

# Application of feature propagation to dependency parsing

**Kepa Bengoetxea**

**Koldo Gojenola**

IXA NLP Group, Technical School of Engineering, Bilbao  
University of the Basque Country, Plaza La Casilla 3, 48012, Bilbao  
kepa.bengoetxea@ehu.es, koldo.gojenola@ehu.es

## Abstract

This paper presents a set of experiments performed on parsing the Basque Dependency Treebank. We have applied feature propagation to dependency parsing, experimenting the propagation of several morphosyntactic feature values. In the experiments we have used the output of a parser to enrich the input of a second parser. Both parsers have been generated by Maltparser, a freely data-driven dependency parser generator. The transformations, combined with the pseudoprojective graph transformation, obtain a LAS of 77.12% improving the best reported results for Basque.

## 1 Introduction

This work presents several experiments performed on dependency parsing of the Basque Dependency Treebank (BDT, Aduriz et al. 2003). We have experimented the idea of feature propagation through dependency arcs, in order to help the parser. Feature propagation has been used in classical unification-based grammars as a means of propagating linguistic information through syntax trees. We apply this idea in the context of inductive dependency parsing, combining a reduced set of linguistic principles that express feature propagation among linguistic elements with Maltparser (Nivre et al. 2007a), an automatic dependency parser generator.

We have concentrated on propagating several morphosyntactic feature values from: a) auxiliary verbs to the main verb, b) the last constituent to the head noun, in noun phrases c) the last conjunct to the conjunction, in coordination.

This work was developed in the context of dependency parsing exemplified by the CoNLL shared task on dependency parsing in years 2006 and 2007 (Nivre et al. 2007b), where several systems had to compete analyzing data from a typologically varied range of 11 languages. The treebanks for all languages were standardized using a previously agreed CoNLL-X format (see Figure 1). BDT was one of the evaluated treebanks, which will allow a direct comparison of results.

Many works on treebank parsing have dedicated an effort to the task of pre-processing training trees (Nilsson et al. 2007). When these approaches have been applied to dependency parsing several works (Nilsson et al. 2007; Bengoetxea and Gojenola 2009) have concentrated on modifying the structure of the dependency tree, changing the shape of the graph. In contrast, rather than modifying the tree structure, we will experiment changing the information contained in the nodes of the tree. This approach requires having an initial dependency tree in order to apply the feature propagation process, which will be obtained by means of a standard trained model. This way, the features will be propagated through some incorrect dependency arcs, and the process will be dependent on the reliability of the initial arcs. After enriching the tree, a second parsing model will try to use this new information to improve the standard model. This process can also be seen as an example of stacked learning (Martins et al. 2008, Nivre and McDonald 2008) where a second parser is used to improve the performance of a first one.

The rest of the paper is organized as follows. Section 2 presents the main resources used in this work. Section 3 presents three different proposals for the propagation of the most important morphological features. Next, section 4 will evaluate the results of each transformation, and the last section outlines the main conclusions.

## 2 Resources

This section will describe the main elements that have been used in the experiments. First, subsection 2.1 will present the Basque Dependency Treebank data, while subsection 2.2 will describe the main characteristics of Maltparser, a state of the art and data-driven dependency parser.

### 2.1 The Basque Dependency Treebank

The BDT can be considered a pure dependency treebank, as its initial design considered that all the dependency arcs would connect sentence tokens. Although this decision had consequences on the annotation process, its simplicity is also an advantage when applying several of the most

| Index | Word    | Lemma  | Category | Subcategory | Features       | Head | Dependency |
|-------|---------|--------|----------|-------------|----------------|------|------------|
| 1     | etorri  | etorri | V        | V           | —              | 3    | lot        |
| 2     | dela    | izan   | AUXV     | AUXV        | SC:CMP SUBJ:3S | 1    | auxmod     |
| 3     | eta     | eta    | CONJ     | CONJ        | —              | 6    | ccomp_obj  |
| 4     | joan    | joan   | V        | V           | —              | 3    | lot        |
| 5     | dela    | izan   | AUXV     | AUXV        | SC:CMP SUBJ:3S | 4    | auxmod     |
| 6     | esan    | esan   | V        | V           | —              | 0    | ROOT       |
| 7     | zien    | *edun  | AUXV     | AUXV        | SUBJ:3S OBJ:3P | 6    | auxmod     |
| 8     | mutil   | mutil  | NOUN     | NOUN        | —              | 6    | ncsubj     |
| 9     | txikiak | txiki  | ADJ      | ADJ         | CASE:ERG NUM:S | 8    | ncmod      |
| 10    | .       | .      | PUNT     | PUNT_PUNT   | —              | 9    | PUNC       |

Figure 1: Example of a BDT sentence in the CONLL-X format

(V = main verb, AUXV = auxiliary verb, SC = subordinated clause, CMP = completive, ccomp\_obj = clausal complement object, SUBJ:3S: subject in 3rd person sing., OBJ:3P: object in 3rd person pl.).

efficient parsing algorithms. The treebank consists of 55,469 tokens forming 3,700 sentences, 334 of which were used as test data.

```
(1) Etorri (come) dela (that-has) eta
    (and) joan (go) dela (that-has) esan
    (tell) zien (did) mutil (boy)
    txikiak(the-little)
He told the little boy that he has come
and he has gone
```

Figure 1 contains an example of a sentence (1), annotated in the CoNLL-X format. The text is organized in eight tab-separated columns: word-number, form, lemma, category, subcategory, morphological features, and the dependency relation (headword + dependency). Basque is an agglutinative language and it presents a high power to generate inflected word-forms. The information in Figure 1 has been simplified due to space reasons, as typically the Features column will contain many morphological features, which are relevant for parsing.

## 2.2 Maltparser

Maltparser (Nivre et al. 2007a) is a state of the art dependency parser that has been successfully applied to typologically different languages and treebanks. While several variants of the base parser have been implemented, we will use one of its standard versions (Maltparser version 0.4).

The parser obtains deterministically a dependency tree in linear-time in a single pass over the input. To determine which is the best action at each parsing step, the parser uses history-based feature models and discriminative machine learning. In all the following experiments, we made use of a SVM classifier. The specification of the features used for learning can in principle be any kind of data in Figure 1 (such as word-form, lemma, category or morphological features).

## 3 Experiments

We applied the following steps:

- Application of feature propagation to the training data, using the gold standard arcs, obtaining a “enriched training data”.
- Training Maltparser on the “enriched training data” to generate a “enriched parser”.
- Training Maltparser with the training data, without any transformation, to generate a “standard parser”.
- Parse the test data with the “standard parser”, obtaining the “standard output”.
- Apply feature propagation to the “standard output”, using the dependency arcs given by the parser (with some incorrect arcs), obtaining the “standard parser’s enriched output”.
- Finally, parsing the “standard parser’s enriched output” with the “enriched parser”,

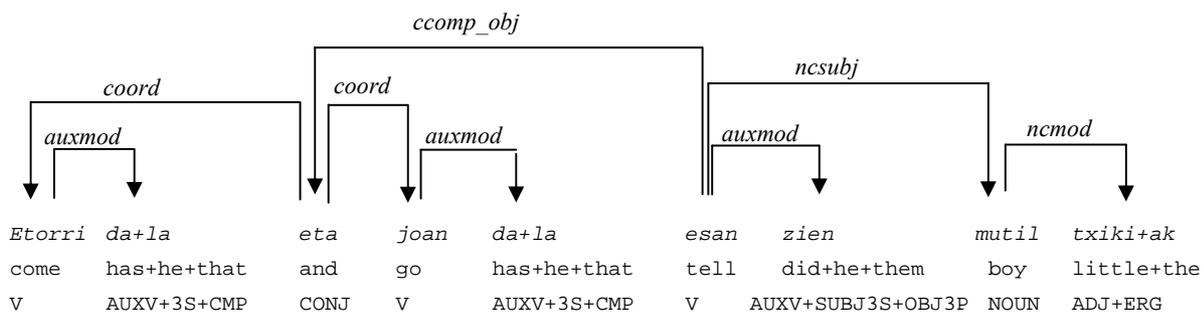


Figure 2: Dependency tree for the sentence in Figure 1.

(V = main verb; AUXV: auxiliary verb; CMP: completive subordinated mark; CONJ: conjunction; ERG: ergative case).

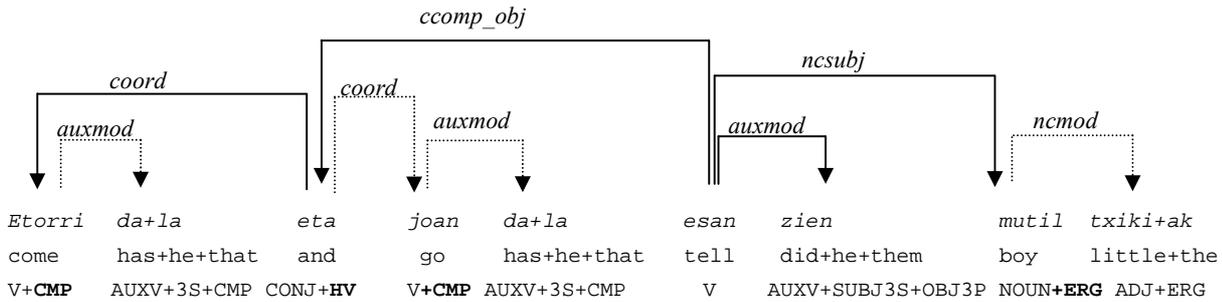


Figure 3: Dependency tree after propagating the morphological features.

evaluating the output with the gold test data.

We have applied three types of feature propagation of the most important morphological feature values: a) from auxiliary verbs to the main verb (verb phrases) b) from post-modifiers to the head noun (noun phrases) c) from the last conjunct to the conjunction (coordination). This was done because Basque is a head final language, where many relevant features are located at the end of constituents. Figure 3 shows (dotted lines) the arcs that will propagate features from child to parent. The three transformations will be described in the following subsections.

### 3.1 Verb compounds

In BDT the verbal elements are organized around the main verb, but much syntactically relevant verbal information, like subordination type, aspect, tense and agreement usually appear attached to the auxiliary verb, which is the dependent. Its main consequence for parsing is that the elements bearing the relevant information for parsing are situated far in the tree with respect to their head. In Figure 2, we can see that the morpheme *-la*, indicating a subordinated completive sentence, appears down in the tree, and this could affect the correct attachment of the two coordinated verbs to the conjunction (*eta*), as conjunctions should link elements showing similar grammatical features (*-la* in this example). Similarly, it could affect the decision about the dependency type of *eta* (and) with respect to the main verb *esan* (to say), as the dependency relation *ccomp\_obj* is defined by means of the *-la* (completive) morpheme, far down in the tree.

Figure 3 shows the effect of propagating the completive feature value (CMP) from the auxiliary verb to the main verb through the *auxmod* (auxiliary modifier) relation.

### 3.2 Noun Phrases

In noun phrases and postpositional phrases, the most important morphological feature values

(case and number) are situated in the last post-modifier after the noun. Figure 3 shows the effect of propagating the ergative (ERG) case feature value from the adjective (the last constituent of the noun phrase) to the noun through the relation *nmod* (non-clausal modifier).

### 3.3 Coordination

Coordination in BDT was annotated in the so called Prague Style, where the conjunction is taken as the head, and the conjuncts depend on it. Basque is head final, so usually the last conjunct contains syntactically relevant features. We experimented the promotion of the category, case and subordination information from the last conjunct to the conjunction. In the example in Figure 3, the conjunction (*eta*) receives a new feature (HV for Head:Verb) from its dependent. This can be seen as an alternative to (Nilsson et al. 2007) who transform dependency arcs.

## 4 Evaluation

Evaluation was performed dividing the treebank in three sets: training set (45,000 tokens), development and test sets (5,000 tokens each). Training and testing of the system have been performed on the same datasets presented at the CoNLL 2007 shared task, which will allow for a direct comparison. Table 1 presents the Labeled Attachment Score (LAS) of the different tests on development and test data. The first row presents the best system score (76.94% LAS) in CoNLL 2007. This system combined six variants of a base parser (Maltparser). The second row shows the single Maltparser approach which obtained the fifth position. Row 3 presents Bengoetxea and Gojenola's results (76.80% LAS) when applying graph transformations (pseudo-projective, coordination and verb groups) to Basque, in the spirit of Nilsson et al. (2007). Row 4 shows our results after applying several feature optimizations, which we will use as our baseline.

|    | System  | LAS            |                 |
|----|---|----------------|-----------------|
|    |   | Development    | Test            |
| 1  | Nivre et al. 2007b (CoNLL 2007)                               | -              | 76.94%          |
| 2  | Hall et al. 2007 (CoNLL 2007)                                 |                | 74.99%          |
| 3  | Bengoetxea and Gojenola 2009                                  |                | 76.80%          |
| 4  | Feature optimization (baseline)                               | 77.46%         | 75.07%          |
| 5  | Proj  | 78.16% (+0.70) | *75.99% (+0.92) |
| 6  | P <sub>VG</sub>   | 78.14% (+0.68) | 75.54% (+0.47)  |
| 7  | P <sub>COOR</sub>   | 77.36% (-0.10) | 75.22% (+0.15)  |
| 8  | P <sub>CAS</sub>  | 77.32% (-0.14) | 74.86% (-0.21)  |
| 9  | P <sub>VG</sub> + P <sub>CAS</sub>                            | 78.53% (+1.09) | 75.42% (+0.35)  |
| 10 | P <sub>COOR</sub> + P <sub>VG</sub> + P <sub>CAS</sub>        | 78.31% (+0.85) | *75.93% (+0.86) |
| 11 | P <sub>COOR</sub> + P <sub>VG</sub>                           | 78.25% (+0.79) | *75.93% (+0.86) |
| 12 | Proj + P <sub>VG</sub>  | 78.91% (+1.45) | *76.12% (+1.05) |
| 13 | Proj + P <sub>VG</sub> + P <sub>COOR</sub> + P <sub>CAS</sub> | 78.31% (+0.85) | *77.12% (+2.05) |

Table 1. Evaluation results

(Proj: Pseudo-projective, P<sub>VG</sub>, P<sub>CAS</sub>, P<sub>COOR</sub>: Propagation on verb compounds, case (NPs) and coordination; \*: statistically significant in McNemar's test with respect to labeled attachment score with  $p < 0.01$ )

Feature propagation in verb groups (P<sub>VG</sub>) improves LAS in almost 0.5% (row 6 in Table 1). While coordination and case propagation do not improve significantly the accuracy by themselves (rows 7 and 8), their combination with P<sub>VG</sub> (verb groups) significantly increases LAS (+0.86%, see row 10). Looking at the accuracy of the dependency arcs used for feature propagation, auxiliary verbs are the most reliable elements, as their arcs (linking it to its head, the main verb) have 97% precision and 98% recall. This is in accord with P<sub>VG</sub> giving the biggest increase, while arcs related to coordination (63% precision and 65% recall) give a more modest contribution.

BDT contains 2.9% of nonprojective arcs, so we experimented the effect of combining the pseudoprojective transformation (Nilsson et al. 2007) with feature propagation, obtaining a LAS of 77.12%, the best reported results for the BDT.

## 5 Conclusions

We have performed a set of experiments using the output of a parser to enrich the input of a second parser, propagating the relevant morphological feature values through dependency arcs. The best system, after applying three types of feature propagation, obtains a 77.12% LAS (2.05% improvement over the baseline) on the test set, which is the best reported result for Basque dependency parsing, improving the better published result for a combined parser (76.94%).

## Acknowledgements

This research was supported by the Basque Government (EPEC-RS, S-PE08UN48) and the Uni-

versity of the Basque Country (EHU-EJIE, EJIE07/05).

## References

- I. Aduriz, M. J. Aranzabe, J. M. Arriola, A. Atutxa, A. Diaz de Ilarraza, A. Garmendia and M. Oronoz. 2003. *Construction of a Basque dependency treebank*. Treebanks and Linguistic Theories.
- Kepa Bengoetxea and Koldo Gojenola. 2009. *Exploring Treebank Transformations in Dependency Parsing*. Proceedings of RANLP'2009.
- Johan Hall, Jens Nilsson, Joakim Nivre J., Eryigit G., Megyesi B., Nilsson M. and Saers M. 2007. *Single Malt or Blended? A Study in Multilingual Parser Optimization*. Proceedings of the CoNLL Shared Task EMNLP-CoNLL.
- André F. T. Martins, Dipanjan Das, Noah A. Smith, Eric P. Xing. 2008. *Stacking Dependency Parsing*. EMNLP-2008.
- Jens Nilsson, Joakim Nivre and Johan Hall. 2007. *Tree Transformations for Inductive Dependency Parsing*. Proceedings of the 45th ACL.
- Joakim Nivre, Johan Hall, Jens Nilsson, Chaney A., Gülsen Eryigit, Sandra Kübler, Marinov S., and Edwin Marsi. 2007a. *MaltParser: A language-independent system for data-driven dependency parsing*. Natural Language Engineering.
- Joakim Nivre, Johan Hall, Sandra Kübler, Ryan McDonald, Jens Nilsson, Sebastian Riedel and Deniz Yuret. 2007b. *The CoNLL 2007 Shared Task on Dependency Parsing*. EMNLP-CoNLL.
- Joakim Nivre and Ryan McDonald. 2008. *Integrating graphbased and transition-based dependency parsers*. ACL-2008.