The Turin University Parser at Evalita 2011

Leonardo Lesmo
Dipartimento di Informatica – Università di Torino
Corso Svizzera 185 – I-10149 Torino – Italy
lesmo@di.unito.it

Abstract. This paper describes some extensions of the parser that has been used in the Evalita 2011 contest. The paper does not focuses on the actual parser, since it was already used in Evalita 2009: describing it would be a duplication of the description already given there. On the contrary, I address two extensions that has been adopted in a more recent version of the parser. The reason why this version was not used in Evalita is that it is based on a domain ontology not available for open domain texts and that more testing is required before using it in such larger domains.[1]

Keywords: Rule-based parsing, syntax-semantics interface, syntactic traces

1 Introduction

The Turin University Parser (TUP) is a rule-based parser that produces a dependency tree in what is called TUT (Turin University Treebank) format. This format is described in various documents downloadable from the TUT site [1].

This format was devised to enhance human readability and to include all useful dependency information. The arcs connecting the nodes of the parse tree (mainly words of a sentence) are labeled according to a set of labels organized in a hierarchy, thus enabling to annotate sentences at different level of specificity and detail. For instance, an adverb of negation can be linked to its governor as RMOD (Restrictive MODifier), ADVB-RMOD (ADVerBial Restrictive Modifier) or ADVB-RMOD-NEG (ADVerBial Restrictive Modifier with a semantic role of NEGation).

The TUT format is used in the annotation of the Italian TUT treebank, but it was also applied (to check its coverage) to small English and French corpora. It was also used for the annotation of a corpus of signed sentences in the Italian Language of Sign (LIS), developed in a project for the translation from written Italian to LIS (ATLAS). It will be delivered at the end of the project, i.e. around July 2012.

For the Evalita parsing task, TUP acted as a module of a pipeline that included the extraction of the words from the CONLL test file, the actual parsing, and the conversion of the result into CONLL format. I will not describe the details of the pre- and post-processing steps, since they do not affect the parser results.

With respect to the parser architecture, nothing has changed with respect to the version which ranked first (it drew with Pisa parser) in Evalita 2009. The architecture

[1] This work has partially been funded by the PARLI Project (Portale per l'Accesso alle Risorse Linguistiche per l'Italiano – MIUR – PRIN 2008) and by the ATLAS Project (Automatic Translation into sign LAnguageS – Regione Piemonte – Converging Technologies 2008)

is shown in Figure 1. A comparison of the results of Evalita 2009 with the ones of Evalita 2011 can be surprising, since TUP ranked last among the four participants in the 2011 competition. Certainly, the main reason for this was that all competitors improved their performances in the last two years. This did not happen for TUP, whose overall results have been a bit worse than in 2009.

Although this outcome may depend on many subtle factors, I believe there is one main reason for this apparently poor performance. It is deeply related with the rule-based approach which TUP is based on. Synthetically, I can say that the learning set has not been used at all. In a sense, this is both a weakness and a strength of the existing parser. It is a weakness, because a tuning of the parsing rules takes some human effort that could be greater than the one required for applying automatic learning procedures; it is a strength, because the parser appears to be robust enough to accept a comparison without any tuning.

It is difficult to assess both points (the weakness and the strength) with greater accuracy. With respect to the first one, I must say that probably, with a couple of week of human work, it could have been possible to gain a couple of points in parsing accuracy, that do not affect the rank, but could preserve the results obtained in the previous Evalita: it is a default of mine not having had the time for doing that. The second point is the one that enabled us to use TUP in many projects without substantial changes, with one exception that I will describe below. I will introduce it in this paper, although it is not fully related with Evalita, first because it can provide the reader with some feelings about the direction towards which TUP is moving, but also because it is a (partial) justification for not having paid to the tuning of the parser on the training set the attention it deserved.

The next section includes a brief description of the TUP modules.. The third section is devoted to ontology-based semantic interpretation (which should, in the next future, affect the parser behavior). The fourth section describes some extensions to the use of traces in TUP. A Conclusion section closes the paper.

2 The Parser Architecture

Concerning the architecture of the parser, the interested reader is addressed to the Notes of Evalita 09 [2]. I include here just a figure and a few words of comment.

After morphological analysis (based on a dictionary including around 26,000 entries) and POS-Tagging (based on handcrafted tagging rules), the parsing process is entered. It is split into 4 phases. The first of them applies a set of rules that, for each word, checks if any of the surrounding words can act as a dependent of it. The rules inspect the left and right context and are applied starting from “simpler” governors (e.g. adverbs) and going on to more complex ones (e.g. nouns). The result of this chunking phase is that most words of the sentence are grouped in chunks.

The second phase takes care of coordination, by trying to put together chunks separated by conjunctions. Most of this activity is carried out on the basis of dedicated procedures (rather than on declarative rules), but the analysis of coordination is notoriously difficult. After this, we have a set of larger chunks, that should act as verbal dependents; clause boundaries are detected via another group of rules, and each

verb is assigned a set of possible dependents. Each pair <verb, set of dependents> is matched with knowledge about verb subcategorization in order to separate arguments and adjuncts and to find out the syntactic role (subject, direct object, etc.) of each argument. It must be observed that, in this step, some traces are inserted in order to account for missing obligatory dependents (e.g. because of pro-drop). Finally, the overall structure is inspected, in order to ascertain that it has one and only one root, that there are no loops, and so on: in short that it is a real tree.

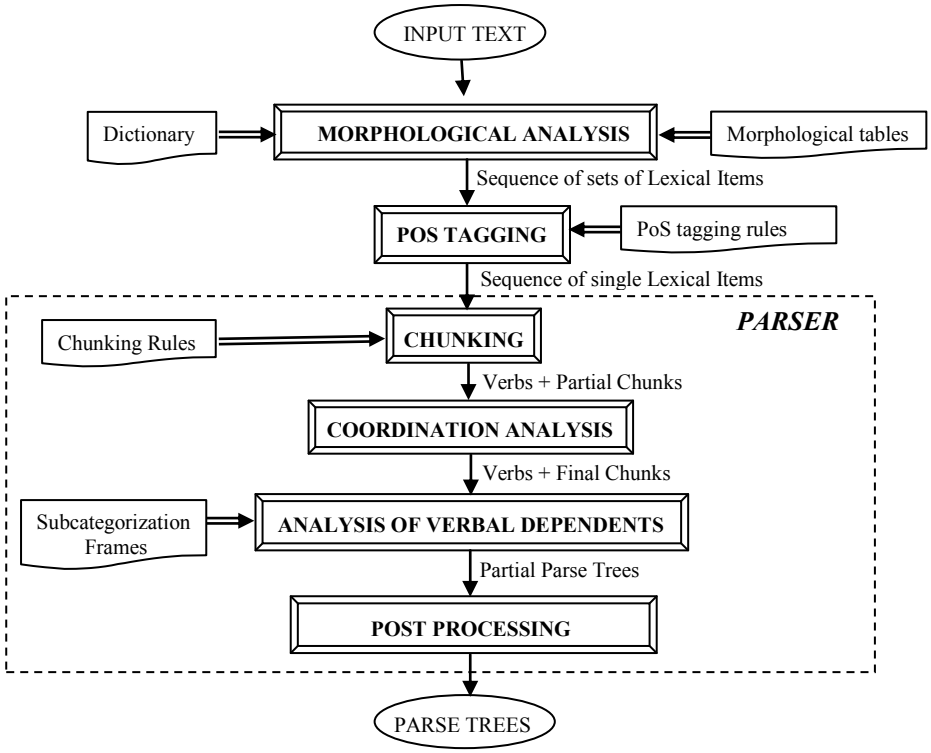

Fig. 1. Architecture of the parser

3 Syntax and Semantics

As stated before, TUP has been used in a number of projects, among which ATLAS (Automatic Translation into sign LAnguageS) [3]. The goal of ATLAS is the translation from written Italian to the Italian Language of Signs (LIS) of the deaf. According to our overall line of research, we hypothesized that translation can be achieved via a full understanding of the input followed by a generation phase. The result of the generator (which is in a kind of written LIS format) is then forwarded to a planner that determines the position of the hands [4] and, finally, to a module that takes care of the animation of the virtual character. Although it is well known that full

translation cannot be achieved with the current technologies, we exploited the fact that the implemented prototype addresses the specific domain of weather forecast. This section gives some hints about the way semantic interpretation is carried out.

It is not possible to survey here the research activities related with the interaction between natural language interpretation and ontologies. A good overview can be found in the Lecture Notes of the course given at ESSLLI 2007 by Paul Buitelaar and Philippe Cimiano [5]. In ATLAS, semantic processing is based on an ontology of the weather domain and is basically compositional (in dependency terms) although there are some exceptions. For translating a sentence as

(1) Domani avremo forti piogge nel sud dell'Italia
(Tomorrow, we will have strong rain in the South of Italy)

The parser builds the parse tree, that is then inspected starting from the root (the verb "to have"). First, the dependents of the verb are examined. The interpreter looks for the shortest path in the ontology linking *££to-have-1* (the ontology concept for this sense of the verb "have"[2]) to *£tomorrow* (an instance of *££deictic-day-description*). Then, "nel sud dell'Italia" (in the South of Italy) is interpreted as a locative expression (instantiating a "situation-location"); however, in this case, compositionality is somewhat bypassed, since the access to the ontology enables the interpreter to detect a reference to the "area" instance *£Italy-southern-area*. This is important, since the same area could be referred to via different expressions (as "Italia del sud", "Italia meridionale"). On the contrary, "strong rain" is fully compositional and is interpreted as "rain such the value of its property 'heaviness' is 'strong'". Finally, "we" (actually, a trace, in the Italian sentence) and the strong rain are assigned the thematic roles of "owner" and "owned-thing", by inspecting the thematic grid of the verbal concept *££to-have-1*.

The reason why I've spent some words about semantic processing is that this approach paves the way for fruitful interactions between syntax and semantics. In ATLAS, in fact (but not in Evalita, because of the absence of an ontology), the ontology is used to solve prepositional attachment ambiguities. Consider, for instance

(2) Le previsioni di neve del servizio meteorologico sono state confermate
The prediction of snow of the weather bureau have been confirmed

Here, we have that "del servizio meteorologico" (of the weather bureau) is preferentially attached to the closest possible governor, in this case 'snow'. And it is clear that only semantics can provide the cues for overcoming this preference.

In the development of the version of TUP used in ATLAS, the effort required for making syntax and semantics interact in a fruitful way has been rather limited, since the basic mechanism for taking the right decision is already available, i.e. the search for the shortest path between two nodes. What is needed is just that the search starts in parallel from the two attachment points, i.e. "neve" (snow) and "previsioni" (prediction). The shortest path found in this case links 'prediction' and 'bureau', so the correct attachment is chosen. Of course, this means that the 'chunking' module of the parser had to access the weather ontology, thus anticipating the semantic interpretation phase. This produced some inefficiency, since the same path had to be

[2] The prefix of the names of items in the ontology are used as mnemonics: *££* identifies concepts, *£* instances, @ relations. The fact that *££to-have-1* is a concept and not a relation is due the reification of states and events.

looked for twice: for attachment disambiguation and, later, for building the semantic representation. However, this can only be avoided by implementing an incremental interpreter, which would involve substantial changes to the entire architecture.

4 Traces

One of the main features of TUT is the presence of traces (empty nodes). This choice was made to preserve the projectivity of the parse tree and to include in the tree all obligatory dependents of verbs. The first point is related with movement, as in

(3) A Maria penso di fare un regalo
(To Mary, I think to give a present)

where 'a Maria' has been moved at the beginning of the sentence, but its correct attachment point is the verb 'to give'[3] . However, this attachment breaks projectivity, so what we do is to link 'to Mary' with 'think' as a VISITOR (syntactic link without semantic role), and to put under 'give' a trace linked as VERB-OBJ. In the same example, the subject of 'think' is pro-dropped, and a VERB-SUBJ trace is inserted. Another trace is inserted as VERB-SUBJ of 'give', because of the control of 'think'.

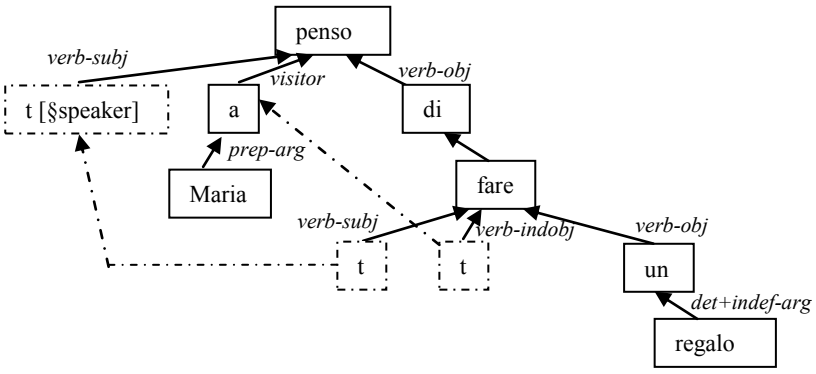

Fig. 2. Syntactic tree of "A Maria, penso di fare un regalo" (To Mary, I think to give a present)

In order to keep the same format for TUT and TUP, the parser is able to insert traces, in particular for missing arguments and in case of control (e.g. for modal verbs). In ATLAS, the mechanism of traces has been extended to cover more complex phenomena, as gaps. The next example is the same as (1), but the verb is missing.

(4) Domani forti piogge nel sud dell'Italia
(Tomorrow, strong rain in the south of Italy)

The meaning of (4) is the same as the one of (1), but we must have a way to specify that 'Tomorrow' is the time of an event that is in the class denoted by the ontology

[3] In Italian, which is partially free-word order, the example sounds much more natural than in English

concept *££to-have-1*. Analogously for 'strong rain' and 'in the south of Italy'. Since TUP is a robust parser that always builds a parse tree, so that also for sentence (4) a single-root connected tree is built. Consequently, the extension which is being described here is based on diagnostics that tend to identify unreasonable patterns in the resulting tree; in these cases, the tree is modified by inserting in a proper position the trace for the gap. One example of such a diagnostics is the following:

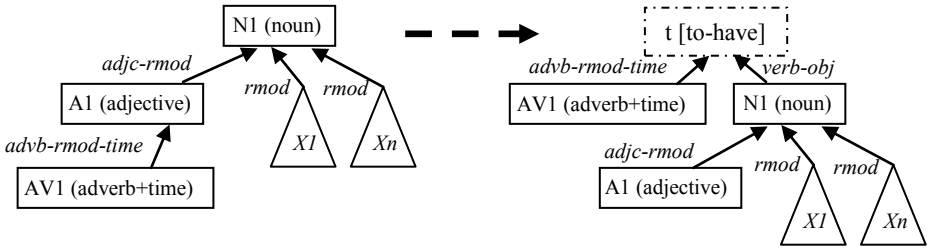

Fig. 3**.** A diagnostics for trace insertion and the resulting tree. Triangles represent full subtrees

Currently 7 such diagnostic rules are defined, but this covers just part of the whole ATLAS corpus (approximately 40%, i.e. 170 sentences), so that more rules will be defined in the next future. An assessment of the actual coverage of this mechanism on general texts will be carried out after the end of ATLAS.

5 Conclusions

This paper has described some extensions of the Turin University Parser that are only partially relevant to the Evalita contest, since they are not included in the TUP version used in Evalita. Notwithstanding, the effort required for their development partially justifies the poor performances of the parser, since this effort was devoted to future extensions and limited the refinements based on the Evalita learning set.

I think that parsing is just one (fundamental) step of linguistic processing, and that in the next future an effective parser must be part of any application devoted to text processing (document indexing, information extraction, sentiment analysis, …). But the parser cannot work alone: its output must be the starting point of other processes. This paper tried to show how a dependency parser can interact with other modules, in particular modules taking advantage of ontology-based semantic information. This implies that the resulting parse tree is reasonably complete, so that mechanisms guaranteeing this completeness (as traces) must find their place in the parsing process.

References

1. Turin University Treebank (TUT) homepage: http://www.di.unito.it/~tutreeb/
2. Lesmo, L.: The Turin University Parser at Evalita 2009. In: Poster and Workshop Proc. 11th Conf. of the Italian Association for Artificial Intelligence, Reggio Emilia, Italy, ISBN 978-88-903581-1-1 (2009)
3. ATLAS project homepage: http://www.atlas.polito.it
4. Ruggeri, A; Battaglino, C.; Tiotto, G.; Geraci, C.; Radicioni, D.; Mazzei, A.; Damiano, R.; Lesmo, L.: Where should I put my hands? Planning hand location in sign languages. In: Proc. Workshop on Computational Models of Spatial Language Interpretation and Generation, Boston, pp. 24—31 (2011)
5. Buitelaar, P.; Cimiano, P.: Ontologies and Lexical Semantics in Natural Language Understanding. A course given at ESSLLI 2007 (http://people.aifb.kit.edu/pci/ESSLLI07/)