Combining Rule-Based and Statistical Syntactic Analyzers

**María Jesús Aranzabe[*], Arantza Díaz de Ilarraza, Nerea Ezeiza, Kepa Bengoetxea,
Iakes Goenaga, Koldo Gojenola,**
Department of Computer Languages and Systems / [*] Department of Basque Philology
University of the Basque Country UPV/EHU
{maxux.aranzabe@ehu.es, kepa.bengoetxea, jipdisaa@si.ehu.es,
n.ezeiza@ehu.es, koldo.gojenola@ehu.es, iakes@gmail.com}

Abstract

This paper presents the results of a set of preliminary experiments combining two knowledge-based partial dependency analyzers with two statistical parsers, applied to the Basque Dependency Treebank. The general idea will be to apply a stacked scheme where the output of the rule-based partial parsers will be given as input to MaltParser and MST, two state of the art statistical parsers. The results show a modest improvement over the baseline, although they also present interesting lines for further research.

1. Introduction

In this paper we present a set of preliminary experiments on the combination of two knowledge-based partial syntactic analyzers with two state of the art data-driven statistical parsers. The experiments have been performed on the Basque Dependency Treebank (Aduriz et al., 2003).

In the last years, many attempts have been performed trying to combine different parsers (Surdeanu and Manning, 2010), with significant improvements over the best individual parser's baseline. The two most successful approaches have been stacking (Martins et al., 2008) and voting (Sagae and Lavie, 2006, Nivre and McDonald, 2008, McDonald and Nivre, 2011). In this paper we will experiment the use of the stacking technique, giving the tags obtained by the rule-based syntactic partial parsers as input to the statistical parsers.

Morphologically rich languages present new challenges, as the use of state of the art parsers for more configurational and non-inflected languages like English does not reach similar performance levels in languages like Basque, Greek or Turkish (Nivre et al., 2007a). As it was successfully done on part of speech (POS) tagging, where the use of rule-based POS taggers (Tapanainen and Voutilainen, 1994) or a combination of a rule-based POS tagger with a statistical one (Aduriz et al., 1997, Ezeiza et al., 1998) outperformed purely statistical taggers, we think that exploring the combination of knowledge-based and data-driven systems in syntactic processing can be an interesting line of research.

Most of the experiments on combined parsers have relied on different types of statistical parsers (Sagae and Lavie, 2006, Martins et al., 2008, McDonald and Nivre, 2011), trained on an automatically annotated treebank. Yeh (2000) used the output of several baseline diverse parsers to increase the performance of a second transformation-based parser. In our work we will study the use of two partial rule-based syntactic analyzers together with two data-driven parsers:

- A rule-based chunker (Aduriz et al., 2004) that marks the beginning and end of noun phrases, postpositional phrases and verb chains, in the IOB (Inside/ Outside/Beginning of a chunk) style.

- A shallow dependency relation annotator (Aranzabe et al., 2004), which tries to detect dependency relations by assigning a
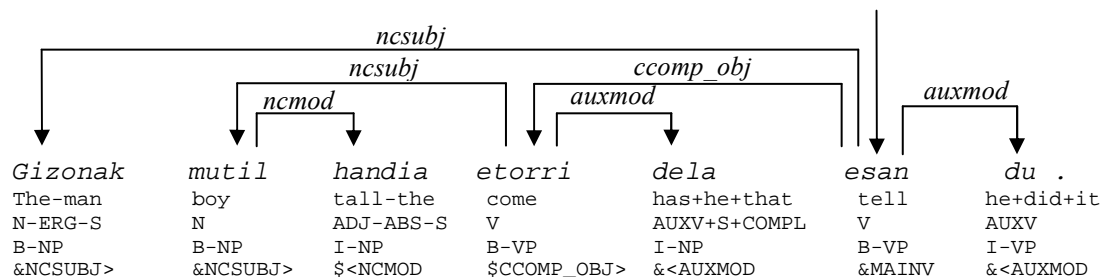
Figure 1. Dependency tree for the sentence *Gizonak mutil handia etorri dela esan du* (the man told that the tall boy has come). The two last lines show the tags assigned by the rule-based chunker and the rule-based dependency analyzer, respectively.
(V = main verb, N = noun, AUXV = auxiliary verb, COMPL = completive, ccomp_obj = clausal complement object, ERG = ergative, S: singular, auxmod = auxiliary, ncsubj = non-clausal subject, B-NP = beginning of NP, I-NP = inside an NP, &MAINV = main verb, &<AUXMOD = verbal auxiliary modifier).

set of predefined tags to each word, where each tag gives both the name of a dependency relation (e.g. subject) together with the direction of its head (left or right).

- We will use two statistical dependency parsers, MaltParser (Nivre et al., 2007b) and MST (McDonald et al, 2005).

In the rest of this paper, section 2 will first present the corpus and the different parsers we will combine, followed by the experimental results in section 3, and the main conclusions of the work.

2. Resources

This section will describe the main resources that have been used in the experiments. First, subsection 2.1 will describe the Basque Dependency Treebank, and then subsection 2.2 will explain the main details of the analyzers that have been employed. The analyzers are a rule-based chunker, a rule-based shallow dependency parser and two state of the art data-driven dependency parsers, MaltParser and MST.

2.1 Corpora

Our work will make use the second version of the Basque dependency Treebank (BDT II, Aduriz et al., 2003), containing 150,000 tokens (11,225 sentences). Figure 1 presents an example of a syntactically annotated sentence. Each word contains its form, lemma, category or coarse part of speech (CPOS), POS, morphosyntactic features such as case, number of subordinate relations, and the dependency relation (headword + dependency). The information in figure 1 has been simplified due to space reasons, as typically each word

contains many morphosyntactic features (case, number, type of subordinated sentence, ...), which are relevant for parsing. The last two lines of the sentence in figure 1 do not properly correspond to the treebank, but are the result of the rule-based partial syntactic analyzers (see subsection 2.2). For evaluation, we divided the treebank in three sets, corresponding to training, development, and test (80%, 10%, and 10%, respectively). The experiments were performed on the development set, leaving the best system for the final test.

2.2 Analyzers

This subsection will present the four types of analyzers that have been used. The rule-based analyzers are based on the Contraint Grammar (CG) formalism (Karlsson et al., 1995), based on the assignment of morphosyntactic tags to words using a formalism that has the capabilities of finite state automata or regular expressions, by means of a set of rules that examine mainly local contexts of words to determine the correct tag assignment.

The rule-based chunker (RBC henceforth, Aranzabe et al., 2009) uses 560 rules, where 479 of the rules deal with noun phrases and the rest with verb phrases. The chunker delimits the chunks with three tags, using a standard IOB marking style (see figure 1). The first one is to mark the beginning of the phrase (B-VP if it is a verb phrase and B-NP whether it's a noun phrase) and the other one to mark the continuation of the phrase (I-NP or I-VP, meaning that the word is inside an NP or VP). The last tag marks words that are outside a chunk. The evaluation of the chunker on the BDT gave a result of 87% precision and 85% recall over all chunks. We must take into account that this evaluation was

performed on the gold POS tags, rather than on automatically assigned POS tasks, as in the present experiment. For that reason, the results can serve as an upper bound on the real results.

The rule-based dependency analyzer (RBDA, Aranzabe et al., 2004) uses a set of 505 CG rules that try to assign dependency relations to wordforms. As the CG formalism only allows the assignment of tags, the rules only aim at marking the name of the dependency relation together with the direction of the head (left or right). For example, this analyzer assigns tags of the form &NCSUBJ> (see figure 1), meaning that the corresponding wordform is a non-clausal syntactic subject and that its head is situated to its right (the ">" or "<" symbols mark the direction of the head). This means that the result of this analysis is on the one hand a partial analysis and, on the other hand, it does not define a dependency tree, and can also be seen as a set of constraints on the shape of the tree. The system was evaluated on the BDT, obtaining f-scores between 90% for the *auxmod* dependency relation between the auxiliary and the main verb and 52% for the *subject* dependency relation, giving a (macro) average of 65%.

Regarding the data-driven parsers, we have made use of MaltParser (Nivre et al., 2007b) and MST Parser (McDonald et al., 2006), two state of the art dependency parsers representing two dominant approaches in data-driven dependency parsing, and that have been successfully applied to typologically different languages and treebanks (McDonald and Nivre, 2007).

MaltParser (Nivre, 2006) is a representative of local, greedy, transition-based dependency parsing models, where the parser obtains deterministically a dependency tree in a single pass over the input using two data structures: a stack of partially analyzed items and the remaining input sequence. To determine the best action at each step, the parser uses history-based feature models and discriminative machine learning. The learning configuration can include any kind of information (such as word-form, lemma, category, subcategory or morphological features). Several variants of the parser have been implemented, and we will use one of its standard versions (MaltParser version 1.4). In our experiments, we will use the Stack-Lazy algorithm with the *liblinear* classifier.

The MST Parser can be considered a representative of global, exhaustive graph-based parsing (McDonald et al., 2005, 2006). This algorithm finds the highest scoring directed spanning tree in a dependency graph forming a valid dependency tree. To learn arc scores, it uses large-margin structured learning algorithms, which optimize the parameters of the model to maximize the score margin between the correct dependency graph and all incorrect dependency graphs for every sentence in a training set. The learning procedure is global since model parameters are set relative to classifying the entire dependency graph, and not just over single arc attachments. This is in contrast to the local but richer contexts used by transition-based parsers. We use the freely available version of MSTParser[1]. In the following experiments we will make use of the second order non-projective algorithm.

3. Experiments

We will experiment the effect of using the output of the knowledge-based analyzers as input to the data-driven parsers in a stacked learning scheme. Figure 1 shows how the two last lines of the example sentence contain the tags assigned by the rule-based chunker (B-NP, I-NP, B-VP and I-VP) and the rule-based partial dependency analyzer (&NCSUBJ, &<NCMOD, &<AUXMOD, &CCOMP_OBJ and &MAINV) .

The first step consisted in applying the complete set of text processing tools for Basque, including:

- Morphological analysis. In Basque, each word can receive multiple affixes, as each lemma can generate thousands of word-forms by means of morphological properties, such as case, number, tense, or different types of subordination for verbs. Consequently, the morphological analyzer for Basque (Aduriz et al. 2000) gives a high ambiguity. If only categorial (POS) ambiguity is taken into account, there is an average of 1.55 interpretations per word-form, which rises to 2.65 when the full morphosyntactic information is taken into account, giving an overall 64% of ambiguous word-forms.
- Morphological disambiguation. Disambiguating the output of morphological analysis, in order to obtain a single interpretation for each word-form,

	MaltParser		MST Parser	
	LAS	UAS	LAS	UAS
Baseline	76.77%	82.09%	77.96%	84.04%
+ RBC	77.10% (+0.33)	82.29% (+0.20)	77.99% (+0.03)	83.99% (-0.05)
+ RBDA	*77.15% (+0.38)	82.27% (+0.18)	78.03% (+0.07)	83.76% (-0.28)
+ RBC + RBDA	*77.25% (+0.48)	82.18% (+0.09)	78.00% (+0.04)	83.34% (-0.70)

Table 1. Evaluation results

(RBC = rule-based chunker, RBDA = rule-based dependency analyzer, LAS: Labeled Attachment Score,

UAS: Unlabeled Attachment Score, *: statistically significant in McNemar's test, $p < 0.05$)

can pose an important problem, as determining the correct interpretation for each word-form requires in many cases the inspection of local contexts, and in some others, as the agreement of verbs with subject, object or indirect object, it could also suppose the examination of elements which can be far from each other, added to the free constituent order of the main sentence elements in Basque. The erroneous assignment of incorrect part of speech or morphological features can difficult the work of the parser.

- Chunker
- Partial dependency analyzer

When performing this task, we found the problem of matching the treebank tokens with those obtained from the analyzers, as there were divergences on the treatment of multiword units, mostly coming from Named Entities, verb compounds and complex postpositions (formed with morphemes appearing at two different words). For that reason, we performed a matching process trying to link the multiword units given by the morphological analysis module and the treebank, obtaining a correct match for 99% of the sentences.

Regarding the data-driven parsers, they are trained using two kinds of tags as input:

- POS and morphosyntactic tags coming from the automatic morphological processing of the dependency treebank. Disambiguation errors, such as an incorrect POS category or morphological analyses (e.g. the assignment of an incorrect case) can harm the parser, as tested in Bengoetxea et al. (2011).

- The output of the rule-based partial syntactic analyzers (two last lines of the example in figure 1). These tags contain errors of the CG-based syntactic taggers. As the analyzers are applied after

morphological processing, the errors can be propagated and augmented.

Table 1 shows the results of using the output of the knowledge-based analyzers as input to the statistical parsers. We have performed three experiments for each statistical parser, trying with the chunks provided by the chunker, the partial dependency parser, and both. The table shows modest gains, suggesting that the rule-based analyzers help the statistical ones, giving slight increases over the baseline, which are statistically significant when applying MaltParser to the output of the rule-based dependency parser and a combination of the chunker and rule-based parsers. As table 1 shows, the parser type is relevant, as MaltParser seems to be sensitive when using the stacked features, while the partial parsers do not seem to give any significant improvement to MST.

3.1 Error analysis

Looking with more detail at the errors made by the different versions of the parsers, we observe significant differences in the results for different dependency relations, seeing that the statistical parsers behave in a different manner regarding to each relation, as shown in table 2. The table shows the differences in f-score[2] corresponding to five local dependency relations, (determination of verbal modifiers, such as subject, object and indirect object).

McDonald and Nivre (2007) examined the types of errors made by the two data-driven parsers used in this work, showing how the greedy algorithm of MaltParser performed better with local dependency relations, while the graph-based algorithm of MST was more accurate for global relations. As both the chunker and the partial dependency analyzer are based on a set of local rules in the CG formalism, we could expect that the stacked parsers could benefit mostly on the local dependency relations.

[2] f-score = 2 * precision * recall / (precision + recall)

Dependency relation	MaltParser				MST Parser			
	Baseline	+ RBC	+ RBDA	+ RBC + RBDA	Baseline	+ RBC	+ RBDA	+ RBC + RBDA
ncmod	75,29	75,90	76,08	**76,40**	77,15	**77,44**	76,39	76,92
ncobj	67,34	68,49	**69,67**	69,54	64,85	64,86	65,56	**66,18**
ncpred	61,37	61,92	61,26	**63,50**	**60,37**	57,55	58,44	59,27
ncsubj	61,92	61,90	**63,96**	63,91	59,19	59,26	62,23	**61,61**
nciobj	75,76	76,53	**77,16**	76,29	74,23	**74,47**	72,16	69,08

Table 2. Comparison of the different parsers' f-score with regard to specific dependency relations
(ncmod = non-clausal modifier, ncobj = non-clausal object, ncpred = non-clausal predicate, ncsubj = non-clausal subject, nciobj = non-clausal indirect object)

Table 2 shows how the addition of the rule-based parsers' tags performs in accord with this behavior, as MaltParser gets f-score improvements for the local relations. Although not shown in Table 2, we also inspected the results on the long distance relations, where we did not observe noticeable improvements with respect to the baseline on any parser. For that reason, MaltParser, seems to mostly benefit of the local nature of the stacked features, while MST does not get a significant improvement, except for some local dependency relations, such as ncobj and ncsubj.

We performed an additional test using the partial dependency analyzer's gold dependency relations as input to MaltParser. As could be expected, the gold tags gave a noticeable improvement to the parser's results, reaching 95% LAS. However, when examining the scores for the output dependency relations, we noticed that the gold partial dependency tags are beneficial for some relations, although negative for some others. For example the non-clausal modifier (ncmod) relation's f-score increases 3.25 points, while the dependency relation for clausal subordinate sentences functioning as indirect object decreases 0.46 points, which is surprising in principle.

For all those reasons, the relation between the input dependency tags and the obtained results seems to be intricate, and we think that it deserves new experiments in order to determine their nature. As each type of syntactic information can have an important influence on the results on specific relations, their study can shed light on novel schemes of parser combination.

4. Conclusions

We have presented a preliminary effort to integrate different syntactic analyzers, with the objective of getting the best from each system. Although the potential gain is in theory high, the experiments have shown very modest improvements, which seem to happen in the set of local dependency relations. We can point out some avenues for further research:

- Development of the rule-based dependency parser using the dependencies that give better improvements on the gold dependency tags, as this can measure the impact of each kind of shallow dependency tag on the data-driven parsers.

- Development of rules that deal with the phenomena where the statistical parsers perform worse. This requires a careful error analysis followed by a redesign of the manually developed CG tagging rules.

- Application of other types of combining schemes, such as voting, trying to get the best from each type of parser.

Finally, we must also take into account that the rule-based analyzers were developed mainly having linguistic principles in mind, such as coverage of diverse linguistic phenomena or the treatment of specific syntactic constructions (Aranzabe et al., 2004), instead of performance-oriented measures, such as precision and recall. This means that there is room for improvement in the first-stage knowledge-based parsers, which will have, at least in theory, a positive effect on the second-phase statistical parsers, allowing us to test whether knowledge-based and machine learning-based systems can be successfully combined.

Acknowledgements

References

Itziar Aduriz, José María Arriola, Xabier Artola, Arantza Díaz de Ilarraza, Koldo Gojenola and Montse Maritxalar. 1997. Morphosyntactic disambiguation for Basque based on the Constraint Grammar Formalism. Conference on *Recent Advances in Natural Language Processing* (RANLP), Bulgaria.

Itziar Aduriz, Eneko Agirre, Izaskun Aldezabal, Iñaki Alegria, Xabier Arregi, Jose Mari Arriola, Xabier Artola, Koldo Gojenola, Montserrat Maritxalar, Kepa Sarasola, and Miriam Urkia. 2000. A word-grammar based morphological analyzer for agglutinative languages. *Coling 2000*, Saarbrucken.

Itziar Aduriz, José María Arriola, Arantza Díaz de Ilarraza, Koldo Gojenola, Maite Oronoz and Larraitz Uria. 2004. A cascaded syntactic analyser for Basque. In *Computational Linguistics and Intelligent Text Processing*, pages 124-135. LNCS Series. Springer Verlag. Berlin. 2004

Itziar Aduriz, Maria Jesus Aranzabe, Jose Maria Arriola, Aitziber Atutxa, Arantza Diaz de Ilarraza, Aitzpea Garmendia and Maite Oronoz. 2003. Construction of a Basque dependency treebank. *Treebanks and Linguistic Theories*.

María Jesús Aranzabe, José María Arriola and Arantza Díaz de Ilarraza. 2004. Towards a Dependency Parser for Basque. In *Proceedings of the Workshop on Recent Advances in Dependency Grammar*, Geneva, Switzerland.

Maria Jesus Aranzabe, Jose Maria Arriola and Arantza Díaz de Ilarraza. 2009. Theoretical and Methodological issues of tagging Noun Phrase Structures following Dependency Grammar Formalism. In Artiagoitia, X. and Lakarra J.A. (eds) Gramatika Jaietan. Patxi Goenagaren omenez. Donostia: Gipuzkoako Foru Aldundia-UPV/EHU.

Kepa Bengoetxea and Koldo Gojenola. 2010. Application of Different Techniques to Dependency Parsing of Basque. Proceedings of the *1st Workshop on Statistical Parsing of Morphologically Rich Languages* (SPMRL), NAACL-HLT Workshop.

Kepa Bengoetxea, Arantza Casillas and Koldo Gojenola. 2011. Testing the Effect of Morphological Disambiguation in Dependency Parsing of Basque. Proceedings of the *International Conference on Parsing Technologies (IWPT). 2nd Workshop on Statistical Parsing Morphologically Rich Languages (SPMRL)*, Dublin, Ireland.

Gülsen Eryiğit, Joakim Nivre and Kemal Oflazer. 2008. Dependency Parsing of Turkish. *Computational Linguistics*, Vol. 34 (3).

Nerea Ezeiza, Iñaki Alegria, José María Arriola, Rubén Urizar and Itziar Aduriz. 1998. Combining Stochastic and Rule-Based Methods for Disambiguation in Agglutinative Languages. *COLING-ACL'98*, Montreal.

Fred Karlsson, Atro Voutilainen, Juka Heikkila and Arto Anttila. 1995. *Constraint Grammar: A Language-independent System for Parsing Unrestricted Text*. Mouton de Gruyter.

André F. T. Martins, Dipanjan Das, Noah A. Smith and Eric P. Xing. 2008. Stacking Dependency Parsing. *Proceedings of EMNLP-2008*.

Ryan McDonald, Kevin Lerman and Fernando Pereira. 2005. Online large-margin training of dependency parsers. In *Proceedings of ACL*.

Ryan McDonald, Kevin Lerman and Fernando Pereira. 2006. Multilingual dependency analysis with a two-stage discriminative parser. In Proc. CoNLL.

Ryan McDonald and Joakim Nivre. 2007. Characterizing the Errors of Data-Driven Dependency Parsing Models. Proceedings of the 2007 *Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP/CoNLL.

Ryan McDonald and Joakim Nivre. 2011. Analyzing and Integrating Dependency Parsers. *Computational Linguistics,* Vol. 37(1), 197-230.

Joakim Nivre. 2006. *Inductive Dependency Parsing*. Springer.

Joakim Nivre, Johan Hall, Sandra Kübler, Ryan McDonald, Jens Nilsson, Sebastian Riedel and Deniz Yuret. 2007a. The CoNLL 2007 Shared Task on Dependency Parsing. *Proceedings of EMNLP-CoNLL*.

Joakim Nivre, Johan Hall, Jens Nilsson, Atanas Chanev, Gülsen Eryiğit, Sandra Kübler, S. Marinov and Edwin Marsi. 2007b. MaltParser: A language-independent system for data-driven dependency parsing. *Natural Language Engineering*.

Joakim Nivre and Ryan McDonald. 2008. Integrating graph-based and transition-based dependency parsers. *Proceedings* of ACL-2008.

Kenji Sagae and Alon Lavie. 2006. Parser Combination by Reparsing. *Proceedings of the Human Language*

Technology Conference of the North American Chapter of the ACL, pages 129–132, New York.

Mihai Surdeanu and Christopher D. Manning. 2010. Ensemble Models for Dependency Parsing: Cheap and Good? *Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL.*

Pasi Tapanainen and Atro Voutilainen. 1994. Tagging Accurately-Don´t guess if you know. *Proceedings of the Conference on Applied Natural Language Processing*, ANLP´94.

Alexander Yeh. 2000. Using existing systems to supplement small amounts of annotated grammatical relations training data. *Proceedings of ACL* 2000.