Finding Dependency Parsing Limits over a Large Spanish Corpus

Muntsa Padró[1] **Miguel Ballesteros**[2] **Héctor Martínez**[3] **Bernd Bohnet**[4]
[1]Institute of Informatics, Federal University of Rio Grande do Sul, Porto Alegre, Brazil
[2]Natural Language Processing Group, Universitat Pompeu Fabra, Barcelona, Spain
[3]Centre for Language Technology, University of Copenhaguen, Denmark
[4]School of Computer Science, University of Birmingham, United Kingdom
muntsa.padro@inf.ufrgs.br, miguel.ballesteros@upf.edu
alonso@hum.ku.dk, bohnetb@cs.bham.ac.uk

Abstract

This paper studies the performance of different parsers over a large Spanish treebank. The aim of this work is to assess the limitations of state-of-the-art parsers. We want to select the most appropriate parser for Subcategorization Frame acquisition, and we focus our analysis on two aspects: the accuracy drop when parsing out-of-domain data, and the performance over specific labels relevant to our task.

1 Introduction

Dependency parsing has been addressed from different perspectives, improving performance as better techniques are developed. Nevertheless, we may wonder whether those results are good enough to be useful for tasks that need parsed sentences as input. Depending on the task we want to tackle, we will prefer some labels to be correct with respect to others. For example, in tasks related to extraction of verb complements such as verb Subcategorization Frame (SCF) or Selectional Preference acquisition, we are specially interested in a parser that correctly detects and labels the arguments of the verbs, while we do not need to have a high accuracy in other kind of relations, such as specifiers or modifiers.

In this work, we present a study of the performance of different parsers, following the trend started by McDonald and Nivre (2007) and Hara et al. (2009). We want to maximize the performance of different systems, for Spanish, with the final goal of applying them to concrete tasks, in our case SCF acquisition. For this task, we need to develop a parser that performs well not only in terms of Labeled Attachment Score (LAS), but also that labels verb complements correctly and that performs well when annotating data that is substantially different from the training corpus.

2 Motivation

This work was motivated by the intention of building a SCF acquisition system for Spanish (Padró et al., 2013). SCF acquisition consists of acquiring from data the kind of complements which a verb can appear with (Direct Object, Indirect Object, etc.) and how this complements are fulfilled (Noun Phrase, Clause, etc.). To perform this task, state-of-the-art systems (Briscoe and Carroll, 1997; Korhonen, 2002; Messiant, 2008) use parsed data, where the complements of the verbs are already detected. Thus, the first requirement to develop a SCF acquisition system is to have a parser to annotate the input data.

We started by training *MaltParser* (Nivre and Hall, 2005; Nivre et al., 2007b) for Spanish[1] using IULA Spanish LSP Treebank (Marimon et al., 2012), which is built from technical text. The results obtained in terms of LAS were high but when studying the performance of this parser in terms of which labels and complements were correctly detected, the results showed not to be good enough to lead to satisfactory results in SCF acquisition (Padró et al., 2013). For instance, Indirect Objects (IO) were parsed with F1 around 50%, which means that it will be very unlikely to correctly learn SCFs that contain the very relevant IO label.

Furthermore, we need a parser that performs well when annotating sentences of a domain different from the training Treebank. For that reason we evaluated the parser results over the Tibidabo Treebank (Marimon, 2010), which is made

[1] http://www.iula.upf.edu/recurs01_mpars_uk.htm

up domain-general texts. Testing on different tests sections that come from different domains is customary for English, where both PTB sections 22 and 23 are used, as well as the Brown corpus. This method is to our knowledge new for Spanish dependency parsing. The results show that, as expected, the performance of the parser over this corpus decreases, making even more difficult the extraction of SCFs.

Thus, we detected two main weaknesses of the parser system: the low performance on labels that may be very important for determined tasks and its dependency on the domain. With that in mind, we evaluated other state-of-the art parsers (§3.2) to determine which parsers suffer less from these limitations.

3 Experiments

3.1 Corpus

We ran our experiments using IULA Spanish LSP Treebank[2] (Marimon et al., 2012). This corpus (henceforth IULA) contains the syntactic annotation of 42,000 sentences (around 590,000 tokens) taken from domain-specific (technical literature) texts. We used the train and test partitions provided by the Treebank developers which are publicly available for replicability.[3]

Furthermore, we used the Tibidabo Treebank (Marimon, 2010) as an alternative test set. Tibidabo contains a set of sentences extracted from the Ancora corpus (Taulé et al., 2008), which was used in the CoNLL-X Shared Task of dependency parsing (Buchholz and Marsi, 2006).

The Tibidabo Treebank was annotated using the same guidelines as IULA Treebank. Therefore, it has the same functions and tag-set as IULA, but since the sentences come from a completely different corpus, it represents a good evaluation frame with regards to the influence of domain change.

In summary, we used a training set to train the different models and two different test sets to evaluate each model. See table 1 for details about the size of the different partitions.

The treebanks used in this work contain up to 25 different dependency relations. In this work we will pay special attention to verbal arguments, i.e., verb complements and subject. Thus, the labels we are interested in are SUBJ (subject), DO (Direct Object), IO (Indirect Object), OBLC (oblique complement, a prepositional phrase with bound preposition) and PP-DIR and PP-LOC, which mark prepositional phrases for direction and location respectively.

corpus	sentences	tokens
IULA - Train	33,679	471,624
IULA - Test	8,125	114,610
Tibidabo	3,376	41,620

Table 1: Sizes of the used corpora

3.2 Parsers

In what follows we briefly describe the dependency parsers used in our experiments, the parsing approach they belong to, and how we searched for the best possible configuration

3.2.1 Transition-based parser - MaltParser and MaltOptimizer

MaltParser (Nivre and Hall, 2005; Nivre et al., 2007b) is a transition-based dependency parser generator. It was one of the best parsers in the CoNLL Shared Tasks in 2006 and 2007 (Buchholz and Marsi, 2006; Nivre et al., 2007a) and it contains four different families of transition-based parsers. A transition-based parser is based on an automaton that performs shift-reduce operations, whose transitions manage the input words in order to assign dependencies between them.

MaltOptimizer (Ballesteros and Nivre, 2012) is a system designed to optimize MaltParser models by implementing a search of parameters and feature specifications. MaltOptimizer takes a training set in CoNLL data format,[4] and provides an optimal configuration that includes the best parsing algorithm, parsing parameters and a complex feature model over the data structures involved in the parsing process.

MaltOptimizer searches the optimal model maximizing the score of a single evaluation measure, either LAS, LCM (Labeled complete match) or unlabeled evaluation measures. As we mentioned in section 2, our intention is to enhance the performance of specific labels and we have been willing to sacrifice some overall accuracy in favor of better specific models. To this end, we modified the MaltOptimizer source code to make it able to optimize over precision and recall for a specific

[2] http://iula.upf.edu/recurs01_tbk_uk.htm

[3] https://repositori.upf.edu/handle/10230/20408

[4] http://ilk.uvt.nl/conll/#dataformat

dependency label.[5] Besides improving the accuracy of the parser for a given dependency label, our intention was that we can enhance the general performance of the parser when we optimize over a dependency label which is very frequent. The idea is that the parsers have fewer candidate words for these frequent labels, and therefore, we provide better recall for the rest of labels, thereby reducing error propagation. In our experiments, besides optimizing over general LAS and LCM, we optimized for DO (a very frequent label) and for IO (a rare but relevant label).

In our experiments we ran MaltOptimizer using 5-fold cross-validation over the training corpus in order to ensure the reliability of the outcomes.

3.2.2 Maximum Spanning Tree Parser - MSTParser

MSTParser is an arc-factored spanning tree parser (McDonald et al., 2005; McDonald et al., 2006). It implements a graph-based second-order parsing model which scores all possible dependency arcs in the sentence and then extracts the dependency tree with the highest score. The score of the trees is calculated basically adding the score of every arc, having at the end a sum with the score of the whole tree.

3.2.3 Joint tagger Transition-based parser and Graph-based parser with Hash Kernel and Beam Search- Mate tools

The Mate-Tools provide two parser types: a graph-based (Bohnet, 2010) and a transition-based parser with graph-based re-scoring (completion model) that is able to perform joint PoS tagging and dependency parsing (Bohnet and Kuhn, 2012; Bohnet and Nivre, 2012). We refer to the graph-based parser as Mate-G, to the transition-based parser without graph-based re-scoring as Mate-T, and to the transition-based parser with enabled graph-based completion model to Mate-C. These parsers benefit from a passive-aggressive perceptron algorithm implemented as a Hash Kernel, which makes the parser fast to train and improves accuracy.

Mate-T provides joint PoS tagging and dependency parsing to give account for the interaction between morphology and syntax. It uses a beam search over the space of possible transitions and keeps a k number of possible PoS tags for each word instead of basing its attachment decisions on hard previously calculated PoS tags.

Mate-C is essentially a transition-based parser that uses global information to score again the elements of the beam. The completion model depends on a set of graph parameters called second and third-order factors, which describe the dependency environment of the word, such as the second-order factor that gives account for the head, the dependent and the rightmost grandchild, or the third-order factor that lists the first two children of the dependent.

4 Results and Discussion

Table 2 summarizes the results obtained (in terms of LAS) with the different parsers over both test sets. The results obtained over IULA Test are very high, which is probably due to the specificity of the treebank. The results over Tibidabo Treebank can be seen as more general results, and they show how parsers trained with IULA treebank behave when applied to a different domain.

The results given in the table are those obtained with the configuration that leads to better LAS results over IULA Test. The same configuration is used to annotate Tibidabo. For MaltParser, the best LAS was obtained when optimizing the parser for the DO dependency label, which was obtained by applying the MaltOptimizer modifications that we explained in section 3.2.1. This therefore confirms our expectations that optimizing over very frequent labels may improve the overall accuracy[6]. The algorithm selected by MaltOptimizer was Covington non-projective, which is described by Covington (2001) and included in MaltParser by Nivre (2008). For the Mate parsers, we used the default training settings for the graph-based parser. For the transition-based parser, we used a beam size of 40 and 25 training iterations.

Parser	IULA Test LAS (%)	Tibidabo LAS (%)
Malt	93.16	89.04
MST	92.72	89.36
Mate-T	94.47	91.05
Mate-C	94.70	91.43
Mate-G	94.49	91.26

Table 2: Obtained LAS for each parser

[5] The source code and the package with the changes included in MaltOptimizer can be downloaded from: http://nil.fdi.ucm.es/maltoptimizer/MaltOpt_Specific.zip

[6] Optimizing over IO did not improve significantly the results over that complement nor overall LAS

In the table, we can see that the differences of performance assert the domain difference between the corpora and that Mate parsers clearly outperform the others. The best performance is obtained with Mate-C[7].

It is not surprising that the best results are obtained with the Mate parsers since those parsers use enhanced parsing models. Nevertheless, to obtain these high results it was necessary to change the treebank configuration to use the short PoS (this is, just the category) in the position of the long PoS, and keep the long PoS (i.e, the morphology) in the feature column. When using the original configuration of the treebank (long PoS) the results obtained were much lower, and specially suffered from the domain change (LAS=93.69% for IULA Test and LAS=88.77% for Tibidabo). The Mate parsers were optimized for for the usage of CoNLL-09 data format which includes PoS tags and morphological features only. Under these conditions, the short PoS tags fit best into the PoS column and the fine grained tags into the morphologic column. The other parsers use CoNLL-X format. MST uses all columns for training, and Malt only uses the features provided in the feature model which is one of the outputs of MaltOptimizer, but changing the data did not improve the results, neither for Malt nor for MST.

4.1 Specific Label Performance

The results obtained in terms of LAS are very satisfactory (the best parsing results reported for Spanish so far), specially for Mate parsers. Nevertheless, when we study the performance over concrete labels, we see that we can not rely on the parser for some of them. Table 3 presents the results for the labels we are interested in (§3.1). The table shows Precison, Recall and F1 scores obtained with Mate-C, which is the parser that performs better not only in terms LAS but also for individual complements. The table also shows the relative frequency of the complements in each corpus. Note that some of the studied complements are terribly infrequent.

Note that, from the labels we are interested in, just the frequent ones (SUBJ and DO) are annotated with high F1. OBLC has acceptable results, but the other complements are a big source of error, having low P and R. One of the goals of the present work was to see whether it was possible to build a parser that had better performance for the critical complements (specially in terms of Precision) even if it had worse LAS. Nevertheless, the results showed that even with the parser that performed better, the Precision and Recall of the infrequent complements is too low to obtain good results in subsequent tasks that require high label-specific performance, as shown by Padró et al. (2013) for SCF acquisition.

Label	IULA Test				Tibidabo			
	Freq.	P	R	F1	Freq.	P	R	F1
SUBJ	5.90	93.23	93.43	93.33	7.35	89.12	88.66	88.89
DO	4.64	93.02	93.25	93.13	7.03	85.84	85.64	85.74
IO	0.09	67.90	51.89	58.83	0.46	66.67	48.42	56.10
OBLC	0.20	83.56	83.49	83.53	1.30	75.25	69.00	71.99
PP-DIR	0.05	56.67	43.59	49.28	0.18	57.14	16.44	25.53
PP-LOC	0.03	61.84	39.83	48.45	0.14	56.00	24.56	34.14

Table 3: Results for some labels with Mate-C. All figures are percentages.

5 Conclusions and Future Work

This work studied the limitations of state-of-the-art parsers. We trained different systems over a large Spanish treebank and tested them over a treebank from a different domain. Our experiments show that though the obtained LAS is high, the performance over some concrete labels is very low in all cases, limiting the usability of the parsed data for tasks than rely on label-specific accuracy.

One important future line is to look for parser modifications that allow the system to perform better in the labels we are interested in. To do so, one idea would be to use semantic features to give more information to the parser like in (Agirre et al., 2012). We did some preliminary experiments in that line, using information about the semantic classes for common nouns (specifically for the classes human and location), but the results showed that this information did not lead to a better performance of the parser. This is probably due to the sparsity of this information, but it is still an interesting line to study, since it lead to good results in other cases (Agirre et al., 2012).

Acknowledgements

We thank Núria Bel and Joakim Nivre for their useful comments. HM has been partially funded by the European Commission's 7th Framework Program under grant agreement 238405 (CLARA). MP has been funded by the European Project PANACEA (FP7-ICT-2010- 248064).

[7] All differences are significant (using T-test with α set to 0.05) except between Mate-C and Mate-G over Tibidabo

References

Eneko Agirre, Aitziber Atutxa, and Kepa Sarasola. 2012. Contribution of complex lexical information to solve syntactic ambiguity in basque. In *Conference on Computational Linguistics, COLING 2012*, Mumbai, India, 12/2012.

Miguel Ballesteros and Joakim Nivre. 2012. MaltOptimizer: A System for MaltParser Optimization. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC)*.

Bernd Bohnet and Jonas Kuhn. 2012. The best of bothworlds - a graph-based completion model for transition-based parsers. In *EACL*.

Bernd Bohnet and Joakim Nivre. 2012. A transition-based system for joint part-of-speech tagging and labeled non-projective dependency parsing. In *EMNLP-CoNLL*.

Bernd Bohnet. 2010. Top accuracy and fast dependency parsing is not a contradiction. In *COLING*, pages 89–97.

Ted Briscoe and John Carroll. 1997. Automatic extraction of subcategorization from corpora. In *ANLP*, pages 356–363.

Sabine Buchholz and Erwin Marsi. 2006. CoNLL-X shared task on multilingual dependency parsing. In *Proceedings of the 10th Conference on Computational Natural Language Learning (CoNLL)*, pages 149–164.

Michael A. Covington. 2001. A fundamental algorithm for dependency parsing. In *Proceedings of the 39th Annual ACM Southeast Conference*, pages 95–102.

Tadayoshi Hara, Yusuke Miyao, and Jun'ichi Tsujii. 2009. Effective analysis of causes and inter-dependencies of parsing errors. In *Proceedings of the 11th International Conference on Parsing Technologies*, IWPT '09, pages 180–191, Stroudsburg, PA, USA. Association for Computational Linguistics.

Anna Korhonen. 2002. *Subcategorization acquisition*. Ph.D. thesis, February.

M. Marimon, B. Fisas, N. Bel, B. Arias, S. Vázquez, J. Vivaldi, S. Torner, M. Villegas, and M. Lorente. 2012. The iula treebank.

Montserrat Marimon. 2010. The tibidabo treebank. Procesamiento del lenguaje natural , 2010, vol. 45, num. 1, p. 113-119.

Ryan McDonald and Joakim Nivre. 2007. Characterizing the errors of data-driven dependency parsing models. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 122–131.

Ryan McDonald, Fernando Pereira, Kiril Ribarov, and Jan Hajič. 2005. Non-projective dependency parsing using spanning tree algorithms. In *Proceedings of the Human Language Technology Conference and the Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP)*, pages 523–530.

Ryan McDonald, Kevin Lerman, and Fernando Pereira. 2006. Multilingual dependency analysis with a two-stage discriminative parser. In *Proceedings of the 10th Conference on Computational Natural Language Learning (CoNLL)*, pages 216–220.

Cédric Messiant. 2008. A subcategorization acquisition system for french verbs. In *ACL (Student Research Workshop)*, pages 55–60.

Joakim Nivre and Johan Hall. 2005. MaltParser: A language-independent system for data-driven dependency parsing. In *Proceedings of the 4th Workshop on Treebanks and Linguistic Theories (TLT)*, pages 137–148.

Joakim Nivre, Johan Hall, Sandra Kübler, Ryan McDonald, Jens Nilsson, Sebastian Riedel, and Deniz Yuret. 2007a. The CoNLL 2007 shared task on dependency parsing. In *Proceedings of the CoNLL Shared Task of EMNLP-CoNLL 2007*, pages 915–932.

Joakim Nivre, Johan Hall, Jens Nilsson, Atanas Chanev, Gülşen Eryiğit, Sandra Kübler, Svetoslav Marinov, and Erwin Marsi. 2007b. Maltparser: A language-independent system for data-driven dependency parsing. *Natural Language Engineering*, 13:95–135.

Joakim Nivre. 2008. Algorithms for deterministic incremental dependency parsing. *Computational Linguistics*, 34:513–553.

Muntsa Padró, Núria Bel, and Aina Garí. 2013. Verb SCF extraction for Spanish with dependency parsing. *Procesamiento del Lenguaje Natural*, (51), September.

Mariona Taulé, M. Antònia Martí, and Marta Recasens. 2008. Ancora: Multilevel annotated corpora for Catalan and Spanish. In *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco, may. European Language Resources Association (ELRA).