text, and all the information about them such as:
  - Their position in the text: number of sentence and words.
  - The kind of anaphor: e.g. *persRefl* stands for reflexive pronouns, or *persIt* stands for an *it* pronoun. Where all these labels will be used to present evaluation measures grouped into these types of anaphors.
  - The type of reference: anaphors (<), cataphors (>), exophors (*!*) or any kind of reference (e.g. bound anaphora or references to new objects in discourse: *$*).
- A file per each anaphor that contains the position of each possible candidate in the text, a list with those candidates that satisfy constraints, and the final solution.

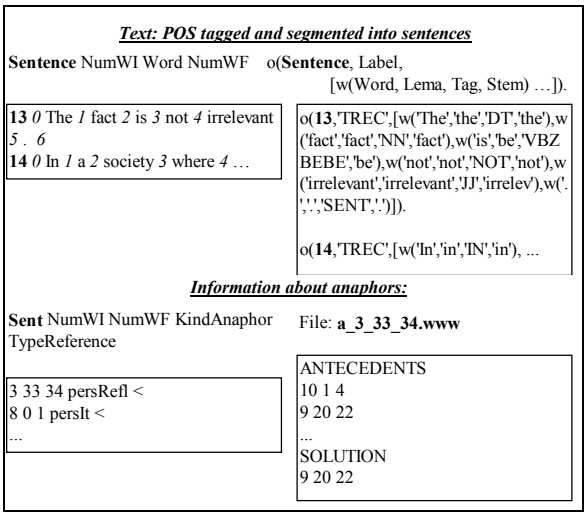

Figure 3. The input for the tagging tool.

In Figure 4, the tool is shown, where the active anaphor appears bold cursive and underlined, the remaining anaphors appear underlined, the solution of the active anaphor crossed-out, candidates discarded by restrictions in yellow and candidates that satisfy restrictions in green. This tool also allows one to set the co-reference chains, since an anaphor can have as solution another anaphor. In this way, in the following automatic evaluation module, it would be considered as a right solution whether the system returns as selected antecedent the other anaphor or its solution. Moreover, it can tag different kinds of anaphors such as definite descriptions, zero-pronouns, cataphors or exophors as it is shown in Figure 3.

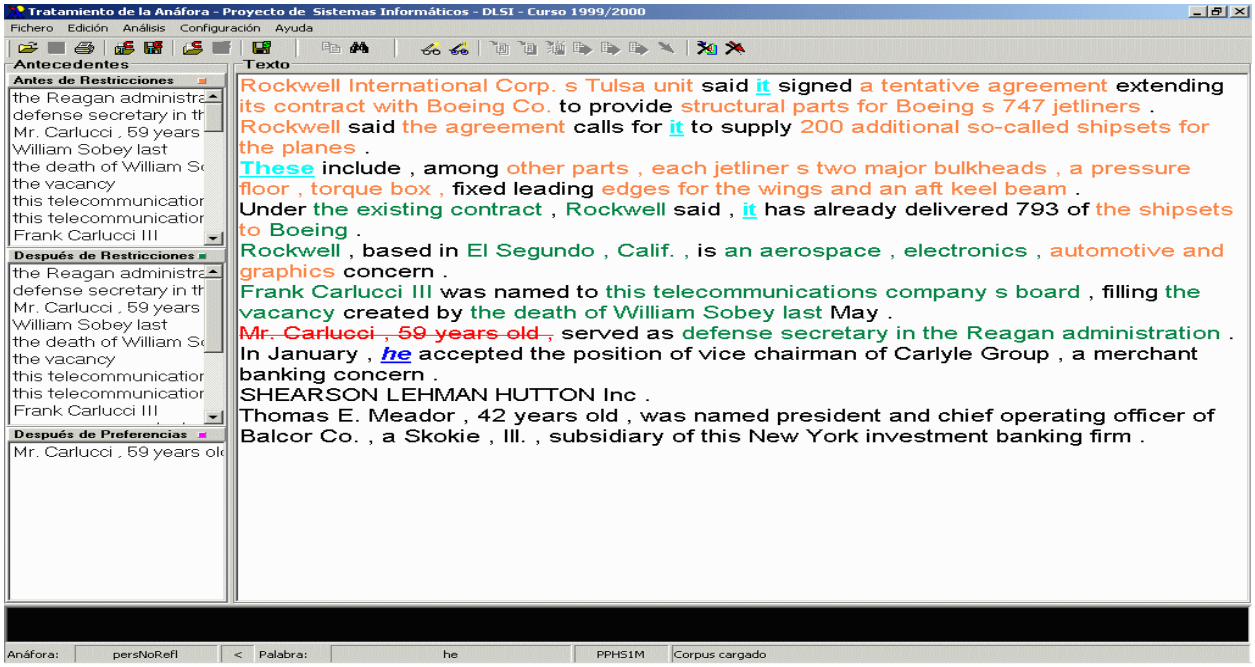

Figure 4. The tagging tool.

The tagging format proposed in this paper is different from the one used in MUC-6 or MUC-7 although it could be easily exchanged. For example, in the SGML tagging of MUC, the TYPE attribute is equivalent to the attribute *TypeReference* in Figure 3. Moreover, the MIN attribute is performed in our tagging proposal, by means of the partial parsing on POS tagging information, by means of identifying the head of the antecedent, and by means of the evaluation measures presented in the following sub-section. Finally, we should remark that zero-pronouns (that were not tagged in MUC) have also being tagged by marking the verb of the clause in which the zero-pronoun appears (which contains all the morphological information about the pronoun, i.e. person and number knowledge). Anyway, in the future, we expect to create an interface between both formats.

Researchers that would like to compare their anaphora resolution systems with SUPAR on the same corpora, just have to provide the system with the information described in Figure 3. The interface will be very similar and can be accessed in *http://gplsi.dlsi.ua.es/SUPAR*, and they will obtain the similarity measures described in the following subsection. In case that one would need the text segmented into NPs, the texts are also provided in the format presented in Figure 5. Where words, sentences, NPs, heads of NPs, PPs, genitives, appositions, relative clauses, verbs, clauses and anaphors are marked by SGML tags, including their lemma and POS tags.

```
<@OOO,1,example of sentence syntactically tagged>
<@CCC>
<@SNS,suj,comun,personaOAnimal,>
<@NSN>
Rockwell NNP rockwell
International NNP international
Corp. NNP corp.
<@/NSN>
<@GSJ>
s POS s
Tulsa NNP tulsa
unit NN unit
<@/GSJ>
<@/SNS,suj,comun,personaOAnimal,>
<@VBC>
said VBD said
<@/VBC>
<@/CCC>
<@CCC>
<@ANF>
<@SNS,suj,pronEnglish,,>
it PPH1R1 it
<@/SNS,suj,pronEnglish,,>
<@/ANF>
...
```

Figure 5. Corpus tagged with NP, clauses, etc.

3.2 The evaluation tool

The output of the tagging tool is the input in the evaluation module. In this text file all the following information is stored: position of anaphors, their solutions, type of anaphor, co-reference chains, etc. The evaluation module will return a text file as the shown in Figure 6. It means the different number of anaphors (corresponding to the categories of anaphors described in Figure 3) that have been resolved (exit and failure), the number of candidates before restrictions, after restrictions and the number of anaphors that has been resolved just with constraints.

```
persIt<exit 2 77 28 0
persIt< failure 4 133 68 0
persRefl<exit 1 35 1 1
persNoRefl<exit 40 1414 570 0
persNoRefl<failure 14 511 200 0

NumberSentences: 100 – NumberWords: 2265 Words/Sent: 22.65
```

Figure 6. Output of evaluation module.

The evaluation is carried out by automatically comparing the SUPAR's output with the tagged text. It can be performed with two different measures:

- By comparing the heads of the solution stored and the head of the solution given by SUPAR.
- By comparing the whole solution with the whole solution given by SUPAR.

As it can be observed, the second evaluation measure is stricter than the first one. The first one is used when an automatic parsing of the text is carried out, and differences between the tagged solution and the SUPAR's solution could be produced by differences in the parsed noun phrase. For example, in *Peter saw the boy with the telescope*, let us suppose that the system chooses as the NP *the boy with the telescope* as solution, but the tagged solution is *the boy*. Then, it would success in the first measure, but it would fail in the second one.

Therefore, it can be easily obtained the evaluation measures reported in other works (Barbu and Mitkov, 2001; Byron, 2001) such as precision, recall, success rate and critical success rate. Moreover, the evaluation module can additionally return the results when the errors produced by previous incorrectly resolved anaphors are automatically resolved. That is to say, if an anaphor is incorrectly resolved, the evaluation module automatically substitute it by the proper solution stored in the tagged text (although it is obviously considered as a failure in the final evaluation). In this way, the following anaphors will not be affected by the present error. For example, if an anaphor chooses as its solution the antecedent that is the solution of the previous anaphor (i.e. it is establishing a co-reference chain), then the second anaphor will not fail in case the first anaphor is incorrectly resolved.

4 Some SUPAR's evaluation results

This section shows some evaluation results of the anaphora resolution module, both in pronominal anaphora and elliptical zero-subject constructions, with no semantic knowledge:

- SUPAR automatically detects pleonastic pronouns, e.g. *It's tea-time*, with a precision of 91%, evaluation that has been run on 970 pronouns of the TREC Federal Register collection.
- For Spanish zero-pronouns, personal or demonstrative pronouns on texts of different genres (newspapers, technical manuals, novels, etc.), we have obtained the following success rate, i.e. number of correctly solved pronouns divided by the total number of solved pronouns: 921 / 1,144 = 81%.
- For English pronouns: 835 / 1,163 = 74%.

The resolution failures have been caused by the input errors (i.e. POS tagger, partial parser, sentence and clause segmentator) and the lack of semantic knowledge.

Next, we are presenting some figures that can give us an idea of the SUPAR efficiency. The following experiments have been carried out on 887 randomly selected documents of the TREC collections: the Los Angeles Times (LAT) and the Foreign Broadcast Information Service (FBIS), as it is described in Table 1. In this Table, we can observe that anaphora resolution module takes about 89% of the total running time (3,389 seconds). This time is quite higher than the parsing time, whose speed reaches up to 2,001 words per second, which makes a global SUPAR speed up to 256 words per second. These measures of time have been obtained on a Pentium III, 1000 GHz, 128 Mb RAM. The anaphora module takes so long

time because it has to segment each sentence into clauses and it has to create the list of possible candidates, which contains every noun phrase in the text. The list of candidates should contain all configurational knowledge, i.e. the verb of the clause, position with reference to the verb of the clause, if they are included in a prepositional phrase or in another noun phrase, the number of times that the noun phrase has appeared in the text and/or with the verb of the clause, etc. We have measured the time of processing the list of candidates as an 89.7% of the anaphora resolution time in LAT collection. In this evaluation, SUPAR has resolved 216 reflexive pronouns and 8,722 personal and demonstrative pronouns. For all kinds of pronouns, we have only considered the noun phrases in the same sentence as the pronoun or in the previous four sentences, and 396,977 candidates have been found, which means an average of 44.4 candidates per pronoun. After constraints (c-command and morphological agreement), there are 17.8 candidates per non-reflexive pronoun, and 1.3 candidates per reflexive pronoun, on average. This means that a high degree of ambiguity has to be finally resolved by preferences.

	N. Doc.	*Total Words*	*Average Words per sentence*	*Total Time (second)*	*SUPAR Speed (w/sec.)*	*Parsing Speed (w/sec.)*	*% Time anaphora*
LAT	370	281,149	20.9	1,580	178	1,939	89.1 %
FBIS	517	462,221	26.1	1,809	256	2,001	89.4 %

Table 1. SUPAR evaluation.

An example of the segmentation of a sentence into clauses and candidates has been extracted from LAT collection in (1), where the candidates have been numerated and delimited between square brackets, and the two clauses are divided by the conjunction *that*.

(1) *[[David R. Marples's]$_1$ new book, his second on [the Chernobyl accident of [April 26, 1986]$_2$]$_3$]$_4$, is [a shining example of [the best type of [non-Soviet analysis into [topics]$_5$]$_6$]$_7$]$_8$* ***that*** *only recently were [absolutely taboo in [Moscow official circles]$_9$]$_{10}$.*

5 Conclusions

This paper describes the evaluation module that has been included in the *Slot Unification Parser for Anaphora Resolution (SUPAR)* system. This module automatically evaluates different kinds of anaphors: pronouns, zero-pronouns, and definite descriptions. Moreover, it can works on texts in different languages. In this paper we have shown several evaluation measures on Spanish and English texts.

It has also been presented a tool that facilitates the anaphorical annotation of texts. It works independently of the language of the text, and it can tag different kinds of anaphors, cataphors or exophors. At the moment, we have tagged 921 Spanish pronouns and 1,163 English pronouns. In the future, this module will allow different researchers to test their anaphora resolution algorithms on the same texts.

References

Barbu C. and Mitkov, R. 2001. Evaluation tool for rule-based anaphora resolution methods. Proceedings of the 39th Annual Meeting and 10th Conference of the European Chapter of the Association for Computational Linguistics (ACL'2001). 34-41. Toulouse, France.

Byron, D. 2001. The Uncommon Denominator: A Proposal for Consistent Reporting of Pronoun Resolution Results. *Computational Linguistics*. Special Issue on Anaphora and Ellipsis Resolution. 27(4): 569-577.

Ferrández A., Palomar M., and Moreno L. 1998. Anaphora resolution in unrestricted texts with partial parsing. Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics (COLING – ACL'98). 385-391. Montreal, Canada.

Ferrández, A., Palomar, M., and Moreno, L. 1999. An empirical approach to Spanish anaphora resolution. *Machine Translation*. Special Issue on Anaphora Resolution in Machine Translation. 14(3/4): 191-216.

Ferrández, A., and Peral, J. 2000. A Computational Approach to Zero-pronouns in Spanish. Proceedings

of the 38th Annual Meeting of the Association for Computational Linguistics (ACL'2000). 166-172. Hong-Kong, China.

Muñoz, R., Palomar, M., and Ferrández, A. Processing of Spanish Definite Descriptions. 2000. Lecture Notes in Artificial Intelligence (1793). Subseries of Lecture Notes in Computer Science. Springer Verlag. MICAI 2000: Advances in Artificial Intelligence. Mexican International Conference on Artificial Intelligence. Proceedings. Osvaldo Cairo, L. Enriguez Sucar, Franciso J. Cantu (Eds.). 526-537. Acapulco, Mexico.

Palomar, M., Ferrández, A., Moreno, L., Martínez-Barco, P., Peral, J., Saiz-Noeda, M., and Muñoz, R. 2001. An Algorithm for Anaphora Resolution in Spanish Texts. *Computational Linguistics*. Special Issue on Anaphora and Ellipsis Resolution. 27(4): 569-577.

Vicedo, J.L., and Ferrández, A. 2000. A semantic approach to Question Answering systems. Proceedings of the Ninth Text REtrieval Conference (TREC-9). Gaithersburg, Maryland, EEUU.