Automatic Selection of HPSG-Parsed Sentences for Treebank Construction

Montserrat Marimon*
Universitat de Barcelona

Núria Bel**
Universitat Pompeu Fabra

Lluís Padró†
Universitat Politècnica de Catalunya

This article presents an ensemble parse approach to detecting and selecting high-quality linguistic analyses output by a hand-crafted HPSG grammar of Spanish implemented in the LKB system. The approach uses full agreement (i.e., exact syntactic match) along with a MaxEnt parse selection model and a statistical dependency parser trained on the same data. The ultimate goal is to develop a hybrid corpus annotation methodology that combines fully automatic annotation and manual parse selection, in order to make the annotation task more efficient while maintaining high accuracy and the high degree of consistency necessary for any foreseen uses of a treebank.

1. Introduction

Treebanks constitute a crucial resource for theoretical linguistic investigations as well as for NLP applications. Thus, in the past decades, there has been increasing interest in their construction and both theory-neutral and theory-grounded treebanks have been developed for a great variety of languages. Descriptions of available annotated corpora can be found in Abeillé (2003) and in the proceedings from the annual editions of the International Workshop on Treebanks and Linguistic Theories.

Quantity and quality are two very important objectives when building a treebank, but speed and low labor costs are also required. In addition, guaranteeing consistency, that is, that the same phenomena receive the same annotation through the corpus, is crucial for any of the possible uses of the treebank. The first attempts at treebank projects used manual annotation mainly and devoted many hours of human labor to their construction. Human annotation is not only slow and expensive, but it also introduces errors and inconsistencies because of the difficulty and tiring nature of the

* Gran Via de les Corts Catalanes 585, 08007-Barcelona. E-mail: montserrat.marimon@ub.edu.
** Roc Boronat 138, 08018-Barcelona. E-mail: nuria.bel@upf.edu.
† Jordi Girona 1-3, 08034-Barcelona. E-mail: padro@lsi.upc.edu.

Submission received: 16 October 2012; revised submission received: 20 October 2013; accepted for publication: 5 December 2013.

doi:10.1162/COLI_a_00190

task.[1] Therefore, automating parts of the annotation process aims to leverage effectiveness, producing a larger number of high-quality and consistent analyses in shorter time and using fewer resources.

This article presents research that attempts to increase the degree of automation in the annotation process when constructing a large treebank for Spanish (the IULA Spanish LSP Treebank) in the framework of the European project METANET4U (Enhancing the European Linguistic Infrastructure, GA 270893GA).[2]

The treebank was developed using the following bootstrapping approach, details of which are presented in Sections 3 and 4:

- First, we annotated the sentences using the DELPH-IN development framework, in which the annotation process is effected by manually selecting the correct parses from among all the analyses produced by a hand-built symbolic grammar.
- Second, when a number of human-validated parsed sentences were available, we trained a MaxEnt ranker.
- Third, we trained a dependency parser with the human-validated parsed sentences converted to the CoNLL format.
- Fourth, we provided a fully automated chain based on an ensemble method that compared the parse delivered by the dependency parser and the one delivered by the MaxEnt ranker, and then accepted the automatically proposed analysis, but only if both were identical.
- Fifth, sentences rejected by the ensemble were given to human annotators for manual disambiguation.

Obviously, using fully automatic parsing would have been the best solution for speed and consistency, but no statistical parsers for Spanish are good enough yet, and when using symbolic parsers, there is no way to separate good parses from incorrect ones. The ensemble method we propose is a way of avoiding monitoring automatic parsing; the error is more than acceptable and recall is expected to be augmented by re-training and the refinement of the different parses.

After this introduction, Section 2 presents an overview of related work on automatic parse selection, Section 3 summarizes the set-up, Section 4 presents our experiments and results and, finally, Section 5 concludes.

2. Related Work

In the broadest sense, this work is situated with respect to research into automatic parse selection. Such projects have had a variety of different goals as well as different approaches, based on (i) semantic filtering techniques (Yates, Schoenmackers, and Etzioni 2006), (ii) sentence-level features (e.g., length; Kawahara and Uchimoto

1 In order to control errors, a common strategy is to control inter-annotator agreement by making two annotators work on the same sentences. This makes the task even slower and more expensive.

2 The IULA Spanish LSP Treebank contains 43,000 annotated sentences, distributed among different domains (Law, Economy, Computing Science, Medicine, and Environment) and sentence lengths (ranging from 4 to 30 words). The treebank is publicly available at `http://metashare.upf.edu`.

2008), (iii) statistics about PoS sequences in a batch of parsed sentences (Reichart and Rappoport 2009), and (iv) ensemble parse algorithms (Reichart and Rappoport 2007; Sagae and Tsujii 2007; Baldridge and Osborne 2003). Here, we focus on ensemble approaches.

Reichart and Rappoport (2007) selected high-quality constituency parses by using the level of agreement among 20 copies of the same parser, trained on different subsets of a training corpus. Experiments using training and test data for the same domain and in the *parser-adaptation* scenario showed improvements over several baselines.

Sagae and Tsujii (2007) used an ensemble to select high-quality dependency parses. They compared the outputs of two statistical shift-reduce LR models and selected only identical parses, in their case to retrain the MaxEnt model. Following this procedure, they achieved the highest score in the domain adaptation track of the CoNLL 2007 shared task.

Finally, Baldridge and Osborne (2003) used an ensemble of parsers in the context of HPSG grammars applied to *committee-based active learning*, that is, to select the most informative sentences to be hand-annotated and used as training material to improve the statistical parser and to minimize the required amount of such sentences. Using the English Resource Grammar (Flickinger 2002) and the Redwoods treebank (Oepen et al. 2002), they showed that sample selection according to preferred parse disagreement between two different machine learning algorithms (log-linear and perceptron), or between the same algorithm trained on two independent feature sets (*configurational* and *ngram* sets, based on the HPSG derivation trees), reduced the amount of human-annotated material needed to train an HPSG parse selection model compared with a *certainty-based* method based on tree entropy and several baseline selection metrics.

Like Baldridge and Osborne (2003), we investigate ensemble parsing in the context of HPSG grammars; however, our goal does not involve selecting the most informative sentences to retrain the parser, but rather to select those sentences most reliably parsed, in order to enlarge the treebank automatically. Thus, rather than selecting sentences on which two models disagree, we select those where they agree completely. In addition, we present two important contributions, going beyond what has been done in previous work. First, although parsing ensembles have previously been proposed only for closely related language models (i.e., parsers that use algorithms under the machine-learning paradigm, varying only the feature set or training data), the presented work is the first to combine parsers from different paradigms: stochastic dependency parsing and MaxEnt parse selection over parses produced by a symbolic grammar. Second, the current work is the first to propose such a methodology for parse selection as a way of overcoming the seemingly impossible task of automatically selecting good parses from automatic parsing to speed treebank production and, more importantly, to meet the requirements of high precision and high consistency that are good for all of the uses of the treebank.

3. Set-up

We select high-quality HPSG analyses using full agreement among a MaxEnt parse selection model and a dependency parser. A comparison between the two is performed on the dependency structures that we obtain converting the parse tree produced by a symbolic grammar to the CoNLL format.

3.1 HPSG Parsing and Disambiguation

Our investigation uses the Deep Linguistic Processing with HPSG Initiative (DELPH-IN),[3] an open-source processing framework also used in several treebank projects within this international initiative (Oepen et al. 2002; Flickinger et al. 2012). Using this framework, the annotation process is divided into two parts: (1) the corpus is parsed using a hand-built HPSG (Pollard and Sag 1994); (2) the grammar output is ranked by a MaxEnt-based parse ranker (Toutanova et al. 2005), and the best parse is manually selected.

The grammar applied in parsing is a broad-coverage, open-source Spanish grammar implemented in the Linguistic Knowledge Builder (LKB) system (Copestake 2002), the Spanish Resource Grammar (SRG) (Marimon 2013).

The manual selection task is performed with an interface provided as part of the [incr tstb()] grammar profiling environment (Oepen and Carroll 2000) that allows the annotator to reduce the set of parses incrementally by choosing so-called **discriminants** (Carter 1997); that is, by selecting the features that distinguish between the different parses, until the appropriate parse is left or, if none of the displayed parses is the correct one, all parses are rejected.

As always the case with symbolic grammars, the SRG produces several hundreds of analyses for a sentence. The DELPH-IN framework, however, provides a MaxEnt-based ranker that sorts the parses produced by the grammar. Although this stochastic ranker cannot be used to select automatically the correct parse without introducing a considerable number of errors (as we will show, it only achieves accuracy of about 61%), it nevertheless allows the annotator to reduce the forest to the *n*-best trees, typically the 500 top readings. The statistics that form the model of the MaxEnt ranker are gathered from disambiguated parses and can be updated as the number of annotated sentences increases.

3.2 Conversion to the CoNLL Format

The linguistic analysis produced by the LKB system for each parsed sentence provides, together with a constituent structure and a Minimal Recursion Semantics (MRS) semantic representation (Copestake et al. 2005), a derivation tree, obtained from a complete syntactico-semantic analysis represented in a parse tree with standard HPSG-typed feature structures at each node.

The derivation tree is encoded in a nested, parenthesized structure whose elements correspond to the identifiers of grammatical rules and the lexical items used in parsing. Phrase structure rules—marked by the suffix '_*c*' (for 'construction')—identify the daughter sequence, separated by a hyphen, and, in headed-phrase constructions, a basic dependency relation between sentence constituents (e.g., subject-head (*sb-hd*) and head-complement (*hd-cmp*)). Lexical items are annotated with part-of-speech information according to the EAGLES tag set for Spanish[4] and their lexical entry identifier, and they optionally include a lexical rule identifier. Figure 1 shows an example.

In order to compare the first-best trees selected by the MaxEnt selection model and the outputs of the dependency parser, we convert the derivation trees to a dependency

3 http://www.delph-in.net/.

4 See http://www.ilc.cnr.it/EAGLES96/annotate/annotate.html.

```
(hd-ad_c
   (hd_optc_c
      (hd-cmp_c
         (vmn0000 (conceder_v-np-ppa "conceder"))
         (hd-nbar_c
            (hd-pt_c
               (ncfp000 (licencia_n "licencias"))
                (fc (comma_pt ",")))))))
   (hd-cmp_c
      (cs (cuando_p-cl "cuando"))
      (fl-hd_c
         (hd_advnp_c
            (nc00000 (asi_av "así")))
         (hd_sb_c
            (cl-hc_c
               (pp3msa00 (lo_pr "lo"))
               (vmsp3p0 (v_acc_dlr (disponer_v-np "dispongan"))))
            (sp-hd_c
               (da0fp0 (el_d "la"))
               (nccp000 (ordenanza_n "ordenanzas")))))))
```

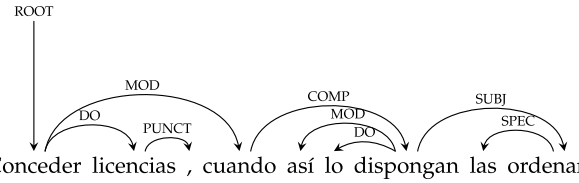

Figure 1
Derivation tree and dependency graph of *Conceder licencias, cuando así lo dispongan las ordenanzas* [To grant licences, when so stipulated by ordinances].

format, also illustrated in Figure 1. In this target annotation, lexical elements are linked by asymmetrical dependency relations in which one of the elements is considered the head of the relation and the other one is its dependant. The conversion is a fully automatic and unambiguous process that produces the dependency structure in the CoNLL format (Buchholz and Marsi 2006). A deterministic conversion algorithm makes use of the identifiers of the phrase structure rules mentioned previously, in order to identify the heads, dependants, and some dependency types that are directly transferred onto the dependency structure (e.g., subject, specifier, and modifier). The identifiers of the lexical entries, which include the syntactic category of the sub-categorized elements, enable the identification of the argument-related dependency functions.[5]

3.3 Dependency Parsing

For dependency parsing, we use MaltParser (Nivre et al. 2007). To train it, we use manually disambiguated parses among those parses produced by the HPSG grammar, converted to the dependency format we describe earlier.

5 An alternative proposal for projecting HPSG trees to CoNLL is described in Ivanova et al. (2012).

Table 1
Results of the MaxEnt model and MaltParser as labeled attachment scores, unlabeled attachment scores, labeled accuracy score, and exact syntactic match.

	LAS	*UAS*	*Label Accur Score*	*Exact Synt Match*
MaxEnt model	95.4%	96.8%	97.6%	61.0%
MaltParser	92.0%	95.0%	94.5%	43.1%

4. Experiments and Results

In our experiments, we tested the ability of the ensemble approach to select only correct parses. The experiment proceeded as follows:

- We divided a set of 15,329 sentences into a training and test set (13,901 and 1,428 sentences, respectively). Sentence length ranged from 4 to 20 words (longer sentences had not been annotated yet).
- We trained the MaxEnt model and MaltParser and ran each of the models on the test set. The results we achieved are displayed in Table 1.
- We compared the outputs of the two models and selected those sentences where both parses produced identical analyses.

The performance of our parser ensemble approach was measured through precision and recall on the task of selecting those sentences for which the first tree proposed by the MaxEnt model was the correct one. Table 2 shows the confusion matrix resulting from the experiment. The row *predicted ok* counts the number of sentences selected by our ensemble method (Malt and MaxEnt delivered parses are identical), and the row *predicted nok* contains the number of sentences not selected because the parsers disagreed. Columns *gold* present the manual evaluation of a MaxEnt model first ranked parse. From this table, we can compute precision and recall of our sentence selector: 445 sentences were selected out of the 1,428 sentences in the test set (31.2%). Precision (number of correctly selected sentences among all the selected sentences) stood at 90.6% (403/445), and recall (number of correctly selected sentences among all the actually correctly ranked first sentences) was 46.6% (403/864).

We compared the results of our ensemble method with two parse selection methods based on: (i) a simple probability-based threshold (baseline) and (ii) a parser uncertainty measure computed as *tree entropy* as used by Baldridge and Osborne (2003). The baseline consisted of selecting sentences for which the ratio between the probabilities of the two highest ranked analyses delivered by the MaxEnt model was over a given threshold.

Table 2
Confusion matrix used to assess the results in terms of precision and recall.

		gold *ok*	*gold* *nok*	*total*
predicted	*ok*	403	42	445
	nok	461	522	983
	total	864	564	1,428

The idea was that a very high ratio would indicate that the parse ranked first had a large advantage over the others, whereas if the ratio was close to 1, both the first and the second analyses would have similar probabilities, indicating lower confidence of the model in the decision. Tree entropy takes into account not just the two highest ranked analyses, but all trees proposed by the parser for that sentence. The rationale is that high entropy indicates a scattered probability distribution among possible trees (and thus less certainty of the model in the prediction), whereas low entropy should indicate that one tree (or a few) gets most of the probability mass.

Results for different thresholds (both for the baseline and tree entropy) are shown in Table 3 (top). As we can see, setting a high threshold for the baseline, we can select a small subset of 20% of the sentences with precision similar to that achieved by our parse ensemble approach. To select 31% of the sentences (i.e., about the same proportion we obtained with the ensemble approach) we need to set a threshold of 4.5, obtaining a precision of 84%, which is lower than the 90% obtained with the ensemble method.

Tree entropy exhibits similar behavior, in that a restrictive threshold can select about 15% of sentences with precision over 90%, while setting a threshold such that about 31% of sentences are selected, we obtain precision of about 75%.

Note that although the baseline has an F_1 score slightly higher than the ensemble, our goal is a high precision filter that can be utilized to select correctly parsed sentences. From this point of view, our approach beats both baselines.

The fact that tree entropy yields worse values than the baseline is somehow predictable: Given a sentence with n possible trees (note that n may be in the order of dozens or even hundreds), if a small number m of those analyses ($1 < m << n$) concentrate a large portion of probability mass but exhibit small differences between them, the sentence will be rejected by the baseline (because there is not enough distance between the first and second analyses) but will be accepted by tree entropy (because entropy will be relatively low, given the large value of n). Thus, tree entropy is a good measure for Baldridge and Osborne (2003), whose purpose is to select sentences where the model is *less* confident, but our simple baseline seems to be better when the goal is to select sentences where the first parse is the correct one.

Table 3
Top: Comparative results using different threshold values for the baselines. Bottom: Results per sentence length when selecting about 31% over all sentences. *Thr* = threshold; *%sel* = percentage of selected sentences; *P* = precision; *R* = recall; *Len* = sentence length.

Baseline				Tree entropy				Ensemble		
Thr.	*%sel*	*P*	*R*	*Thr.*	*%sel*	*P*	*R*	*%sel*	*P*	*R*
2	50.9%	67.6%	70.2%	0.2	59.6%	60.1%	73.2%			
3	38.4%	77.5%	60.8%	0.15	38.2%	71.7%	56.0%			
4.5	**31.0%**	**84.1%**	**53.3%**	**0.133**	**31.3%**	**75.7%**	**48.3%**	**31.2%**	**90.6%**	**46.6%**
10	20.8%	91.1%	38.6%	0.1	21.4%	82.0%	35.8%			
20	12.1%	97.3%	24.0%	0.075	15.1%	91.5%	28.2%			
30	9.9%	98.7%	19.9%	0.05	11.2%	96.6%	22.2%			

Len.	Baseline			Tree entropy			Ensemble		
	%sel	*P*	*R*	*%sel*	*P*	*R*	*%sel*	*P*	*R*
1-10	56.0%	96.7%	70.6%	42.8%	96.6%	53.9%	43.6%	97.7%	70.4%
11-20	19.8%	68.2%	36.9%	26.1%	60.3%	43.0%	10.3%	83.7%	33.8%
All	**31.0%**	**84.1%**	**53.3%**	**31.3%**	**75.7%**	**48.3%**	**31.2%**	**90.6%**	**46.6%**

As shown in Table 3 (bottom), behavior is different for sentences of up to 10 words than for longer sentences. All three systems have a bias towards selecting short rather than long sentences (because short sentences are more often correctly analyzed by the parser). The results for short sentences are similar in all three cases, but the ensemble approach is clearly more precise for long sentences, with only a moderate loss in recall.

5. Conclusion

We have described research that aims to increase the degree of automation when building annotated corpora. We propose a parser ensemble approach based on full agreement between a MaxEnt model and a dependency parser to select correct linguistic analyses output by an HPSG grammar. This enables a hybrid annotation methodology that combines fully automatic annotation and manual parse selection, which makes the annotation task more efficient while maintaining high accuracy and the high degree of consistency necessary for a useful treebank. Our approach is grammar-independent and can be used by any DELPH-IN-style treebank. In the future, we plan to investigate the impact of automatic treebank enlargement on the performance of statistical parsers.

Acknowledgments

This work was supported by grant *Ramón y Cajal* from Spanish MICINN and the project METANET4U. We thank the reviewers for their comments and Carlos Morell for his support.

References

Abeillé, Anne (editor). 2003. *Treebanks: Building and Using Parsed Corpora*. Kluwer, Amsterdam.

Baldridge, Jason and Miles Osborne. 2003. Active learning for HPSG parse selection. In *Proceedings of the 7th Conference on Computational Natural Language Learning*, pages 17–24, Edmonton.

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

Carter, David. 1997. The TreeBanker: A tool for supervised training of parsed corpora. In *Proceedings of the 14th National Conference on Artificial Intelligence*, pages 598–603, Providence, RI.

Copestake, Ann. 2002. *Implementing Typed Feature Structure Grammars*. CSLI Publications, Stanford, CA.

Copestake, Ann, Dan Flickinger, Carl Pollard, and Ivan A. Sag. 2005. Minimal recursion semantics: An introduction. *Research on Language and Computation*, 3(4):281–332.

Flickinger, Dan. 2002. On building a more efficient grammar by exploiting types. In *Natural Language Engineering (6)1—Special Issue: Efficiency Processing with HPSG: Methods, Systems, Evaluation*, 16(1):1–17.

Flickinger, Dan, Valia Kordoni, Yi Zhang, António Branco, Kiril Simov, Petya Osenova, Catarina Carvalheiro, Francisco Costa, and Sérgio Castro. 2012. ParDeepBank: Multiple parallel deep treebanking. In *Proceedings of the 11th Workshop on Treebanks and Linguistic Theories*, pages 97–108, Lisbon.

Ivanova, Angelina, Stephan Oepen, Lilja Ovrelid, and Dan Flickinger. 2012. Who did what to whom? A contrastive study of syntacto-semantic dependencies. In *Proceedings of the 6th Linguistic Annotation Workshop*, pages 2–11, Jeju Island.

Kawahara, Daisuke and Kiyotaka Uchimoto. 2008. Learning reliability of parses for domain adaptation of dependency parsing. In *Proceedings of the 3rd International Joint Conference on Natural Language Processing*, pages 709–714, Hyderabad.

Marimon, Montserrat. 2013. The Spanish DELPH-IN grammar. *Language Resources and Evaluation*, 47(2):371–397.

Nivre, Joakim, Johan Hall, Jens Nilsson, Atanas Chanev, Gülsen Eryigit, Sandra Kübler, Svetoslav Marinov, and Erwin Mars. 2007. Maltparser: A language-independent system for data-driven dependency parsing. *Natural Language Engineering*, 13(2):95–135.

Oepen, Stephan and John Carroll. 2000. Performance profiling for parser engineering. In *Natural Language Engineering (6)1—Special Issue: Efficiency Processing with HPSG: Methods, Systems, and Evaluation*, 16(1):81–97.

Oepen, Stephan, Dan Flickinger, K. Toutanova, and C. D. Manning. 2002. LinGo Redwoods. A rich and dynamic treebank for HPSG. In *Proceedings of the 1st Workshop on Treebanks and Linguistic Theories*, pages 139–149, Sozopol.

Pollard, Carl and Ivan A. Sag. 1994. *Head-driven Phrase Structure Grammar*. The University of Chicago Press and CSLI Publications, Chicago.

Reichart, Roi and Ari Rappoport. 2007. An ensemble method for selection of high quality parses. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, pages 408–415, Prague.

Reichart, Roi and Ari Rappoport. 2009. Automatic selection of high quality parses created by a fully unsupervised parser. In *Proceedings of the 13th Conference on Computational Natural Language Learning*, pages 156–164, Boulder, CO.

Sagae, Kenji and Jun-Ichi Tsujii. 2007. Dependency parsing and domain adaptation with LR models and parser ensembles. In *Proceedings of the Joint Meeting of the Conference on Empirical Methods in Natural Language Processing and Conference on Computational Natural Language Learning*, pages 1,044–1,050, Prague.

Toutanova, Kristina, Christoper D. Manning, Dan Flickinger, and Stephan Oepen. 2005. Stochastic HPSG parse disambiguation using the Redwoods corpus. *Research on Language and Computation*, 3(1):83–105.

Yates, Alexander, Stefan Schoenmackers, and Oren Etzioni. 2006. Detecting parser errors using Web-based semantic filters. In *Proceedings of the 11th Conference of Empirical Methods in Natural Language Processing*, pages 27–34, Sydney.