Testing the Effect of Morphological Disambiguation in Dependency Parsing of Basque

Kepa Bengoetxea, Koldo Gojenola
IXA NLP Group
University of the Basque Country
Technical School of Engineering, Bilbao
{kepa.bengoetxea, koldo.gojenola}@ehu.es

Arantza Casillas
IXA NLP Group
University of the Basque Country
Faculty of Science and Technology
arantza.casillas@ehu.es

Abstract

This paper presents a set of experiments performed on parsing Basque, a morphologically rich and agglutinative language, studying the effect of using the morphological analyzer for Basque together with the morphological disambiguation module, in contrast to using the gold standard tags taken from the treebank. The objective is to obtain a first estimate of the effect of errors in morphological analysis and disambiguation on the parsers. We tested two freely available and state of the art dependency parser generators, MaltParser, and MST, which represent the two dominant approaches in data-driven dependency parsing.

1 Introduction

There have been lots of attempts at parsing the Basque Dependency Treebank (BDT, Aduriz et al. 2003), starting from the CoNLL 2007 Shared Task on Dependency Parsing (Nivre et al. 2007a), where multiple systems competed on getting the best parsing results, and continued by the work done by Bengoetxea and Gojenola (2009a, 2009b, 2010). However, in all of these works, the input to the parser was the set of gold standard part of speech (POS) and morphosyntactic tags (corresponding to case, number and a number of morphological information types) taken directly from the treebank, meaning that there were no errors in the first stage of converting raw texts to morphosyntactically analyzed ones, previous to applying the parsers.

Typically, morphologically rich languages are morphologically very ambiguous. For example, in the case of 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. Disambiguating the output of morphological analysis, in order to obtain a single interpretation for each word-form, 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 interpretations, regarding to part of speech or to morphological features, can difficult the work of the parser.

For that reason, in this work we have attempted the first evaluation of two data-driven parser generators, taking the output of the morphological analysis and disambiguation as their input. As morphological ambiguity is very high compared to other languages such as English, this could hypothetically harm the results of syntactic analyzers.

Although there have been several attempts at integrating morphological and syntactic processing of several languages such as Hebrew (Goldberg and Tsarfaty 2008) or Latin, Czech, Greek and Hungarian (Lee et al. 2011), in the present work we will test the simpler option of using a pipelined approach, where the texts are passed first through morphosyntactic analysis and disambiguation, forcing a single interpretation per word-form, and then passing it to the parser. This can give an upper limit on the increase of the error rate due to incorrect interpretations from morphological disam-

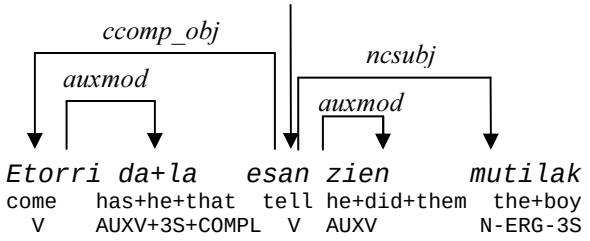

Figure 1. Dependency tree for the sentence *Etorri dela esan zien mutilak*.
(V = main verb, N = noun, AUXV = auxiliary verb, COMPL = completive, ccomp_obj = clausal complement object, ERG = ergative, S: singular, auxmod = auxiliary, ncsubj = non-clausal subject).

biguation, and could also serve as a starting point for more elaborate integrated approximations.

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 (BDT), subsection 2.2 will explain the main details of the morphological analysis and disambiguation modules for Basque (Aduriz et al. 1997, 2000), while subsection 2.3 will present the main characteristics of MaltParser, and MST, two state of the art data-driven dependency parsers.

2.1 The Basque Dependency Treebank

Basque can be described as an agglutinative language that presents a high power to generate inflected word-forms, with free constituent order of sentence elements with respect to the main verb. The BDT can be considered a pure dependency treebank from its original design, due mainly to the syntactic characteristics of Basque.

```
Etorri  dela    esan  zien  mutilak
come  that-has tell  did   boy-the
The boy told them that he had come
```

Example 1. Example of a treebank sentence.

Figure 1 presents the sentence from example 1. Each word contains its form, lemma, category (POS), subcategory, morphological features, 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*[1] features (case, number, type of subordinated sentence, …), which are relevant for parsing. The first version of the Basque Dependency Treebank contained 55,469 tokens forming 3,700 sentences (Aduriz et al., 2003), and it was used as one of the evaluated treebanks in the CoNLL 2007 Shared Task on Dependency Parsing (Nivre et al., 2007a). Our work will make use of the second version of the BDT, which is the result of a extension and redesign of the original requirements, containing 150,000 tokens (11,225 sentences), a three-fold increase.

2.2 Morphological Analysis

The morphological analyzer for Basque (Aduriz et al. 2000) consists of two subsystems. The first one performs a robust analysis based on two-level morphology, while the second part organizes the rich information contained in each word-form, by means of a unification-based grammar. This word-level grammar organizes the wealth of information provided by suffixes corresponding to derivation, word composition, and affixes that convey information about case or number (nouns, adjectives, determiners but also verbs), aspect, tense and morphemes corresponding to different types of subordination (for verbal categories).

The output of the morphological analyzer contains 2.65 interpretations per word-form. For example, the verb *zien* in figure 1 is ambiguous between a main verb and an auxiliary, and each interpretation is also ambiguous, as it can be a past tense verb, a relative sentence or an indirect interrogative question, giving 6 interpretations.

Next, there is a module for morphological disambiguation (Ezeiza et al. 1998), which uses a combination of knowledge-based disambiguation, by means of the Constraint Grammar formalism (Karlsson et al. 1995, Aduriz et al. 1997), and a posterior statistical disambiguation module, using an HMM. This second statistical module can be parameterized according to the level of disambiguation that the user wants to obtain, in an attempt to trade off precision and recall. For example, the system allows to only disambiguate based on the main categories, abstracting over

[1] We will use the term *morphosyntactic* to name the set of features attached to each word-form, which by the agglutinative nature of Basque correspond to both morphology and syntax.

morphosyntactic features. This option maintains most of the correct interpretations but, on the other hand, it still gives an output with several interpretations per word-form (for example, the system chooses the correct category, but does not decide on case or number ambiguity). In our experiments we applied the option that disambiguated most. This option maintains more than 97% of the correct interpretations, still leaving a remaining ambiguity of 1.3 interpretations, that can be considered high compared to languages like English. The 97% limit was established as a compromise between recall and precision, as in Karlsson et al. (1995).

2.3 Parsers

We have made use of MaltParser (Nivre et al. 2007b, Nivre 2006) and MSTParser (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 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 specification of 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.

MSTParser 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[2]. In the experiments we will make use of the second order non-projective algorithm, which gave the better results on the base treebank.

3 Experiments and Evaluation

In this section we will first present the process of annotating the treebank with the tags given by morphological analysis and disambiguation, and then we will report the main results.

3.1 Morphological Analysis / Disambiguation

When applying morphological analysis and disambiguation to a treebank that was manually annotated, there is the problem of matching the tokens of the treebank with those obtained from the morphological analyzer, as there were divergences on the treatment of multiword units, mostly coming from Named Entities, verb compounds and complex postpositions (those 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 those of the treebank, obtaining a correct match for about two thirds of the multiwords. Named Entities had the best matching score, while other phenomena such as complex postpositions, which have a wide variety, were not covered at all. After this matching stage, we selected those sentences giving a one-to-one direct correspondence for each token. This left us with a considerable reduction of the data, from the original 150,000 tokens to 97,000. The alignment of the rest of the sentences is left as future work. The reduction on the treebank size could lead to question about the relevance of the remaining data after the non-matching sentences have been discarded, because it could seem that those sentences were harder to parse (in principle they are candidates to having more morphological errors). However, the results on the full and the reduced treebanks confirmed that the reduction in accuracy was proportional to the treebank size, meaning that discarding a portion of the treebank did not have any side effects apart from a proportional drop in the results.

[2] http://mstparser.sourceforge.net

	MaltParser		MSTParser	
	LAS	UAS	LAS	UAS
Baseline (training = gold tags, test = gold tags)	78.78%	84.02%	78.93%	84.94%
Training = gold tags, test = automatic tags	76.57% (-2.21)	82.24% (-1.78)	76.62% (-2.31)	82.91% (-2.03)
Training = automatic tags, test = automatic tags	76.77% (-2.01)	82.46% (-1.56)	77.20% (-1.73)	83.76% (-1.18)

Table 1. Evaluation results.

(LAS: Labeled Attachment Score, UAS: Unlabeled Attachment Score)

As the morphological disambiguation process leaves a reduced ambiguity of 1.3 interpretations per word-form, and the parsers we will use require a single interpretation, we took the simplest option of choosing the first interpretation, which corresponds to taking the most frequent option. This leaves open the investigation of more complex approaches trying to select the most appropriate reading. This is not an easy task, as the ambiguity left is the hardest to solve, because the knowledge-based and statistical disambiguation processes have not been able to determine a single reading. Among the remaining types of ambiguity that were left, we can distinguish several types:

- Nominal. It includes all the categories that can bear case, such as nouns, adjectives and determiners (but also verbs). The *case* feature is important, as it carries the information necessary to correctly attach NPs and postpositional phrases to main verbs. It appears only at the last noun of the whole phrase.
- Verbal. Auxiliary verbs are very ambiguous, as all of them can also be interpreted as main verbs. Moreover, all of the past tense verbs are additionally ambiguous regarding several types of subordination sentences (relative clause, indirect interrogative or modal).

3.2 Results

Table 1 shows the results of applying the two parsers on the selected data. We did the typical train-development-test split, using 80%, 10% and 10% of the test data, respectively. In the present work we only performed a single run for each experiment, so we did not made use of the development set, using directly the test set for evaluation. For future work, we plan to use the development set for experimenting different alternatives. The first line in table 1 shows the baseline when using the gold standard tags, in accord with previous results on parsing the BDT (Bengoetxea and Gojenola 2010).

For testing the output of automatic morphological processing we performed two different kinds of experiment. In the first set we used the treebank with the gold standard tags for training. In the second option we trained the parsers giving as input the training set with the tags obtained after the process of morphological disambiguation. This way, the parsers were trained on the output from morphological disambiguation, and we will be able to compare whether it is better to train the parser using gold morphological tags or otherwise the parsers can benefit learning from the real input using morphological analysis and disambiguation.

The second line in table 1 shows that, when using the gold standard tags from the treebank for training, both parsers suffer a similar decrease in accuracy in LAS and UAS of approximately two absolute points, which is surprising in our opinion, as we expected a bigger drop in performance due to the potentially hard task of reducing 2.65 interpretations per word-form to a single interpretation. This can be due in part to the careful approach to disambiguation, combining both rule-based and statistical disambiguation (Ezeiza et al. 1998), but we must also acknowledge the use of a very robust tool for morphological analysis (Aduriz et al. 2000), which reduces the number of unrecognized or incorrectly analyzed words, incorporating sophisticated algorithms for handling out of vocabulary words, e.g. special types of two-level rules for them. On the other hand, we can also say that some morphosyntactic errors can be transparent to the parsers, as some categorial errors, e.g. noun versus adjective, will not harm the parser as long as the morphological information (basically case) is correct, because the correct determination of the case is what the parser needs to assign the correct dependency relation (subject, object or modifier).

The table also shows (third line) that the results improve when training the parsers with the same tags provided by automatic morphological analysis and disambiguation, as the parsers can in some

	MaltParser	MSTParser
	LAS	LAS
Baseline (training and test with gold tags)	78.78%	78.93%
Training = automatic tags, test = automatic tags	76.77%	77.20%
Sentences with errors in morphological tags (correct POS)	75.48%	75.96%
Sentences with errors in POS tags	72.13%	72.21%

Table 2. Evaluation results on sentences with morphosyntactic errors.

way *learn* working on the errors of the morphological modules. We also see that MSTParser seems to be slightly more robust than MaltParser when dealing with automatically obtained morphosyntactic tags, although not statistically significant.

In order to have a more detailed snapshot of the decrease of performance, we selected two subsets of sentences for a more detailed evaluation, with the aim of examining the effect of morphological disambiguation, counting only the sentences containing disambiguation errors. This will allow a better estimate of the impact of errors on these sentences. We distinguished two types of errors:

- Errors in POS. In principle, these errors could be considered the most harmful, as an error determining the main category of a word can have devastating effects. For example, this errors can typically result from the confusion of a verb as a noun or adjective. Another important subtype of this set of errors is the distinction between main and auxiliary verbs.
- Errors in morphosyntactic features (with the correct POS). They can also have an important impact on the results. For example, there is a systematic ambiguity between the ergative and the absolutive cases, which is closely related to determining the subject and object of a sentence. Another type corresponds to past tense verbs, which are ambiguous between a simple past tense verb, a relative sentence or an indirect interrogative sentence.

Table 2 shows how the performance drops around three absolute points with respect to the gold standard tags when we only take the sentences containing morphosyntactic errors (around half of the sentences), and six points when considering sentences with categorial or POS errors (which affects to one quarter of the sentences).

4 Conclusions and future work

We have presented a set of experiments studying the effect of using the morphological analyzer for Basque, in contrast to using the gold standard tags taken from the treebank. The objective was to obtain a first estimate of the effect of errors in morphological analysis and disambiguation on the parsers. We tested two different freely available and state of the art dependency parser generators, MaltParser and MSTParser.

As a main result, we can say that the errors due to incorrect disambiguation are not as important as it could be initially expected due to the high morphosyntactic ambiguity given by the Basque morphological analyzer. We have shown how morphological disambiguation errors drop the performance of the parsers in 2 absolute LAS points. MSTParser seems to be slightly more robust than MaltParser, although by a small difference.

For a future work we leave the task of correctly disambiguating the ambiguous sets of morphosyntactic readings. This could be solved by either integrating parsing and disambiguation (Cohen and Smith 2007, Goldberg and Tsarfaty 2008, Lee et al. 2011) or also redesigning the currently used modules. The key could be that the morphological disambiguation module that we used was defined independently, trying to maximize the number of correctly disambiguated tokens, while the same system could also be optimized having parsing in mind, that is, examining which kind of disambiguation errors give the most/less parsing errors. Another important line of research consists in a careful examination of the errors regarding to different types of part of speech and dependency relations, which can provide new insights.

Acknowledgements

This research was supported by the Department of Industry of the Basque Government (IE09-262, IT344-10), the University of the Basque Country (GIU09/19, INF10/65) and the Spanish Ministry of Science and Innovation (MICINN, TIN2010-20218, TIN2009-06135-E, TEC2009-06876-E/TEC, TIN2009-14675-C03-01).

References

Aduriz I., Arriola J.M., Artola X., Díaz de Ilarraza A., Gojenola K., Maritxalar M. 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, 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*.

Xabier Artola, Arantza Díaz de Ilarraza, Nerea Ezeiza, Koldo Gojenola, Gorka Labaka, Aitor Sologaistoa, Aitor Soroa. 2005. A framework for representing and managing linguistic annotations based on typed feature structures. *Proceedings of the International Conference on Recent Advances in Natural Language Processing*, RANLP 2005.

Kepa Bengoetxea and Koldo Gojenola. 2009a. Exploring Treebank Transformations in Dependency Parsing. *Proceedings of the International Conference on Recent Advances in Natural Language Processing*, RANLP'2009.

Kepa Bengoetxea and Koldo Gojenola. 2009b. Application of feature propagation to dependency parsing. *Proceedings of the International Workshop on Parsing Technologies* (IWPT'2009).

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, Los Angeles, USA.

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

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

Ezeiza N., Alegria I., Arriola J.M., Urizar R., Aduriz I. 1998. Combining Stochastic and Rule-Based Methods for Disambiguation in Agglutinative Languages. *COLING-ACL'98*, Montreal.

Yoav Goldberg and Reut Tsarfaty. 2008. A Single Generative Model for Joint Morphological Segmentation and Syntactic Parsing. *Proceedings of ACL-HLT* 2008, Colombus, Ohio, USA.

Karlsson F., Voutilainen A., Heikkila J., Anttila A. 1995. *Constraint Grammar: A Language-independent System for Parsing Unrestricted Text*. Mouton de Gruyter.

John Lee, Jason Naradowsky and David A. Smith. 2011. A Discriminative Model for Joint Morphological Disambiguation and Dependency Parsing. ACL-HLT 2011, Portland, USA.

R. McDonald, K. Crammer, and F. Pereira. 2005. Online large-margin training of dependency parsers. In Proc. ACL.

R. McDonald, K. Lerman, and F. 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, Prague.

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-CoNL*L.

Joakim Nivre, Johan Hall, Jens Nilsson, Chanev A., Gülsen Eryiğit, Sandra Kübler, Marinov S., 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.