RECUPERO DA FALLIMENTO CON UN PARSER LEFT CORNER (GRAFO)

RECOVERING FROM FAILURE WITH THE GRAFO LEFT CORNER PARSER

EMANUELE PIANTA

SOMMARIO/ *ABSTRACT*

GraFo è un parser di tipo left corner per l'analisi sintattica dell'italiano, basato su regole codificate manualmente con un formalismo a unificazione. Poiché la copertura linguistica della grammatica italiana di GraFo è ancora bassa, il parser produce analisi sintattiche complete per una percentuale ridotta di frasi. Questo articolo presenta alcune strategie per il recupero da fallimento nei casi in cui GraFo non riesca produrre una analisi sintattica completa. I vari approcci sono stati testati sui dati della campagna di valutazione EVALITA 2007.

GraFo is a left corner parser for Italian, based on explicit rules manually coded in a unification formalism. As the linguistic coverage of GraFo is still quite limited, the parser produces complete parse trees for a small percentage of sentences. This paper presents a number of strategies to recover from GraFo parsing failures. The various techniques have been evaluated on the data provided by the EVALITA 2007 evaluation campaign.

Keywords: parsing, Italian, left corner, failure recover.

1. Introduction

Deep parsers based on explicit representation of linguistic knowledge require big efforts for manual coding of lexical and syntactic information. As a consequence, in early stages of their development, they can suffer a lack of robustness due to low linguistic coverage, unless explicit steps are taken to implement fallback strategies able to recover from failure by producing partial syntactic analysis when complete parse trees are not available.

In this paper we illustrate how this issue is handled by GraFo, a linguistically motivated parser for Italian, based on lexical and syntactic rules manually coded in a unification formalism, built on top of the Prolog unification mechanism. The GraFo Italian grammar is inspired to the Lexical Functional Grammar (LFG) linguistic theory [2], and encodes various kinds of linguistic information in parallel: constituency, grammatical functions (e.g. subject, object), and semantics. When GraFo can parse a sentence, this will produce at the same time a constituency structure, an LFG functional structure, and a situation semantics representation.

The GraFo parser uses a non deterministic left corner parsing strategy [1]. Left corner parsing combines top down expectations with bottom up evidence, and has been described as compatible with findings about human sentence processing [3]. Also, the parser does not handle ambiguity through a chart parsing technique, but is based on the assumption that the first parse tree output by the parser is the right one. Of course this can happen only if a number of syntactic and semantic constraints are checked during the parsing process. GraFo exploits syntactic and semantic sub-categorization information for a consistent number of Italian verbs, adjective and nouns. Semantic constraints are expressed in terms of MultiWordNet synsets [4].

2. GraFo failures

GraFo relies on a vast amount of linguistic knowledge. We can expect that this makes the parser prone to failure, even more so because the grammar is in its early development stage. This assumption was tested on the data provided by the constituency parsing task of EVALITA 2007. Note that if we evaluate GraFo on the EVALITA data we are considering only the first of three kind of information provided by the parser (syntactic structure, functional information, semantic interpretation). But of course no functional structure or semantic interpretation is possible without the constituency structure. So, evaluating GraFo on the EVALITA constituency parsing task is crucial to assess its coverage. When applied on the EVALITA development set (1976 sentences), GraFo was able to produce a complete parse only in 10.8% of the cases. Thus, we decided to explore possible fallback strategies to increase the robustness of GraFo, at least for constituency analysis.

3. GraFo fallback strategies

As first step, we considered three fallback strategies in alternative to left corner parsing: (1) TAG: return a flat tree with a FRAG root, and PoS-tagged input words as branches. (2) CHUNK: the same as above but, if possible, PoS-tagged words are grouped into nominal, verbal, prepositional, adjectival, or adverbial flat chunks. (3) SHALLOW: if possible, chunks are further grouped trying to build clause level partial trees. These 3 strategies have been quickly implemented, and evaluated *per se* (i.e. ignoring left corner parsing). Here are the results in terms of Precision, Recall and F-measure, as calculated by the EVALB utility on the development corpus.

Table 1: Fallback strategies on EVALITA development

Strategy	Prec.	Rec.	F-m
TAG	50.01	4.50	*8.25*
CHUNK	47.92	33.77	*39.62*
SHALLOW	59.51	43.34	*50.15*

Then, we combined left corner parsing (LEFTC) with the best of the above alternative strategies. According to this approach, which we can label as LC+SHA, we first try LEFTC, and, if it fails, we resort to the SHALLOW fallback strategy. This achieves an F-measure of *54.02* on the development corpus, around 4 points higher than what we get by applying a basic shallow parser (*50.15*).

With the LC+SHA strategy we are actually using LEFTC only for 10.8% of the sentences. Note that if LEFTC does not produce a complete parse, this does not mean that it does not produce any parse at all. In fact it will produce a number of *possibly fragmentary partial parses*, one for each step of the parsing process. So, we devised a mechanism to: (a) evaluate the *best partial parse* (BPP) produced by LEFTC; (b) recover the BPP in case of failure; and (c) assemble BPP fragments.

The algorithm for evaluating the best BPP is based on the number of *unparsed words* left (UW) and the number of *recognized trees fragments* (RTF). Consider 2 partial parses PP_i and PP_j, at steps *i* and *j* of the parsing process. We assume that PP_i is *better* than PP_j, if $|UW_i| < |UW_j|$, i.e. we prefer partial parses covering a greater number of words. When $|UW_i| = |UW_j|$, PP_i is *better* than PP_j if $|RTF_i| < |RTF_j|$ – i.e. we prefer partial parses containing a smaller number of recognized tree fragments (the final step of a complete parsing contains only one tree).

We are now in the position to implement a new fallback strategy, let's call it LC+BPP+SHA: if LEFTC fails, then first recover the BPP. In many cases this will leave part of the sentence unparsed; then apply SHALLOW on the unparsed fragments. Unfortunately LC+BPP+SHA does not produce any improvement over LC+SHA, as the F-measure becomes *51.83* (vs *54.02).* However we now have at least partial LEFTC analyses, and hence also functional and semantic information, for *all* the sentences in the corpus (vs. 10.8%).

To maximize this result, we finally devised a further fallback strategy which can be labelled as LC+BPP+LC. Instead of analyzing unparsed fragments with SHALLOW we parse them with LEFTC itself. If the parser fails again, we skip the first word of the unparsed fragment, and run LEFTC on the rest, and so on until we cover all the sentence. The LC+BPP+LC achieves *41.96* F-measure on the EVALITA development set. The same algorithm evaluated on the EVALITA test set achieved a very similar *41.93* (ref. FBKirst _Pianta_PAR).

Table 2: GraFo on EVALITA development/test

Strategy	Corp.	Prec.	Rec.	F-m
LC+SHA	Dev.	56.52	51.74	*54.02*
LC+BPP+SHA	Dev.	55.69	48.48	*51.83*
LC+BPP+LC	Dev.	48.93	36.73	*41.96*
LC+BPP+LC	Test	45.49	38.91	*41.93*

REFERENCES

[1] Alfred V. Aho and Jeffrey D. Ullman. *The Theory of Parsing, Translation, and Compiling. Volume 1: Parsing*. Prentice-Hall, Englewood Cliffs, N.J, 1972.

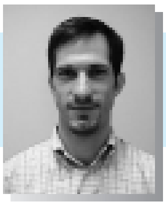
[2] Joan Bresnan. *Lexical-Functional Syntax*. Oxford: Blackwell Publishers, 2001

[3] Philip N. Johnson-Laird. *Mental Models: Towards a Cognitive Science of Language, Inference, and Consciousness*. Cambridge University Press, Cambridge, MA, 1983.

[4] Emanuele Pianta, Luisa Bentivogli and Christian Girardi. MultiWordNet: Developing an aligned multilingual database. In *Proc. of the 1st International Global WordNet Conference*, India, 2002, pp. 293-302.

CONTACT

EMANUELE PIANTA
FBK-irst, via Sommarive, 18, 38050 Povo (Trento)
Email: pianta@itc.it



EMANUELE PIANTA is researcher at FBK-irst, Trento. His research interests include development of multilingual resources (e.g. MultiWordNet), basic linguistic processors for Italian and English, parsing, and information extraction.