Using Machine-Learning to Assign Function Labels to Parser Output for Spanish

Grzegorz Chrupała[1] and **Josef van Genabith**[1,2]
[1]National Center for Language Technology
Dublin City University
Glasnevin, Dublin 9, Ireland
[2]IBM Dublin Center for Advanced Studies
grzegorz.chrupala@computing.dcu.ie
josef@computing.dcu.ie

Abstract

Data-driven grammatical function tag assignment has been studied for English using the Penn-II Treebank data. In this paper we address the question of whether such methods can be applied successfully to other languages and treebank resources. In addition to tag assignment accuracy and f-scores we also present results of a task-based evaluation. We use three machine-learning methods to assign Cast3LB function tags to sentences parsed with Bikel's parser trained on the Cast3LB treebank. The best performing method, SVM, achieves an f-score of 86.87% on gold-standard trees and 66.67% on parser output - a statistically significant improvement of 6.74% over the baseline. In a task-based evaluation we generate LFG functional-structures from the function-tag-enriched trees. On this task we achive an f-score of 75.67%, a statistically significant 3.4% improvement over the baseline.

1 Introduction

The research presented in this paper forms part of an ongoing effort to develop methods to induce wide-coverage multilingual Lexical-Functional Grammar (LFG) (Bresnan, 2001) resources from treebanks by means of automatically associating LFG f-structure information with constituency trees produced by probabilistic parsers (Cahill et al., 2004). Inducing deep syntactic analyses from treebank data avoids the cost and time involved in manually creating wide-coverage resources.

Lexical Functional Grammar f-structures provide a level of syntactic representation based on the notion of grammatical functions (e.g. Subject, Object, Oblique, Adjunct etc.). This level is more abstract and cross-linguistically more uniform than constituency trees. F-structures also include explicit encodings of phenomena such as control and raising, pro-drop or long distance dependencies. Those characteristics make this level a suitable representation for many NLP applications such as transfer-based Machine Translation or Question Answering.

The f-structure annotation algorithm used for inducing LFG resources from the Penn-II treebank for English (Cahill et al., 2004) uses configurational, categorial, function tag and trace information. In contrast to English, in many other languages configurational information is not a good predictor for LFG grammatical function assignment. For such languages the function tags included in many treebanks are a much more important source of information for the LFG annotation algorithm than Penn-II tags are for English.

Cast3LB (Civit and Martí, 2004), the Spanish treebank used in the current research, contains comprehensive grammatical function annotation. In the present paper we use a machine-learning approach in order to add Cast3LB function tags to nodes of basic constituent trees output by a probabilistic parser trained on Cast3LB. To our knowledge, this paper is the first to describe applying a data-driven approach to function-tag assignment to a language other than English.

Our method statistically significantly outperforms the previously used approach which relied exclusively on the parser to produce trees with Cast3LB tags (O'Donovan et al., 2005). Additionally, we perform a task-driven evaluation of our Cast3LB tag assignment method by using the tag-enriched trees as input to the Spanish LFG f-structure annotation algorithm and evaluating the quality of the resulting f-structures.

Section 2 describes the Spanish Cast3LB treebank. In Section 3 we describe previous research in LFG induction for English and Spanish as well

as research on data-driven function tag assignment to parsed text in English. Section 4 provides the details of our approach to the Cast3LB function tag assignment task. In Sections 5 and 6 we present evaluation results for our method. In Section 7 we present the error analysis of the results. Finally, in Section 8 we conclude and discuss ideas for further research.

2 The Spanish Treebank

As input to our LFG annotation algorithm we use the output of Bikel's parser (Bikel, 2002) trained on the Cast3LB treebank (Civit and Martí, 2004). Cast3LB contains around 3,500 constituency trees (100,000 words) taken from different genres of European and Latin American Spanish. The POS tags used in Cast3LB encode morphological information in addition to Part-of-Speech information.

Due to the relatively flexible order of main sentence constituents in Spanish, Cast3LB uses a flat, multiply-branching structure for the S node. There is no VP node, but rather all complements and adjuncts depending on a verb are sisters to the *gv* (Verb Group) node containing this verb. An example sentence (with the corresponding f-structure) is shown in Figure 1.

Tree nodes are additionally labelled with grammatical function tags. Table 1 provides a list of function tags with short explanations. Civit (2004) provides Cast3LB function tag guidelines.

Functional tags carry some of the information that would be encoded in terms of tree configurations in languages with stricter constituent order constraints than Spanish.

Tag	Meaning
ATR	Attribute of copular verb
CAG	Agent of passive verb
CC	Compl. of circumstance
CD	Direct object
CD.Q	Direct object of quantity
CI	Indirect object
CPRED	Predicative complement
CPRED.CD	Predicative of Direct Object
CPRED.SUJ	Predicative of Subject
CREG	Prepositional object
ET	Textual element
IMPERS	Impersonal marker
MOD	Verbal modifier
NEG	Negation
PASS	Passive marker
SUJ	Subject
VOC	Vocative

Table 1: List of function tags in Cast3LB.

3 Previous Work

3.1 LFG Annotation

A methodology for automatically obtaining LFG f-structures from trees output by probabilistic parsers trained on the Penn-II treebank has been described by Cahill et al. (2004). It has been shown that the methods can be ported to other languages and treebanks (Burke et al., 2004; Cahill et al., 2003), including Cast3LB (O'Donovan et al., 2005).

Some properties of Spanish and the encoding of syntactic information in the Cast3LB treebank make it non-trivial to apply the method of automatically mapping c-structures to f-structures used by Cahill et al. (2004), which assigns grammatical functions to tree nodes based on their phrasal category, the category of the mother node and their position relative to the local head.

In Spanish, the order of sentence constituents is flexible and their position relative to the head is an imperfect predictor of grammatical function. Also, much of the information that the Penn-II Treebank encodes in terms of tree configurations is encoded in Cast3LB in the form of function tags. As Cast3LB trees lack a VP node, the configurational information normally used in English to distinguish Subjects (NP which is left sister to VP) from Direct Objects (NP which is right sister to V) is not available in Cast3LB-style trees. This means that assigning correct LFG functional annotations to nodes in Cast3LB trees is rather difficult without use of Cast3LB function tags, and those tags are typically absent in output generated by probabilistic parsers.

In order to solve this difficulty, O'Donovan et al. (2005) train Bikel's parser to output complex category-function labels. A complex label such as *sn-SUJ* (an NP node tagged with the Subject grammatical function) is treated as an atomic category in the training data, and is output in the trees produced by the parser. This baseline process is represented in Figure 2.

This approach can be problematic for two main reasons. Firstly, by treating complex labels as atomic categories the number of unique labels increases and parse quality can deteriorate due to sparse data problems. Secondly, this approach, by relying on the parser to assign function tags, offers

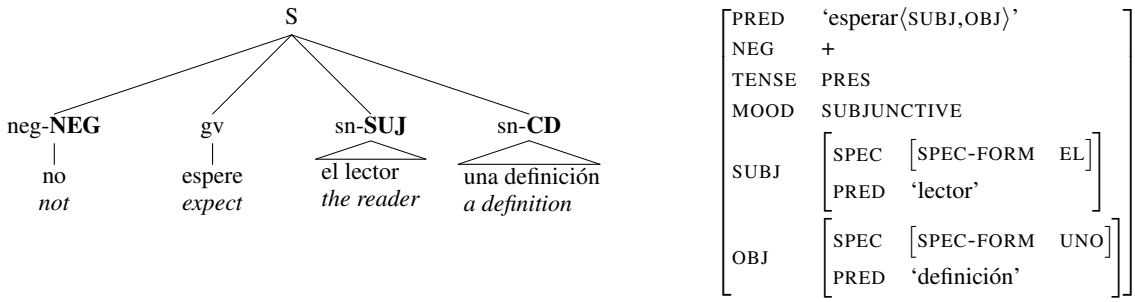

Figure 1: On the left flat structure of S. Cast3LB function tags are shown in bold. On the right the corresponding (simplified) LFG f-structure. Translation: *Let the reader not expect a definition.*

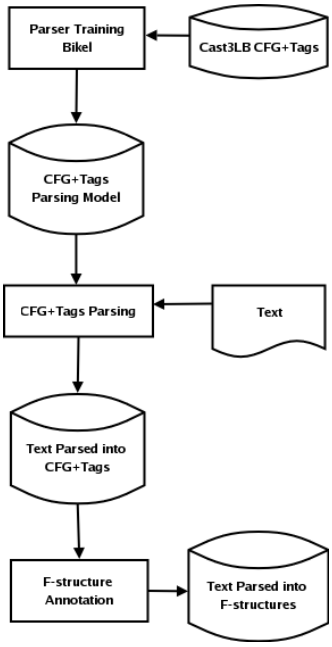

Figure 2: Processing architecture for the baseline.

limited control over, or room for improvement in, this task.

3.2 Adding Function Tags to Parser Output

The solution we adopt instead is to add Cast3LB functional tags to simple constituent trees output by the parser, as a postprocessing step. For English, such approaches have been shown to give good results for the output of parsers trained on the Penn-II Treebank.

Blaheta and Charniak (2000) use a probabilistic model with feature dependencies encoded by means of feature trees to add Penn-II Treebank function tags to Charniak's parser output. They report an f-score 88.472% on original treebank trees and 87.277% on the correctly parsed subset of tree nodes.

Jijkoun and de Rijke (2004) describe a method of enriching output of a parser with information that is included in the original Penn-II trees, such as function tags, empty nodes and coindexations. They first transform Penn trees to a dependency format and then use memory-based learning to perform various graph transformations. One of the transformations is node relabelling, which adds function tags to parser output. They report an f-score of 88.5% for the task of function tagging on correctly parsed constituents.

4 Assigning Cast3LB Function Tags to Parsed Spanish Text

The complete processing architecture of our approach is depicted in Figure 3. We describe it in detail in this and the following sections.

We divided the Spanish treebank into a training set of 80%, a development set of 10%, and a test set of 10% of all trees. We randomly assigned treebank files to these sets to ensure that different textual genres are about equally represented among the training, development and test trees.

4.1 Constituency Parsing

For constituency parsing we use Bikel's (2002) parser for which we developed a Spanish language package adapted to the Cast3LB data. Prior to parsing, we perform one of the tree transformations described by Cowan and Collins (2005), i.e. we add a CP and SBAR nodes to subordinate and relative clauses. This is undone in parser output.

The category labels in the Spanish treebank are rather fine grained and often contain redundant information.[1] We preprocess the treebank and re-

[1] For example there are several labels for Nominal Group,

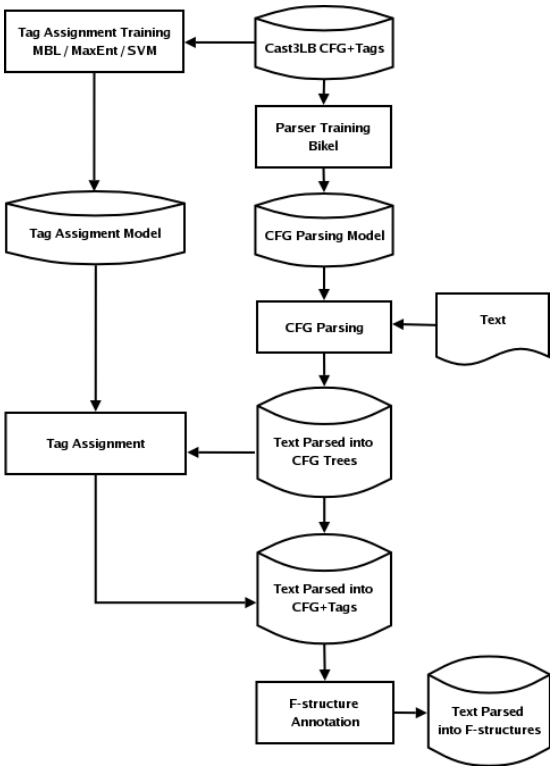

Figure 3: Processing architecture for the machine-learning-based method.

duce the number of category labels, only retaining distinctions that we deem useful for our purposes.[2]

For constituency parsing we also reduce the number of POS tags by including only selected morphological features. Table 2 provides the list of features included for the different parts of speech. In our experiments we use gold standard POS tagged development and test-set sentences as input rather than tagging text automatically.

The results of the evaluation of parsing performance on the test set are shown in Table 3. Labelled bracketing f-score for all sentences is just below 84% for all sentences, and 84.58% for sentences of length ≤ 70. In comparison, Cowan and Collins (2005) report an f-score of 85.1% (≤ 70) using a version of Collins' parser adapted for Cast3LB, and using reranking to boost performance. They use a different, more reduced category label set as well as a different training-test split. Both Cowan and Collins and the present paper report scores which ignore punctuation.

such as *grup.nom.ms* (masculine singular), *grup.nom.fs* (feminine singular), *grup.nom.mp* (masculine plural) etc. This number and gender information is already encoded in the POS tags of nouns heading these constituents.

[2]The labels we retain are the following: *INC, S, S.NF, S.NF.R, S.NF, S.R, conj.subord, coord, data, espec, gerundi, grup.nom, gv, infinitiu, interjeccio, morf, neg, numero, prep, relatiu, s.a, sa, sadv, sn, sp*, and versions of those suffixed with *.co* to indicate coordination).

Part of Speech	Features included
Determiner	type, number
Noun	type, number
Adjective	type, number
Pronoun	type, number, person
Verb	type, number, mood
Adverb	type
Conjunction	type

Table 2: Features included in POS tags. Type refers to subcategories of parts of speech such as e.g. *common* and *proper* for nouns, or *main*, *auxiliary* and *semiauxiliary* for verbs. For details see (Civit, 2000).

	LB Precision	LB Recall	F-score
All	84.18	83.74	83.96
≤ 70	84.82	84.35	84.58

Table 3: Parser performance.

4.2 Cast3LB Function Tagging

For the task of Cast3LB function tag assignment we experimented with three generic machine learning algorithms: a memory-based learner (Daelemans and van den Bosch, 2005), a maximum entropy classifier (Berger et al., 1996) and a Support Vector Machine classifier (Vapnik, 1998). For each algorithm we use the same set of features to represent nodes that are to be assigned one of the Cast3LB function tags. We use a special null tag for nodes where no Cast3LB tag is present.

In Cast3LB only nodes in certain contexts are eligible for function tags. For this reason we only consider a subset of all nodes as candidates for function tag assignment, namely those which are sisters of nodes with the category labels *gv* (Verb Group), *infinitiu* (Infinitive) and *gerundi* (Gerund). For these candidates we extract the following three types of features encoding configurational, morphological and lexical information for the target node and neighboring context nodes:

- Node features: position relative to head, head lemma, alternative head lemma (i.e. the head of NP in PP), head POS, category, definiteness, agreement with head verb, yield, human/nonhuman

- Local features: head verb, verb person, verb number, parent category
- Context features: node features (except position) of the two previous and two following sister nodes (if present).

We used cross-validation for refining the set of features and for tuning the parameters of the machine-learning algorithms. We did not use any additional automated feature-selection procedure. We made use of the following implementations: TiMBL (Daelemans et al., 2004) for Memory-Based Learning, the MaxEnt Toolkit (Le, 2004) for Maximum Entropy and LIBSVM (Chang and Lin, 2001) for Support Vector Machines. For TiMBL we used k nearest neighbors $= 7$ and the gain ratio metric for feature weighting. For MaxEnt, we used the L-BFGS parameter estimation and 110 iterations, and we regularize the model using a Gaussian prior with $\sigma^2 = 1$. For SVM we used the RBF kernel with $\gamma = 2^{-7}$ and the cost parameter $C = 32$.

5 Cast3LB Tag Assignment Evaluation

We present evaluation results on the original gold-standard trees of the test set as well as on the test-set sentences parsed by Bikel's parser. For the evaluation of Cast3LB function tagging performance on gold trees the most straightforward metric is the accuracy, or the proportion of all candidate nodes that were assigned the correct label.

However we cannot use this metric for evaluating results on the parser output. The trees output by the parser are not identical to gold standard trees due to parsing errors, and the set of candidate nodes extracted from parsed trees will not be the same as for gold trees. For this reason we use an alternative metric which is independent of tree configuration and uses only the Cast3LB function labels and positional indices of tokens in a sentence. For each function-tagged tree we first remove the punctuation tokens. Then we extract a set of tuples of the form $\langle GF, i, j \rangle$, where GF is the Cast3LB function tag and $i - j$ is the range of tokens spanned by the node annotated with this function. We use the standard measures of precision, recall and f-score to evaluate the results.

Results for the three algorithms are shown in Table 4. MBL and MaxEnt show a very similar performance, while SVM outperforms both, scoring 89.34% on accuracy and 86.87% on f-score. The learning curves for the three algorithms, shown in Figure 4, are also informative, with SVM outperforming the other two methods for all training set sizes. In particular, the last section of the plot shows SVM performing almost as well as MBL with half as much learning material.

Neither of the three curves shows signs of having reached a maximum, which indicates that in-

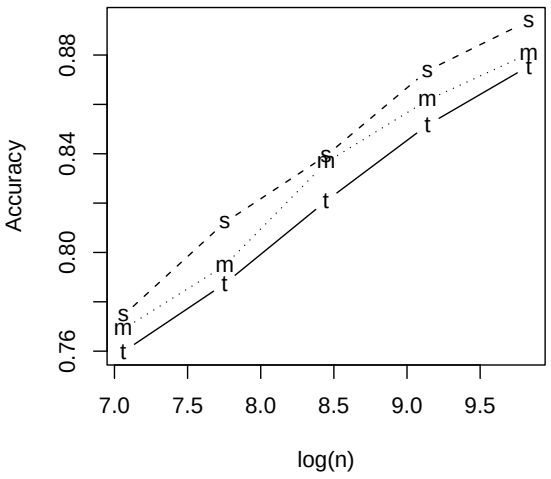

Figure 4: Learning curves for TiMBL (t), MaxEnt (m) and SVM (s).

	Acc.	Prec.	Recall	F-score
MBL	87.55	87.00	82.98	84.94
MaxEnt	88.06	87.66	86.87	85.52
SVM	**89.34**	88.93	84.90	**86.87**

Table 4: Cast3LB function tagging performance for gold-standard trees

	Precision		Recall		F-score	
	all	corr.	all	corr.	all	corr.
Baseline	59.26	72.63	60.61	75.35	59.93	73.96
MBL	64.74	78.09	64.18	78.75	64.46	78.42
MaxEnt	65.48	78.90	64.55	79.44	65.01	79.17
SVM	66.96	80.58	66.38	81.27	**66.67**	**80.92**

Table 5: Cast3LB function tagging performance for parser output, for all constituents, and for correctly parsed constituents only

Methods	p-value
Baseline vs SVM	1.169×10^{-9}
Baseline vs MBL	2.117×10^{-6}
MBL vs MaxEnt	0.0799
MaxEnt vs SVM	0.0005

Table 6: Statistical significance testing results on for the Cast3LB tag assignment on parser output.

	Precision	Recall	F-score
Baseline	73.95	70.67	72.27
SVM	76.90	74.48	75.67

Table 7: LFG F-structure evaluation results for parser output

creasing the size of the training data should result in further improvements in performance.

Table 5 shows the performance of the three methods on parser output. The baseline contains the results achieved by treating compound category-function labels as atomic during parser training so that they are included in parser output. For this task we present two sets of results: (i) for all constituents, and (ii) for correctly parsed constituents only. Again the best algorithm turns out to be SVM. It outperforms the baseline by a large margin (6.74% for all constituents).

The difference in performance for gold standard trees, and the correctly parsed constituents in parser output is rather larger than what Blaheta and Charniak report. Further analysis is needed to identify the source of this difference but we suspect that one contributing factor is the use of greater number of context features combined with a higher parse error rate in comparison to their experiments on the Penn II Treebank. Since any misanalysis of constituency structure in the vicinity of target node can have negative impact, greater reliance on context means greater susceptibility to parse errors. Another factor to consider is the fact that we trained and adjusted parameters on gold-standard trees, and the model learned may rely on features of those trees that the parser is unable to reproduce.

For the experiments on parser output (all constituents) we performed a series of sign tests in order to determine to what extent the differences in performance between the different methods are statistically significant. For each pair of methods we calculate the f-score for each sentence in the test set. For those sentences on which the scores differ (i.e. the number of trials) we calculate in how many cases the second method is better than the first (i.e. the number of successes). We then perform the test with the null hypothesis that the probability of success is chance ($= 0.5$) and the alternative hypothesis that the probability of success is greater than chance (> 0.5). The results are summarized in Table 6. Given that we perform 4 pairwise comparisons, we apply the Bonferroni correction and adjust our target $\alpha_\beta = \frac{\alpha}{4}$. For the confidence level 95% ($\alpha_\beta = 0.0125$) all pairs give statistically significant results, except for MBL vs MaxEnt.

6 Task-Based LFG Annotation Evaluation

Finally, we also evaluated the actual f-structures obtained by running the LFG-annotation algorithm on trees produced by the parser and enriched with Cast3LB function tags assigned using SVM. For this task-based evaluation we produced a gold standard consisting of f-structures corresponding to all sentences in the test set. The LFG-annotation algorithm was run on the test set trees (which contained original Cast3LB treebank function tags), and the resulting f-structures were manually corrected.

Following Crouch et al. (2002), we convert the f-structures to triples of the form $\langle GF, P_i, P_j \rangle$, where P_i is the value of the PRED attribute of the f-structure, GF is an LFG grammatical function attribute, and P_j is the value of the PRED attribute of the f-structure which is the value of the GF attribute. This is done recursively for each level of embedding in the f-structure. Attributes with atomic values are ignored for the purposes of this evaluation. The results obtained are shown in Table 7. We also performed a statistical significance test for these results, using the same method as for the Cast3LB tag assigment task. The p-value given by the sign test was 2.118×10^{-5}, comfortably below $\alpha = 1\%$.

The higher scores achieved in the LFG f-structure evaluation in comparison with the preceding Cast3LB tag assignment evaluation (Table 5) can be attributed to two main factors. Firstly, the mapping from Cast3LB tags to LFG grammatical functions is not one-to-one. For example three Cast3LB tags (CC, MOD and ET) are all mapped to LFG ADJUNCT. Thus mistagging a MOD as

	ATR	CC	CD	CI	CREG	MOD	SUJ
ATR	136	2	0	0	0	0	5
CC	6	552	12	4	25	18	6
CD	1	19	418	5	3	0	26
CI	0	6	1	50	1	0	0
CREG	0	6	0	2	43	0	0
MOD	0	0	0	0	0	19	0
SUJ	0	8	24	2	0	0	465

Table 8: Simplified confusion matrix for SVM on test-set gold-standard trees. The gold-standard Cast3LB function tags are shown in the first row, the predicted tags in the first column. So e.g. SUJ was mistagged as CD in 26 cases. Low frequency function tags as well as those rarely mispredicted have been omitted for clarity.

CC does not affect the f-structure score. On the other hand the Cast3LB CD tag can be mapped to OBJ, COMP, or XCOMP, and it can be easily decided which one is appropriate depending on the category label of the target node. Additionally many nodes which receive no function tag in Cast3LB, such as noun modifiers, are straightforwardly mapped to LFG ADJUNCT. Similarly, objects of prepositions receive the LFG OBJ function.

Secondly, the f-structure evaluation metric is less sensitive to small constituency misconfigurations: it is not necessary to correctly identify the token range spanned by a target node as long as the head (which provides the PRED attribute) is correct.

7 Error Analysis

In order to understand sources of error and determine how much room for further improvement there is, we examined the most common cases of Cast3LB function mistagging. A simplified confusion matrix with the most common Cast3LB tags is shown in Table 8. The most common mistakes occur between SUJ and CD, in both directions, and many also CREGs are erroneously tagged as CC.

7.1 Subject vs Direct Object

We noticed that in over 50% of cases when a Direct Object (CD) was misidentified as Subject (SUJ), the target node's mother was a relative clause. It turns out that in Spanish relative clauses genuine syntactic ambiguity is not uncommon. Consider the following Spanish phrase:

(1) *Sistemas que usan el 95% de*
Systems which use DET 95% of
los ordenadores.
DET computers

Its translation into English is either *Systems that use 95% of computers* or alternatively *Systems that 95% of computers use*. In Spanish, unlike in English, preverbal / postverbal position of a constituent is not a good guide to its grammatical function in this and similar contexts. Human annotators can use their world knowledge to decide on the correct semantic role of a target constituent and use it in assigning a correct grammatical function, but such information is obviously not used in our machine learning methods. Thus such mistakes seem likely to remain unresolvable in our current approach.

7.2 Prepositional Object vs Adjunct

The frequent misidentification of Prepositional Objects (CREG) as Adjuncts (CC) seen in Table 8 can be accounted for by several factors. Firstly, Prepositional Objects are strongly dependent on specific verbs and the comparatively small size of our training data means that there is limited opportunity for a machine-learning algorithm to learn low-frequency lexical dependencies. Here the obvious solution is to use a more adequate amount of training material when it becomes available.

A further problem with the Prepositional Object - Adjunct distinction is its inherent fuzziness. Because of this, treebank designers may fail to provide easy-to-follow, clearcut guidelines and human annotators necessarily exercise a certain degree of arbitrariness in assigning one or the other function.

8 Conclusions and Future Research

Our research has shown that machine-learning-based Cast3LB tag assignment as a post-processing step to raw tree parser output statistically significantly outperforms a baseline where the parser itself is trained to learn category / Cast3LB-function pairs. In contrast to the parser-based method, the machine-learning-based method avoids some sparse data problems and allows for more control over Cast3LB tag assignment. We have found that the SVM algorithm outperforms the other two machine learning methods used.

In addition, we evaluated Cast3LB tag assignment in a task-based setting in the context of automatically acquiring LFG resources for Spanish from Cast3LB. Machine-learning-based Cast3LB tag assignment yields statistically-significantly improved LFG f-structures compared to parser-based assignment.

One limitation of our method is the fact that it treats the classification task separately for each target node. It thus fails to observe constraints on the possible sequences of grammatical function tags in the same local context. Some functions are unique, such as the Subject, whereas others (Direct and Indirect Object) can only be realized by a full NP once, although they can be doubled by a clitic pronoun. Capturing such global constraints will need further work.

Acknowledgements

We gratefully acknowledge support from Science Foundation Ireland grant 04/IN/I527 for the research reported in this paper.

References

A. L. Berger, V. J. Della Pietra, and S. A. Della Pietra. 1996. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39–71, March.

D. Bikel. 2002. Design of a multi-lingual, parallel-processing statistical parsing engine. In *Human Language Technology Conference (HLT)*, San Diego, CA, USA. Software available at `http://www.cis.upenn.edu/~dbikel/software.html#stat-parser`.

D. Blaheta and E. Charniak. 2000. Assigning function tags to parsed text. In *Proceedings of the 1st Conference of the North American Chapter of the ACL*, pages 234–240, Rochester, NY, USA.

J. Bresnan. 2001. *Lexical-Functional Syntax*. Blackwell Publishers, Oxford.

M. Burke, O. Lam, A. Cahill, R. Chan, R. O'Donovan, A. Bodomo, J. van Genabith, and A. Way. 2004. Treebank-based acquisition of a Chinese Lexical-Functional Grammar. In *Proceedings of the 18th Pacific Asia Conference on Language, Information and Computation (PACLIC-18)*.

A. Cahill, M. Forst, M. McCarthy, R. O'Donovan, and C. Roher. 2003. Treebank-based multilingual unification-grammar development. In *Proceedings of the 15th Workshop on Ideas and Strategies for Multilingual Grammar Development, ESSLLI 15*, Vienna, Austria.

A. Cahill, M. Burke, R. O'Donovan, J. van Genabith, and A. Way. 2004. Long-distance dependency resolution in automatically acquired wide-coverage PCFG-based LFG approximations. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*, pages 319–326, Barcelona, Spain.

Chih-Chung Chang and Chih-Jen Lin, 2001. *LIBSVM: a library for support vector machines*. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

M. Civit and M. A. Martí. 2004. Building Cast3LB: A Spanish treebank. *Research on Language and Computation*, 2(4):549–574, December.

M. Civit. 2000. Guía para la anotación morfosintáctica del corpus CLiC-TALP, X-TRACT Working Paper. Technical report. Available at `http://clic.fil.ub.es/personal/civit/PUBLICA/guia_morfol.ps`.

M. Civit. 2004. Guía para la anotación de las funciones sintácticas de Cast3LB. Technical report. Available at `http://clic.fil.ub.es/personal/civit/PUBLICA/funcions.pdf`.

B. Cowan and M. Collins. 2005. Morphology and reranking for the statistical parsing of Spanish. In *Conference on Empirical Methods in Natural Language Processing*, Vancouver, B.C., Canada.

R. Crouch, R. M. Kaplan, T. H. King, and S. Riezler. 2002. A comparison of evaluation metrics for a broad-coverage stochastic parser. In *Conference on Language Resources and Evaluation (LREC 02)*.

W. Daelemans and A. van den Bosch. 2005. *Memory-Based Language Processing*. Cambridge University Press, September.

W. Daelemans, J. Zavrel, K. van der Sloot, and A. van den Bosch. 2004. TiMBL: Tilburg Memory Based Learner, version 5.1, Reference Guide. Technical report. Available from `http://ilk.uvt.nl/downloads/pub/papers/ilk0402.pdf`.

V. Jijkoun and M. de Rijke. 2004. Enriching the output of a parser using memory-based learning. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Barcelona, Spain.

Zh. Le, 2004. *Maximum Entropy Modeling Toolkit for Python and C++*. Available at `http://homepages.inf.ed.ac.uk/s0450736/software/maxent/manual.pdf`.

R. O'Donovan, A. Cahill, J. van Genabith, and A. Way. 2005. Automatic acquisition of Spanish LFG resources from the CAST3LB treebank. In *Proceedings of the Tenth International Conference on LFG*, Bergen, Norway.

V. N. Vapnik. 1998. *Statistical Learning Theory*. Wiley-Interscience, September.