Experiments with a Constraint-based Dependency Parser

Matteo Grella, Marco Nicola, and Daniel Christen

Parsit S.r.l.
{matteo.grella,marco.nicola,daniel.christen}@parsit.it
http://www.parsit.it

Abstract. This brief article describes our contribution to the Evalita 2011 Dependency Parsing Task. The Italian grammar has been expressed for the first time as a set of constraints that need to be satisfied by any parse tree. The constraints propagation technique is then applied to restrict possible analyses. Multiple solutions of a given sentence have been reduced to one (structural disambiguation) by weighting each relation of each different solution according to the number of occurrences of that relation in the indexed version of Italian Wikipedia created for the purpose. A detailed analysis of the results is given, including some consideration on the difference between the LAS and UAS values. The attachment score obtained is 96.16%, giving the best result so far for a dependency parser for the Italian language.

Keywords: evalita, italian, dependency parsing, constraints grammar, constraints propagation

1 Description of the System

The parser architecture used at Evalita 2011 Dependency Parsing Task (DPT) is one of a series of experiments we have been dedicating to in the past few months. The parser was tested against the Evalita 2009 DPT [4] dataset made available by the organizers after the competition and the results encouraged us to participate to Evalita 2011. The idea which stands behind our work was inspired by the Optimality Theory (OT) applied to syntactic analysis [2], whose linguistic model makes the language itself coming from the interaction between conflicting constraints, and mainly consists of viewing the parsing process as a finite configurations problem that can be formulated as a constraint satisfaction problem (CSP) [20]. Deciding for this approach has been for us the logical consequence of the considerations we made after several research and experiments with top-down and mainly statistical graph-based [7, 13] and transition-based [22, 15, 1] bottom-up parsers. We also analyzed in depth TULE [9], a particular rule-based parser (which played a great role in our experiment) developed by Leonardo Lesmo that got the best result (LAS 88.73%) at Evalita 2009 DPT. Even if we came to a constraint-based dependency parser autonomously, it's worthwhile to mention Maruyama [11] who was the first to propose a complete treatment of dependency grammar as a CSP, Harper and Helzerman [8], Menzel and Schröder [14] who further developed the concept, and the outstanding work by Denys Duchier [6] and Debusmann *et al.* [5].

In our approach the dependency parsing problem is reduced to the problem of finding a dependency graph for a sentence that satisfies all the constraints defined by the grammar. The constraint propagation technique has been successfully adopted for an "arc consistency" deterministic inferences process and a custom search strategy has been used to explore the solutions. The structure of the system is made of two components: the constraint solver module and the disambiguation module. The first one has been implemented in Ada 2005 [19] using Gecode [18] constraint programming [17] library which provides a constraint solver with state-of-the-art performances. A great effort has been required to define the grammar that, as already stated, consists of a set of constraints needed to be satisfied by any parse tree. We have first defined the general conditions for well-formed labeled tree, i.e. in the dependency graph each node has only one incoming edge, there are no cycles and there is precisely one root. Subsequently we have encoded the Italian grammar (subset of) in a formalism inspired by the Slot Grammar (SG) [12] wherein every lexical entry contains informations about category, morphological features and a set of slots (grammatical relations) and rules (dependency context constraints and word order constraints) for filling them. Our wide coverage lexicon has been used, which includes subcategorization for nouns, adjectives, verbs and adverbs. The grammar has been then fine-tuned on the TUT Treebank [10].

While the constraint solver implements the constraint propagation to restrict possible solutions, the disambiguation module enters the game when the first module has brought to multiple solutions. Please note that at this stage all the solutions satisfy all the constraints defined by the grammar. These multiple solutions represent a true structural ambiguity. One example is the well known prepositional phrase (PP) attachment (e.g. a PP that can be linked both to nominal and verbal node). The module explores the tree of each solution and assigns a score to every arc connecting either the two conjuncts in a coordination or the two atoms of a PP. The score of each of these relations in a given solution is computed by counting the number of occurrences of that relation in the Italian version of Wikipedia [21] previously analyzed by TULE parser and indexed by our Natural Language Indexer (NLI).[1] The score of the solution is obtained by simply adding the score of its relations. Since only one interpretation is allowed by Evalita 2011 DPT rules, in case of multiple solutions the one with the best score is chosen.

Because of the specific role of the two components described above, the system can be considered a hybrid rule-based and statistical-based parser.

---

[1] NLI is a component of the high performance Natural Language Retrieval Architecture (NLRA) designed and developed by Parsit S.r.l. which allows for complex queries on parse trees data through a particular pattern matching technique. Our indexed version of the Italian Wikipedia can be explored at `http://www.parsit.it/evalita2011`

2 Results

The Arc Accuracy we obtained in the official evaluation at Evalita 2011 DPT (300 sentences for a total of 7401 tokens) are **91.23%** for LAS[2], **96.16**% for UAS[3] and **93.87%** for LAS2[4], which are the best results according to all of the three measures used for the performance evaluation.

Another parameter which is out of the scope of Evalita 2011 DPT, but nevertheless of great importance to us, is the complete Correct Sentence Accuracy (CSA), labeled (CSA-L) and unlabeled (CSA-U), whose values are 22.33% and 57.00% respectively. The average LAS and UAS per sentence are 91.66% and 96.63% respectively. We also decided to evaluate the relationship between parsing accuracy and length of the sentences. To accomplish the above, we divided the set of sentences into five subsets (groups) of sentences whose lengths (no. of words) are 1-10, 11-20, 21-30, 31-40, 41-50 respectively. We than computed the average CSA for each group. The results are shown in Fig. 1 below.

Fig. 1. Accuracy relative to sentence length

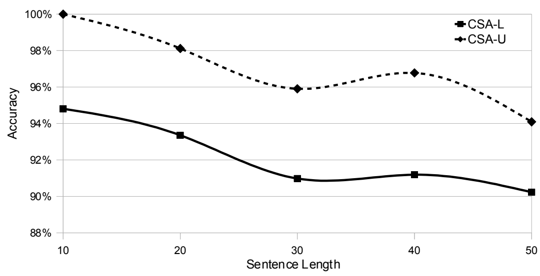


Table 1 shows the number of sentences belonging to each one of the five groups. In our experiment we considered meaningful groups 2, 3, 4 only.

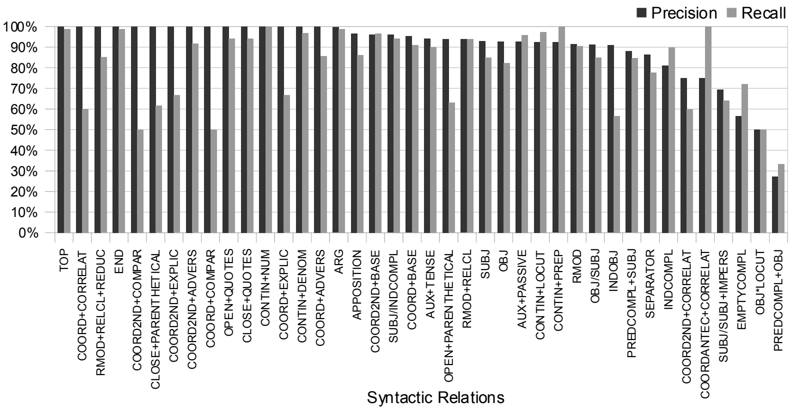
Table 1. No. of sentences in the groups

Group	Range	No. of sentences	Portion of dataset
1	1-10	15	5.00%
2	11-20	95	31.67%
3	21-30	108	36.00%
4	31-40	62	20.67%
5	41-50	20	6.67%

[2] Labeled Attachment Score (LAS), the percentage of tokens with correct head and relation label.
[3] Unlabeled Attachment Score (UAS), the percentage of tokens with correct head.
[4] Label Accuracy Score (LAS2), the percentage of tokens with correct relation label.

Fig. 2 assesses the recall and precision of every syntactic relation label attachment. Please note that syntactic relations labels for which both recall and precision values are 0 have not been reported in the graph.

Fig. 2. Recall and Precision of every syntactic relation label attachment.



3 Discussion

We focused our analysis on the errors made in labeling the relations of tokens with correct attachment, i.e. the errors that caused the difference of 4.93% between LAS and UAS. In this context, eight "cases" of systematic errors have been identified:

1. The parser assigned the deprel INDCOMPL, INDOBJ instead of the generic RMOD of the Gold dataset. Possible cause is that our lexicon has got a richer argumental structure than the one of the Gold dataset with respect to the given relation. Occurrences: 16.
2. The parser assigned the generic RMOD instead of the INDCOMPL, INDOBJ of the Gold dataset. Possible cause is that the Gold dataset has got a richer argumental structure than the one of our lexicon with respect to the given relation. Occurrences: 23.
3. The parser assigned the deprel OBJ instead of the OBJ*LOCUT of the Gold dataset. Possible cause is the same of point no. 2. Occurrences: 5.
4. The parser assigned at dependent of NOUN the generic RMOD instead of either SUBJ and OBJ syntactic role. Possible cause is the same of point no. 2. Another possible cause is that sometimes in the Gold dataset the relevant information (the deriving verb or "dummy" value) that allows the correct label assignment in the features column is missing. Occurrences: 75.
5. The parser assigned the EMPTYCOMPL instead of either SUBJ / SUBJ +IMPERS and OBJ of the Gold dataset and vice versa. Occurrences: 29.

6. The parser never matched the VISITOR labels in the Gold dataset. Reason for this is that the parser handles the non-projective linguistic phenomena in native mode. We did not implement the conversion to the projective configuration. Occurrences: 23.
7. The parser recognizes the PREDCOMPL relation but for some syntactic structures of the sentence it fails to recognize the referent of the predication. Occurrences: 20.
8. The parser assigned the SEPARATOR label instead of both OPEN + PARENTHETICAL and CLOSE+PARENTHETICAL. Possible cause is that currently the constraints we have defined for the lemma "," (comma) are too strict so that they make the parser choose in a few situation the SEPARATOR label instead of the PARENTHETICAL. Occurrences: 23.

The systematic errors listed above are the 58.63% of the total number of errors which make the difference between LAS and UAS. It is interesting to note that once the eight cases of error have been solved or accepted, the LAS value would be 94.12% and a related increment of CSA-L can be expected.

There is also an expected error related to the attachment to the modal verb. In those sentences where a given atom (e.g. modifier or conjunction) can be connected to either the VERB+MODAL-INDCOMPL or to its governor maintaining in both cases the syntactic coherence, in 32 cases the Gold dataset and the parser made different choices.

Furthermore we found that in the Gold dataset the seven instances of the sequence of words "salvo che" are annotated in a different way with respect to TUT Treebank that labels them as "LOCUTION". Since our grammar follows the one of the TUT, this fact caused 14 errors (the discrepancy in the annotation causes actually two errors for every occurrence). We trust the annotation of the TUT so we do not consider those be real errors. Other discrepancies have been found between the Gold dataset and the TUT treebank that in agreement with Bosco and Lavelli [3] they have not been taken into account.

References

1. Attardi, G.: Experiments with a multilanguage non-projective dependency parser. In: Proceedings of the Tenth Conference on Computational Natural Language Learning. pp. 166–170. CoNLL-X '06, Association for Computational Linguistics, Stroudsburg, PA, USA (2006)
2. Barbosa, P., Fox, D., Hagstrom, P., McGinnis, M., Pesetsky, D. (eds.): Is the Best Good Enough? Optimality and Competition in Syntax. MIT Press, Cambridge MA (1998)
3. Bosco, C., Lavelli, A.: Annotation schema oriented evaluation for dependency parsing validation. In: Proceedings of the ninth international workshop on Treebanks and Linguistic Theories (TLT 9). Tartu (2010)
4. Bosco, C., Montemagni, S., Mazzei, A., Lombardo, V., DellOrletta, F., Lenci, A.: Evalita09 parsing task: comparing dependency parsers and treebanks. In: Proceedings of Evalita09, Reggio Emilia (2009), `http://www.evalita.it`

5. Debusmann, R., Duchier, D., Kruijff, G.J.M.: Extensible dependency grammar: A new methodology. In: Proceedings of the COLING 2004 Workshop on Recent Advances in Dependency Grammar. Geneva/SUI (2004)
6. Duchier, D.: Axiomatizing dependency parsing using set constraints. In: 6th Meeting on Mathematics of Language. Orlando/FL (1999)
7. Eisner, J.M.: Three new probabilistic models for dependency parsing: An exploration. In: Proceedings of the 16th International Conference on Computational Linguistics (COLING-96. pp. 340–345 (1997)
8. Harper, M.P., Helzerman, R.A., Helzerman, A.A.: Extensions to constraint dependency parsing for spoken language processing. Computer Speech and Language 9, 187–234 (1995)
9. Lesmo, L.: The Turin University parser at Evalita 2009. In: Proceedings of Evalita09, Reggio Emilia (2009)
10. Lesmo, L., Lombardo, V., Bosco, C.: Treebank development: the TUT approach. In: Proceedings of the International Conference on Natural Language Processing. pp. 61–70. Vikas Publishing House (2002), `http://www.di.unito.it/~tutreeb`
11. Maruyama, H.: Structural disambiguation with constraint propagation. In: Proceedings of the 28th conference on Association for Computational Linguistics. pp. 31–38. Association for Computational Linguistics (1990)
12. McCord, M.C.: Slot grammars. Computational Linguistics 6, 31–43 (Jan 1980)
13. Mcdonald, R., Pereira, F.: Online Learning of Approximate Dependency Parsing Algorithms
14. Menzel, W., Schrder, I.: Decision procedures for dependency parsing using graded constraints. In: Proceedings of ACL'90. pp. 78–87 (1998)
15. Nivre, J.: An efficient algorithm for projective dependency parsing. In: Proceedings of the 8th International Workshop on Parsing Technologies (IWPT). pp. 149–160 (2003)
16. Robaldo, L., Caselli, T., Grella, M.: Rule-based creation of TimeML documents from Dependency Trees. In: AI*IA'11. pp. 389–394 (2011)
17. Rossi, F., Beek, P.V., Walsh, T.: Handbook of Constraint Programming (Foundations of Artificial Intelligence). Elsevier Science Inc., New York, NY, USA (2006)
18. Schulte, C., Tack, G., Lagerkvist, M.Z.: Gecode: A generic constraint development environment (Nov 2006), `http://www.gecode.org/`
19. Taft, S.T., Duff, R.A., Brukardt, R.L., Ploedereder, E., Leroy, P.: Ada 2005 Reference Manual. Language and Standard Libraries: International Standard ISO/IEC 8652/1995(E) with Technical Corrigendum 1 and Amendment 1 (Lecture Notes in Computer Science). Springer-Verlag New York, Inc., Secaucus, NJ, USA (2007)
20. Tsang, E.P.K.: Foundations of constraint satisfaction. Computation in cognitive science, Academic Press (1993)
21. Wikipedia: Wikipedia, l'enciclopedia libera (2011), `http://it.wikipedia.org`
22. Yamada, H., Matsumoto, Y.: Statistical Dependency Analysis with Support Vector Machines. In: Proceedings of IWPT. pp. 195–206 (2003)