Exploring Treebank Transformations in Dependency Parsing

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

Abstract

This paper presents a set of experiments performed on parsing the Basque Dependency Treebank. We have concentrated on treebank transformations, maintaining the same basic parsing algorithm across the experiments. The experiments can be classified in two groups: 1) feature optimization, which is important mainly due to the fact that Basque is an agglutinative language, with a rich set of morphosyntactic features attached to each word, 2) graph transformations, ranging from language independent methods, such as projectivization, to language specific approaches, as coordination and subordinated sentences, where syntactic properties of Basque have been used to reshape the dependency trees used for training the system. The transformations have been tested independently and also in combination, showing that their order of application is relevant. The experiments were performed using a freely available state of the art data-driven dependency parser [11].

Keywords

Dependency parsing, treebank parsing, agglutinative language.

1 Introduction

This work presents several experiments performed on dependency parsing of the Basque Dependency Treebank (BDT) [1]. Several syntactic analyzers based on dependencies have been developed, with proposals ranging from systems that directly construct dependency structures [9] to other systems based on the more traditional constituency structures that allow the extraction of dependencies [2]. The present work has been developed in the context of dependency parsing exemplified by the CoNLL[1] shared task on dependency parsing in years 2006 and 2007 [12], 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 us to make a direct comparison of results.

Many works on treebank parsing have dedicated an effort to the task of pre-processing training trees [4, 13]. This paper extends these works, applying treebank transformations [7, 10] to a morphologically rich, agglutinative language.

The rest of the paper is organized as follows. Section 2 presents the main resources used in this work, including the BDT and a data-driven open source parser. Section 3 presents the different proposals for Treebank transformation that have been devised in order to improve the parser's accuracy. Next, section 4 will evaluate the results of each transformation. Section 5 examines related work, 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 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

BDT [2] 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 efficient parsing algorithms. The treebank consists of 55,469 tokens forming 3,700 sentences, 334 of which were used as test data[2].

```
(1) Etorri de-la eta joan de-la  esan  zien.
 come has-that and go has-that tell he-to-them
 He told them 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 (coarse POS), fine-grained POS, morphosyntactic features, and the dependency relation (headword + dependency). Basque is an agglutinative language, and it presents a high power to generate inflected word-forms.. Verbs offer a lot of grammatical information, as each verb form conveys information about the subject, the two objects, as well as the tense and aspect. As a result of this wealth of information contained within word-forms,

[1] *CoNLL: Computational Natural Language Learning*.

[2] The corpus is freely available. The treebank converted to the *CONLL-X* format can also be obtained from the authors.

International Conference RANLP 2009 - Borovets, Bulgaria, pages 33–38

W	Form	Lemma	CPOS	POS	Features	Head	Dependency
1	Etorri	etorri	V	V	_	3	coord
2	dela	izan	AUXV	AUXV	COMPL|3S	1	auxmod
3	eta	eta	CONJ	CONJ	_	6	ccomp_obj
4	joan	joan	V	V	_	3	coord
5	dela	izan	AUXV	AUXV	COMPL|3S	4	auxmod
6	esan	esan	V	V	_	0	ROOT
7	zien	*edun	AUXV	AUXV	SUBJ3S|OBJ3P	6	auxmod
8	.	.	PUNT	PUNT_PUNT	_	7	PUNC

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

(V = main verb, AUXV = auxiliary verb, COMPL = completive subordinate marker, ccomp_obj = clausal complement object, 3S: third person sing., SUBJ3S: subject in 3rd person sing., OBJ3P: object in 3rd person pl.).

complex structures have to be built to represent complete morphological information at word level. The information in Figure 1 has been simplified due to space reasons, as typically the Features column will contain lots of morphosyntactic features, which are relevant for parsing.

2.2 Maltparser

Maltparser [11] 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 is based on two basic data-structures. A stack stores the dependency-graph that is formed by linking the input sentence's words, while an input sequence contains the elements that have not yet been examined. The basic algorithm applies a set of four parsing actions (shift into the stack, reduce, left-arc, or right-arc) and obtains deterministically a dependency tree in linear-time in a single pass over the input. To determine which is the best action at each step, the parser uses history-based feature models and discriminative machine learning. In all the following experiments, we made use of a SVM[3] classifier. The specification of the features used by the classifier, allows to select the number of elements of both stack and input to be considered during learning, and also indicates the kind of information for each element, which can in principle be any kind of data described in Figure 1 (such as word-form, lemma, category or morphosyntactic features).

3 Experiments

We have performed two classes of experiments. First, we have tested the effect of simplifying morphosyntactic features. Second, we have applied three different tree transformations to the treebank.

3.1 Feature optimization

Basque is an agglutinative and morphologically rich language, and this opens the way to experiment with many combinations of morphological features. The original annotation of the BDT contained 359 different morphosyntactic feature values. This led us to experiment with several modifications:

- Grouping complex features into a set of simpler ones. For example, complex case suffixes were simplified, as in DAT_INS (a complex case suffix that is internally formed by the dative case followed by the instrumental case), which was changed to INS(trumental), as the last case suffix is syntactically more relevant.
- Deletion of several features that were interesting in the description of the internal morphology of a word but were not relevant for syntactic analysis.
- The original annotation of 359 values marked them as totally unrelated values, without indicating which feature (say, case) each value was an instance of. We added a label prefix to each value, which allowed us to experiment the inclusion of a feature. For example, ABS(olutive) was transformed to CASE:ABS.

After these steps, there were 127 values of morphosyntactic features, grouped in 14 features (case, number, tense, aspect, countable, …).

3.2 Graph transformations

Algorithms for dependency-tree transformations are applied in a black box manner in four steps: 1) apply the transformation to the training data, 2) train a parser on the transformed data, 3) parse the test set, and 4) apply the inverse transformation to the parse output, so that the final evaluation is carried over the original tree representations.

We will experiment with three different tree transformations, ranging from a language independent method in one extreme, like projectivization, to a pure language specific approach on the other, going through a transformation on coordinated structures, which lies in the middle, as coordination is present in all languages but needs an adaptation depending on each language and parser.

3.2.1 Projectivization (T_P)

Several parsing algorithms are unable to deal with non-projective arcs, that is, arcs that cross each other. The solution can be either to design a modified algorithm (e.g., Covington's, see [11]) or transform the tree into a projective one. This option is more attractive if the original

[3] We used SVM with a polinomial kernel of degree 2 (LIVSM parameters: -s 0 –t 1 –d 2 –g 0.2 –c 0.4 –r 0 –e 0.1 –S 0)

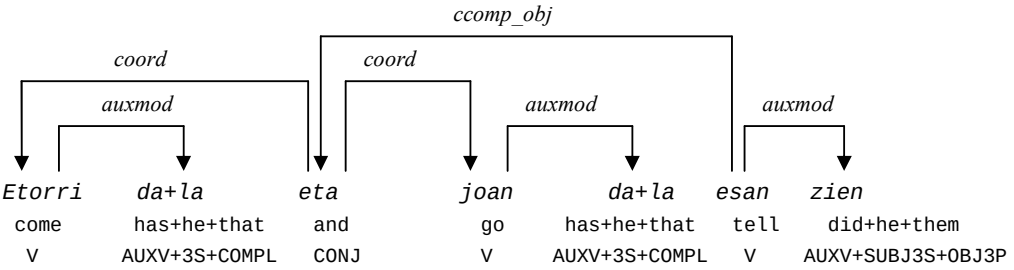

Figure 2: Dependency tree for the sentence in Figure 1,

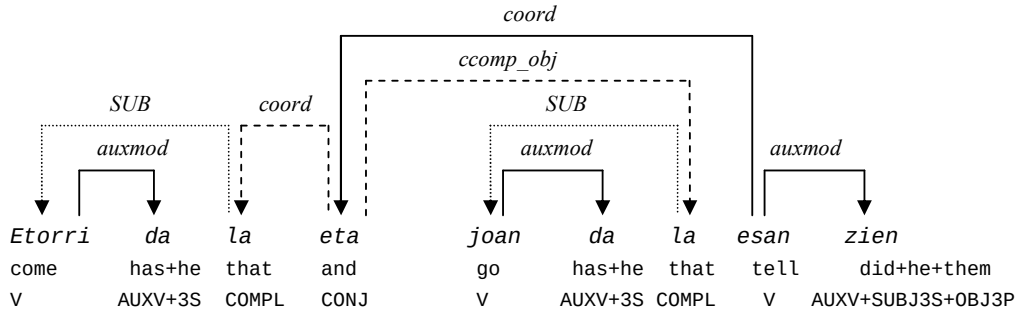

Figure 3: Transformed tree (T_S) (new arcs: dotted lines; modified arcs: discontinuous lines).

algorithm is simple, efficient and accurate, as is the case with Nivre's transition-based algorithm [11]. This transformation is totally language independent, and can be considered a standard transformation. We include it because:

- We want to test the effect of consecutive transformations against the base treebank.
- Its performance on BDT has been already tested [13]. This is in accordance with BDT having a 2.9% of non-projective arcs.

[10] proposes three types of projective transformations: path, head, and head+path. After testing them we found that the head transformation gave the best results, so this will be the one used in the following work.

3.2.2 Subordinated sentences (T_S)

Subordinated sentences are formed in Basque by attaching the corresponding morphemes to verbs, either the main verb (non-finite verbs) or the auxiliary verb (finite verbs). However, in BDT the verbal elements are organized around the main verb (semantic head) while the syntactic head corresponds to the subordination morpheme, which appears usually attached to the auxiliary. 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 the presence of a subordinated completive sentence, appears down in the tree, and this could affect their 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* 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 transforming the original tree given in Figure 2. The subordination morpheme (*-la*) is separated from the auxiliary verb (*da*), and is "promoted" as the syntactic head of the subordinated sentence. New arcs are created from the main verbs (*etorri* and *joan*) to the morpheme (which is now the head), also adding a new dependency relation (SUB). Figure 3 shows that the tree suffers important transformations. However, as the order of sentence elements is maintained, the transformation does not so greatly affect the annotated treebank (see Figure 1), and the transformations can be described by changes in dependency links and splitting of words together with each morpheme's morphological features.

A similar solution was proposed by [6] when parsing the Prague Dependency Treebank, where relative clauses are annotated introducing an additional level with a new

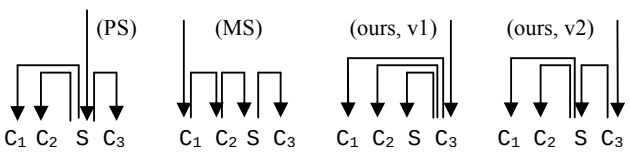

Figure 4: Dependency structures for coordination.

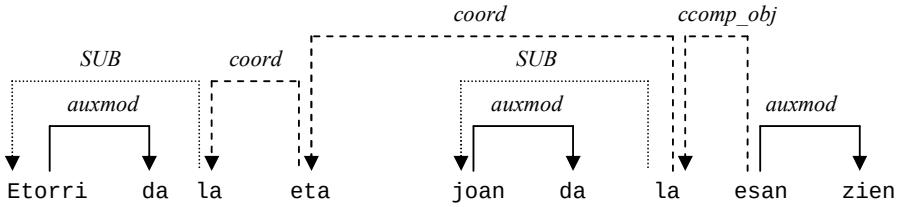

Figure 5: Transformed tree ($T_S + T_{C(v1)}$).

category (SBAR), that helps distinguish simple VPs from relative subordinated sentences. We have extended this idea to most types of subordinated sentences, as relative clauses, temporal clauses and completive, indirect interrogative, causal, adversative and modal clauses. An important difference with respect to this work is that in [4] the change is performed on the shape of the (constituency) trees, not affecting the input sequence of words, while in our case the morphemes are detached from the root words.

Transformations on finite verbs are similar to those in Figure 3 (e.g., *dela* is transformed to *da*(AUXV) + *-la*(COMPletive)). Non-finite verbs are transformed separating the suffix from the main verb (so, *etortzea* is transformed to *etorri*(V) + *-tzea*(COMPletive)).

3.2.3 Coordination (T_C)

This transformation can be considered general but it is also language dependent, as it depends on the specific configurations present in each different language, mainly the set of coordination conjunctions and also the types of elements that can be coordinated, together with their morphosyntactic properties (such as head initial or head final). Basque is considered a head final language, where many important syntactic features, like case or subordinating conjunction are located at the end of constituents. Coordination in BDT has been annotated in the so called Prague Style (PS, see Figure 4), where the conjunction (represented as S in Fig. 4) is taken as the head, and the conjuncts depend on it. [10] advocates the Mel´cuk style (MS) for parsing Czech, taking the first conjunct as the head, and creating a chain where each element depends on the preceding one (they also test its effectiveness with Arabic and Slovene). Being Basque head-final, we propose two symmetric variations of MS. In the first one (v(ersion)1 in Figure 4) the coordinated elements will all be dependents of the last conjunct (which will be the head), going from left to right. In the second version (v2), the final conjunct is again the head, and the coordination conjunction dependent on it, while the rest of the dependents attach to the conjunction. Figure 5 shows the effect of applying the v1 transformation to the tree in Figure 3.

Table 1. Top scores for Basque dependency parsing.

	System	LAS
CoNLL 2007	Nivre et al. [12]	76.94%
	Carreras [3]	75.75%
	Titov and Henderson [14]	75.49%
	Hall et al. (singlemalt) [8]	74.99%

3.3 Impact of transformations

Figure 5 shows that an important number of arcs can be modified. A negative consequence could be that the original tree structure could be lost. This would have the effect that the expected improvement could be compensated by the noise introduced by the algorithms. In this regard, we have evaluated that the transformations can be recovered with more than 97% precision.

4 Evaluation

Training and testing of the system have been performed on the same datasets presented at the CoNLL 2007 shared task, which will allow for direct comparison of the results (see Table 1). The best system obtained a score of 76.94% on Labeled Attachment Score (LAS). This system combined six different variants of a base parser (Maltparser), being the first system in 5 (out of 11) languages, competing with 19 systems in the case of Basque.

Our work will consist in applying different treebank transformations using the same treebank and the same base parser, so we can consider the last system in Table 1 as our baseline. The singlemalt parser described in [8] obtained the fifth position at CoNLL 2007. This system tried to optimize Maltparser's results on BDT by tuning parameters and selecting different training configurations. This system applied the projectivization transformation (T_P).

Evaluation was performed dividing the treebank in two sets: training set (50,000 tokens, using 10-fold cross validation) and test set (5,000 tokens). Table 2[4] presents the LAS scores of the different tests. First, we calculated the result for the system trained in the absence of morphosyntactic features (except POS and CPOS), which

[4] Statistical significance was assessed using Dan Bikel's randomized parsing evaluation comparator with the default setting of 10,000 iterations (*: Statistically significant, with $p < 0.05$; (**: Statistically significant, with $p < 0.01$)

Table 2. Evaluation results
(F+: feature optimization, T_P, T_C, T_S: transformations for projectivization, coordination and subordinated sentences).

		LAS			
	System	**10-fold cross validation**		**Test**	
1	Without morphological features	69.93% 68.35%		66.89%	
2	**Full morphology (baseline)**	76.15%		74.52%	
3	Hall et al., 2007 (full morphology + T_P) [8]	-		74.99%	(+0.47)
4	T_P	76.59%	(+0.44)	**75.54%	(+1.02)
5	$T_{C(MS)}$	72.05%	(-4.10)	69.99%	(-4.53)
6	$T_{C(v1)}$	76.43%	(+0.28)	**75.25%	(+0.73)
7	$T_{C(v2)}$	76.35%	(+0.20)	**74.93%	(+0.41)
8	T_S	76.06%	(-0.09)	73.94%	(-0.58)
9	F_+	75.98%	(-0.17)	75.01%	(+0.49)
10	$T_S + T_P + T_C$	77.32%	(+1.17)	*75.84%	(+1.32)
11	$T_S + T_P$	77.03%	(+0.88)	*75.44%	(+0.92)
12	$F_+ + T_P + T_{C(v1)}$	76.55%	(+0.40)	**75.89%	(+1.37)
13	$F_+ + T_{C(v1)} + T_S + T_P$	77.52%	(+1.37)	**76.51%	(+2.03)
14	$F_+ + T_S + T_P + T_{C(v1)}$	77.52%	(+1.37)	**76.80%**	(+2.28)

gives 66.89% LAS. The second row shows the results using the full set of morphological features, which we take as the baseline, as it presents a system optimized on the basic BDT version (regarding coordination, this version contained the original Prague Style annotation). The second and third rows in Table 2 can be considered a strong baseline, as the CoNLL systems tested many variants of training and parse configurations, mainly taking into account morphological features, that are crucial when dealing with morphologically rich languages.

The table shows the LAS scores calculated on several of the multiple combinations that were experimented. Rows 5, 6, and 7 show the effect of transforming coordinate structures, compared to the baseline (PS, row 2). MS presents the worst results (-4.53 lower than PS on the test set). They also shows that v1 and v2 transformations are more suitable than PS as the target representation. A partial explanation can be found in the effect of "short-dependency preference", as MS presents the longest average dependency-length, followed by PS, v2 and v1. The rest of the tests were performed using the best transformation (v1).

The results show how the application of all kinds of transformations improves significantly the results, giving a best score of 76.80% (14^{th} row) on the test set, which is near the best CoNLL 2007 (combined) system.

The table also shows how the order of application of the tree transformation affects the overall results in both cross validation and test set. For example, T_S is dependent on T_P, as the results vary changing their relative order of application. We corroborated this result when examining the transformed treebanks, and found that T_S leads to loss of projectivity, adding a new set of non-projective arcs. This implies that the results are better if T_S precedes T_P. We made a study of the relations involved between subordinated sentences and their heads, such as cmod (clausal modifier) or xcomp_subj (clausal complement acting as subject), and found that T_S maintained recall on the set of subordinating dependency relations and also augmented precision significantly (for dependencies that link subordinate and main sentences, recall and precision increase 3.05% and 4.13%, respectively).

5 Related work

Collins [4] applied his parser to Czech, a highly-inflected language, which shares several characteristics with Basque. [6] applies Collin's parser to Spanish, concluding that morphological information improves the analyzer.

[7] experiments the use of several types of morphosyntactic information in the analysis of Turkish, showing how the richest the information improves precision. In a related work, Eryiğit and Oflazer (2006) also show that using morphemes as the unit of analysis (instead of words) gets better results, in line with T_S results.

[6] conclude that an integrated model of morphological disambiguation and syntactic parsing in Hebrew Treebank parsing, improves the results of a pipelined approach. Dividing words into morphemes fits into this idea, as we postpone the treatment of subordination morphemes from morphology to syntax.

[9, 10] present the application of pseudoprojective and coordination transformations to several languages using maltparser, showing that they improves the results. As for coordination, they only test the PS and MS variants.

6 Conclusions

We have tested a number of transformations in the Basque Dependency Treebank, such as:

- Feature optimization. Basque is a morphologically rich language and presents many opportunities to tune the set of morphosyntactic features, adding, deleting, generalizing or specializing features.

- Projectivization. This is a language independent transformation already tested in several languages.
- We also tested two language specific transformations, such as coordination and modification of subordinated sentences. They cause important changes in the trees, but also help to improve results. In the case of coordination, we have shown that it is dependent on the specific features of each language.
- We also have found that the order of transformations can be relevant. This effect opens the study of which factors affect the order of transformations, as the creation of non projective arcs or the average length of dependency arcs.

Overall, one of the applied transformations is totally language-independent (projectivization, T_P). T_C (coordination) can be considered in the middle, as it depends on the general characteristics of the language. Finally, feature optimization, and the transformation of subordinated sentences (T_S) are specific to the treebank and intrinsecally linked to the agglutinative nature of Basque. The transformations affect a considerable number of dependencies (between 5.94% and 11.97% of all arcs). The best system, after applying all the transformations, obtains a 76.80% LAS (2.24% improvement over the baseline) on the test set, which is the best reported result for Basque dependency parsing using a single parser, and close to the better published result for a combined parser (76.94%).

The results on feature optimization do not allow us to extract a definite conclusion, as it does not help on development data but gives an improvement on test data. [7] argues that "adding inflectional features as atomic values was better than taking certain subsets with linguistic intuition …" due to the ability of SVMs to do this successfully. However, Table 2 shows that feature optimization slightly increases LAS when transformations are combined (see the improvement in $T_S + T_P + T_C$ with and without F+).

$T_S + T_P$ shows how the use of morphological information gives a substantial improvement in accuracy, even when the number of modified dependency links is modest in relation with the full size of the treebank (this transformation affects 5.94% of all arcs). Another interesting result is that when applying several types of transformations, the order of application is significant, as earlier transformations can condition the following ones. This has been demonstrated in the case of T_S, which introduces a new set of non-projective arcs, and does not give an improvement unless it is combined with T_P. The relations among the rest of the transformations deserve future examination, as the actual results do not allow us to extract a precise conclusion. For example, T_C seems to be independent of the rest of transformations.

7 Acknowledgements

This research was supported in part by the Basque Government (EPEC-RS: Basque corpus annnotated with argumental structures and semantic roles, S-PE08UN48) and the University of the Basque Country (EHU-EJIE: a semantically annotated corpus for Basque, EJIE07/05).

8 References

[1] Itziar Aduriz, Maria J. Aranzabe, Jose M. Arriola, Aitziber Atutxa, Arantza Diaz de Ilarraza, Aitzpea Garmendia and Maite Oronoz. 2003. Construction of a Basque dependency treebank. Workshop on Treebanks and Linguistic Theories.

[2] Dan Bikel. 2004. On the Parameter Space of Generative Lexicalized Statistical Parsing Models. PhD Thesis, University of Pennsylvania.

[3] Xavier Carreras. 2007. Experiments with a high-order projective dependency parser. In Proceedings of the CoNLL 2007 Shared Task (EMNLP-CoNLL).

[4] Michael Collins. 1999. Head-Driven Statistical Models for Natural Language Parsing. Ph.D. thesis, Univ. of Pennsylvania.

[5] Shay B. Cohen and Noah A. Smith. 2007. Joint Morphological and Syntactic Disambiguation. In Proceedings of the CoNLL 2007 Shared Task.

[6] Brooke Cowan and Michael Collins. 2005. Morphology and Reranking for the Statistical Parsing of Spanish. In Proceedings of EMNLP 2005.

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

[8] 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.

[9] Timo Järvinen, Pasi Tapanainen. 1998. Towards an Implementable Dependency Grammar. Workshop on Processing of Dependency-Based Grammars, COLING-ACL.

[10] Jens Nilsson, Joakim Nivre and Johan Hall. 2007. Tree Transformations for Inductive Dependency Parsing. In Proceedings of the 45th Annual Meeting of the ACL.

[11] Joakim Nivre, Johan Hall, Jens Nilsson, Chanev A., Gülsen Eryiğit, Sandra Kübler, Marinov S., and Marsi, E. 2007a. MaltParser: A language-independent system for data-driven dependency parsing. Natural Language Engineering,.

[12] Joakim Nivre, Johan Hall, Sandra Kübler, Ryan McDonald, Jens Nilsson, Sebastian Riedel and Deniz Yuret. 2007b. The CoNLL 2007 Shared Task on Dependency Parsing. In Proceedings of EMNLP-CoNLL 2007, Prague.

[13] Nivre, J. and Nilsson, J. (2005) Pseudo-Projective Dependency Parsing. In Proceedings of the 43rd ACL.

[14] Ivan Titov and James Henderson. 2007. Fast and robust multilingual dependency parsing with a generative latent variable model. Proceedings of EMNLP-CoNLL.