LOST IN *GRAMMAR* TRANSLATION

LOST IN GRAMMAR *TRANSLATION*

FABIO MASSIMO ZANZOTTO

SOMMARIO/*ABSTRACT*

In questo articolo descriviamo il tentativo di valutare un parser sintattico per l'italiano rispetto ad un corpus annotato. Il compito e arduo poiché il parser esistente e il corpus annotato sono stati sviluppati con grammatiche differenti.

In this paper we describe an attempt of evaluating a pre-existing Italian syntactic parser with respect to an annotated dependency treebank. The task is not so simple. The pre-existing syntactic parser and the annotated dependency treebank realize two different grammars.

Keywords: Syntactic parsing, Grammar comparison, Parsing Evaluation

1 Introduction

Grammars are models that determine a view on relations that words have in sentences. Even if the grammatical intuition can suggest that modeled phenomena cannot be so different, the actual *formal* grammars of a given natural language may diverge. In NLP, this problem can intrinsically limit some very important activities:

1. *grammar learning*: different annotated corpora cannot be easily used to induce a single probabilistic grammatical model;

2. *the evaluation of parsing systems* (as noted for example in [3]): syntactically annotated corpora and evaluated pre-existing parsers may not share the same grammar.

For the Italian language, the first problem is extremely relevant. There are at least three different syntactically annotated corpora: the Turin Treebank[1] (TUT), the Venice Italian Treebank[2] (VIT), and the ISST[3]. None of them is comparable in size with the English Penn Treebank. This limits the possibility to have reliable induced grammars for Italian. Initial studies have shown that probabilistic grammars induced on a small corpus have not impressive performances [5]. Building larger corpora is then needed. We have been working on defining general translators that can transform more expressive grammatical annotations in less expressive ones [1]. These translators can be used to merge corpora with different annotation schemes. Such bigger corpus is better suited for learning reliable grammars.

The second problem instead is the one that we had to face in the Evalita comparative study. We wanted to assess the performances of a pre-existing parser, CHAOS[4] [2], against an annotated corpus based on a completely different grammar, the TUT. We then translated the grammatical interpretation produced by CHAOS in the target grammatical representation.

In this paper we describe the parser we used and how we translated its syntactic interpretation for the purpose of the evaluation (Sec. 2). We analyze the results (Sec. 3). Finally, we draw some conclusions (Sec. 4).

2 Adapting a pre-existing Italian syntactic parser

The pre-existing Italian parser is realized on a modular and lexicalized model [2]. This model uses the extended dependency graphs (XDG) as syntactic interpretations. The XDGs allow the representation of tree forests in a single graph. An $\mathcal{XDG} = (C, D)$ is a dependency graph whose nodes C are *constituents* and whose edges D are the *grammatical relations* among the constituents (see Fig. 1.(a)). Constituents are lexicalized syntactic trees with explicit *syntactic heads* and *potential semantic governors*. Dependencies in D represent typed and ambigu-

[1] http://www.di.unito.it/∼tutreeb/

[2] http://project.cgm.unive.it/

[3] http://si-tal.ilc.cnr.it/

[4] The parser can be downloaded at http://ai-nlp.info.uniroma2.it/

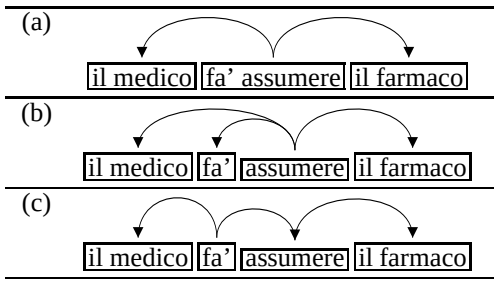

Figure 1: An XDG with two possible different translations

$$XDG'_{TUT} \rightarrow \boxed{T} \rightarrow \boxed{\text{Chaos}} \rightarrow \boxed{T^{-1}} \rightarrow XDG''_{TUT}$$

Figure 2: The overall architecture of the parser

ous relations among a constituent, the $head$, and one of its $modifiers$. Ambiguity is represented using $plausibility$ (a score between 0 and 1). In this modular model, the parser P is a composition of functions $(P_1, ..., P_n)$, i.e.,

$$P(xdg) = P_n \circ P_{n-1} \circ \ldots \circ P_2 \circ P_1(xdg)$$

Each function takes care of an activity such as tokenization, recognition of named entities, chunking, etc.

In addition to the original set of functions of Chaos, we developed a XDG ambiguity resolution module

$$P(xdg) = argmax_{D \in \overline{xdg}} p(D|xdg)$$

where $\overline{xdg}$ is the set of alternative XDGs derived from the xdg, D is one of the XDGs, and $p(D|xdg)$ is the probability of the D with respect to the original xdg. The probability model is similar to the one in [4].

To evaluate the parser we had to translate both the input POS tagged sentences to the Chaos internal grammar and to finally translate the Chaos output back in the external grammar. The overall process is described in Fig. 2. The function T is realized as described in [1]. As the Chaos grammar is less expressive than the TUT grammar this function is complete. On the contrary, the inverse function T^{-1} is only approximated as the target grammar is more expressive than the source.

3 Evaluation and error analysis

The results of the evaluation is presented in following table:

LAS	UAS	$LAS2$
47.615	62.11	54.895

Compared with the best parsers they are not satisfactory. This demonstrates that translating grammatical representations in other grammatical representations is not an easy task. The major source of errors in this case has been the inverse translation function T^{-1}. Its main task is to transform chunks to the related dependency subgraphs. This is not simple. Consider the example in Fig. 1. The translation of the constituent "*fa' assumere*" determines also the attachment sites of the dependencies from that constituent to the others. In the example, two solutions Fig. 1.(b) and Fig. 1.(c) are presented. Only (c) is admissible in the TUT interpretation. Choosing (b) is a catastrophic choice. Yet, (b) has to be preferred when the constituent is like "*e' assunto*". Translation rules strongly depend on the structure of the constituents. Writing these rules is like writing a grammar.

4 Conclusions and final remarks

This study confirms that it is hard to evaluate non-treebank parsers with respect to an annotated treebank [3]. However, the annotated corpus used in Evalita is too small to induce stable parsers or to push a TUT-centric development of syntactic parsers in the Italian community. We still need methods to reuse different annotated corpora to induce a single grammar [1].

REFERENCES

[1] A. Bahgat and F.M. Zanzotto. A dependency-based algorithm for grammar conversion. In *Proc. of LREC*, Genova, Italy, 2006.

[2] R. Basili and F.M. Zanzotto. Parsing engineering and empirical robustness. *Natural Language Engineering*, 8/2-3, 2002.

[3] J. Carroll, T. Briscoe, and A. Sanfilippo. Parser evaluation: a survey and a new proposal. In *Proc. of LREC*, Granada, Spain, 1998.

[4] M. Collins. Head-driven statistical models for natural language parsing. *Computational Linguistics*, 29, 2003.

[5] A. Corazza, A. Lavelli, G. Satta, and R. Zanoli. Analyzing an Italian treebank with state-of-the-art statistical parsers. In *Proc. of the TLT*, Germany, 2004.

CONTACT

FABIO MASSIMO ZANZOTTO
University of Rome "Tor Vergata"
Email: zanzotto@info.uniroma2.it

FABIO MASSIMO ZANZOTTO is an associate professor at the University of Rome "Tor Vergata". He has been working in building models for robust syntactic parsing, for shallow semantic analysis, and for knowledge acquisition from corpora. In the last years, he developed machine learning-based textual entailment recognition models.