IMPROVING TREEBANK-BASED AUTOMATIC LFG INDUCTION FOR SPANISH

Grzegorz Chrupała and Josef van Genabith
National Centre for Language Technology and School of Computing
Dublin City University

Proceedings of the LFG06 Conference
Universität Konstanz
Miriam Butt and Tracy Holloway King (Editors)
2006
CSLI Publications
http://csli-publications.stanford.edu/

Abstract

We describe several improvements to the method of treebank-based LFG induction for Spanish from the Cast3LB treebank (O'Donovan et al., 2005). We discuss the different categories of problems encountered and present the solutions adopted. Some of the problems involve a simple adoption of existing linguistic analyses, as in our treatment of clitic doubling and null subjects. In other cases there is no standard LFG account for the phenomenon we wish to model and we adopt a compromise, conservative solution. This is exemplified by our treatment of Spanish periphrastic constructions. In yet another case, the less configurational nature of Spanish means that the LFG annotation algorithm has to rely mostly on Cast3LB function tags, and consequently a reliable method of adding those tags to parse trees had to be developed. This method achieves over 6% improvement over the baseline for the Cast3LB-function-tag assignment task, and over 3% improvement over the baseline for LFG f-structure construction from function-tag-enriched trees.

1 Introduction

The research reported in this paper has been carried out as part of the GramLab project whose goal is to acquire multilingual wide coverage LFG resources from treebanks for several languages. We report on the ongoing work in LFG induction for Spanish.

Inducing deep syntactic analyses from treebank data avoids the cost and time involved in manually creating wide-coverage resources.

LFG f-structures provide a level of syntactic representation which is more abstract and cross-linguistically more uniform than constituency trees. F-structures include explicit encodings of phenomena such as control and raising, pro-drop and long distance dependencies: those characteristics make this level a suitable representation for many NLP applications such as transfer-based Machine Translation or Question Answering.

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). The f-structure annotation algorithm used for inducing LFG resources from the Penn-II treebank for English uses configurational, categorial, function tag and trace information.

Preliminary research on Spanish LFG induction was carried out by (O'Donovan et al., 2005). In the present paper we discuss several issues which became obvious while trying to expand the coverage of Spanish grammatical constructions and phenomena and while dealing with the peculiarities of the treebank that we are using. The problems arising from adapting a grammar acquisition methodology developed for one language/treebank to another language/treebank combination fall into three broad categories:

- New phenomena and constructions, successfully treated within standard LFG: clitic doubling, null subjects
- New phenomena and constructions, problematic within standard LFG: clitic climbing (i.e. complex predicates)
- Limitations of previous approach due to language/treebank specific assumptions which no longer hold: flexible constituent order and less configurational c-structures

2 Clitic doubling and null subjects

In Spanish pronominal clitics for Direct and Indirect Object can co-occur with non-clitic (full NP) objects.[1] Example 1 shows clitic doubling with Indirect Object, Example 2 with Direct Object. The non-clitic Objects are in italics; the co-occurring clitics are in bold. The clitics agree with the non-clitic arguments in person, number, gender and case.

(1) Algo parecido **les** sucede *a los hombres.*
something similar them occurs to DEF men
Something similar happens to men.

(2) Cada cual **lo** comprende *eso* a su manera.
every which it understands this to POSS manner
Everyone understands this in their own way.

Clitic doubling is quite common with Indirect Objects: in our treebank data in 23% of the cases where there is a non-pronominal Indirect Object it co-occurs with a pronominal clitic. Clitic doubling for Direct Objects is more constrained, but still relatively common at 1% of corpus occurences of non-pronominal Direct Objects.

In clitic doubling constructions, pronominal clitics should not introduce a PRED value, as that would clash with the one introduced by the non-clitic Object. However when clitics are not accompanied by non-clitic Objects, they should introduce PRED = ‘pro’, in order to satisfy the verb’s subcategorization requirements.

[1] This phenomenon is subject to complex, dialect-dependent constraints involving animacy, specicity and information structure, especially for Direct Object. Currently we do not try to model these constraints fully.

We achieve this effect by means of optional equations, as is standard practice in LFG. Example 3 below illustrates the equations associated with the dative *le* (Indirect Object).

(3) *le* **pp3csd00**

((↑ PRED) = 'pro')
((↑ PRON-TYPE) = PERS)
((↑ PRON-FORM) = el)
(↑ CASE) = DAT
(↑ NUM) = SG
(↑ PERS) = 3

An optional equation (*e*) is a disjunction of *e* and *true*. In standard LFG the correct disjunct is chosen as follows: in a clitic-doubling context, the first disjunct is excluded because the PRED value it introduces clashes with the one introduced by the non-clitic Object, and thus the *true* disjunct applies. In non-doubling contexts, the first disjunct applies successfully, while if the second one applies, the resulting f-structure does not satisfy completeness because of the missing PRED value.

In our implementation we do not check for completeness because our PRED values lack subcategorization frames,[2] so we use a slightly different definition of optionality. An optional equation works more like a default equation: the optional equation $((f \;\; a) = v)$ holding of f-structure f is interpreted as a disjunction of the existential constraint $(f \;\; a)$ and the equation $(f \;\; a) = v$. In the clitic-doubling case the second disjunct (which introduces the PRED value) only applies if the PRED value has not been contributed by some other equation.

Another area where we use optional equations is in our treatment of null subjects (pro-drop). In Spanish explicit subjects are often absent. Subject features such as person and number are encoded in agreement morphology on the verb instead. When there is no overt subject, the PRED value that is needed to satisfy the verb's subcategorization is introduced by the inflected verb-form.

All finite verb preterminals optionally introduce a 'pro' subject. Example 4 below illustrates the annotation associated with the inflected verb form *vió* (see-3SG).

(4) *vió* **vmis3s0**

(↑ PRED)= 'ver'
((↑ PRED SUBJ) = 'pro')
(↑ SUBJ NUM) = SG

[2] The subcat frames are acquired separately in our architecture. See (O'Donovan et al., 2004).

$(\uparrow \text{SUBJ PERS}) = 3$
$(\uparrow \text{SUBJ TENSE}) = \text{PAST}$
$(\uparrow \text{SUBJ MOOD}) = \text{INDICATIVE}$
$(\uparrow \text{LIGHT}) = -$

Currently all finite verb forms receive an optional PRED equation. This is not entirely adequate as at least one Spanish verb *haber* (existential be) can never co-occur with an overt subject, so ideally it should receive an obligatory PRED equation. Similarly, weather verbs are normally ungrammatical with explicit subjects (Example 5 a and b). Exceptionally they can take modified cognate subjects (Example 5 c).

(5) (a) * Llovió lluvia.
rained rain

(b) * La lluvia llovió.
the rain rained

(c) Llovió una lluvia fina pero persistente.
rained a rain light but persistent
"A light but persistent rain rained down."

Whether it is possible to learn from treebank data which verbs do not allow overt subjects and under what conditions remains an open question for future investigation.

Our use of optionality in the treatment of Spanish clitic doubling and null subjects illustrates language-specific problems that arise for LFG induction, but for which there are standard solutions in the LFG framework. Those solutions can be adopted and adapted for our data-driven approach to grammar acquisition. They may require additional implementation effort (in this case adding appropriate optionality support to the constraint solver), but otherwise they can be easily accommodated within the existing methodology.

In the following section we discuss a phenomenon which is more problematic: it does not have a widely agreed-upon solution in standard LFG and thus is an issue in any computational implementation including our own.

3 Periphrastic constructions

In Spanish periphrastic constructions, such as in Example 6 a, verbal pronominal clitics which are understood as arguments of the "lower" verb can attach to the "higher" verb. This phenomenon, called clitic climbing, is only grammatical with certain verbs. Others do not admit it, as illustrated in Example 6 b. The verbs that do admit clitic climbing are sometimes called *light* verbs.

(6) (a) **La** puedo *ver*. Puedo *ver***la**.
her can-1SG see can-1SG see-her

(b) * **La** insistí en *ver*. Insistí en *ver***la**.
her insisted-1SG in see insisted-1SG in see-her

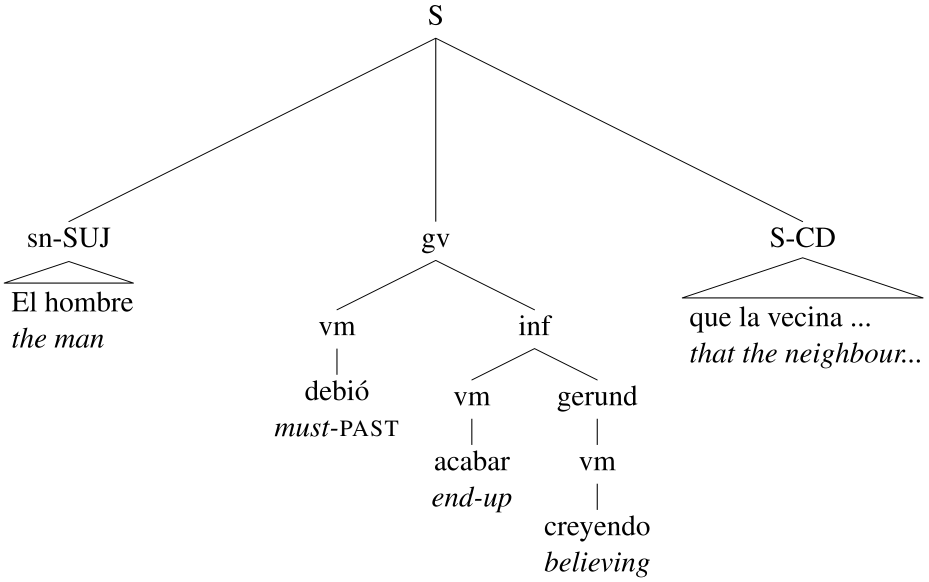
Normally only the clitic climbing versions of periphrastic constructions present difficulties for an LFG account due to the mismatch of the position of arguments in the tree and where they should end up in the f-structure. However, the configuration adopted for periphrastic constructions in Cast3LB generalizes this problematic mismatch to all contexts.

As illustrated in Figure 1, all verbs participating in the periphrastic construction are under the *gv* (Verb Group) node, with the argument of the lowest verb being attached as sister to the *gv*. This example also illustrates that periphrastic constructions can be combined with each other, so in principle the lowest non-light verb could be nested a number of levels deep.

There are several proposals of how to deal with periphrastic constructions with clitic climbing within LFG. Both (Alsina, 1997) and (Butt, 1997) propose a predicate composition analysis. As in standard LFG PRED values can never unify, this approach requires modifications to the unification operation. In (Andrews and Manning, 1999) the authors propose an even more radical departure from standard LFG and replace the projection architecture with *differential information spreading* within the f-structure.

As there seems to be no consensus as to the best treatment of Romance constructions involving light verbs, we decided in favor of a conservative approach which avoids non-standard extensions to the LFG formalism. We use functional uncertainty and a nested XCOMP configuration in our treatment of periphrastic constructions. The mechanism is illustrated in Figure 2. The *inf(initive)* and *gerund* daughters of the *gv* node constrain the f-structure corresponding to their mother nodes to be LIGHT $+$, and introduce their own f-structure as the value of XCOMP attribute.

Non-subject sisters of the *gv* are annotated with functional uncertainty equations which specify that their f-structure is the value of the GF attribute arbitrarily embedded in a series of XCOMPs. There is an off-path constraint that specifies that the f-structure containing each of the XCOMPs in the path has to be LIGHT $+$. Another off-path constraint on the f-structure containing the final GF restricts it to be LIGHT $-$. Together those annotations ensure that arguments are always



$$\begin{bmatrix} \textsc{subj} & \begin{bmatrix}\text{“el hombre”}\end{bmatrix}\boxed{1} \\ \textsc{pred} & \text{‘}\textsc{deber}\text{’} \\ \textsc{tense} & \textsc{past} \\ \textsc{light} & + \\ \textsc{xcomp} & \begin{bmatrix} \textsc{subj} & \boxed{1} \\ \textsc{pred} & \text{‘}\textsc{acabar}\text{’} \\ \textsc{light} & + \\ \textsc{xcomp} & \begin{bmatrix} \textsc{subj} & \boxed{1} \\ \textsc{pred} & \text{‘}\textsc{creer}\text{’} \\ \textsc{light} & - \\ \textsc{comp} & \begin{bmatrix}\text{“que la vecina ...”}\end{bmatrix} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

Figure 1: Periphrastic construction with two light verbs: The treebank tree, and the f-structure produced

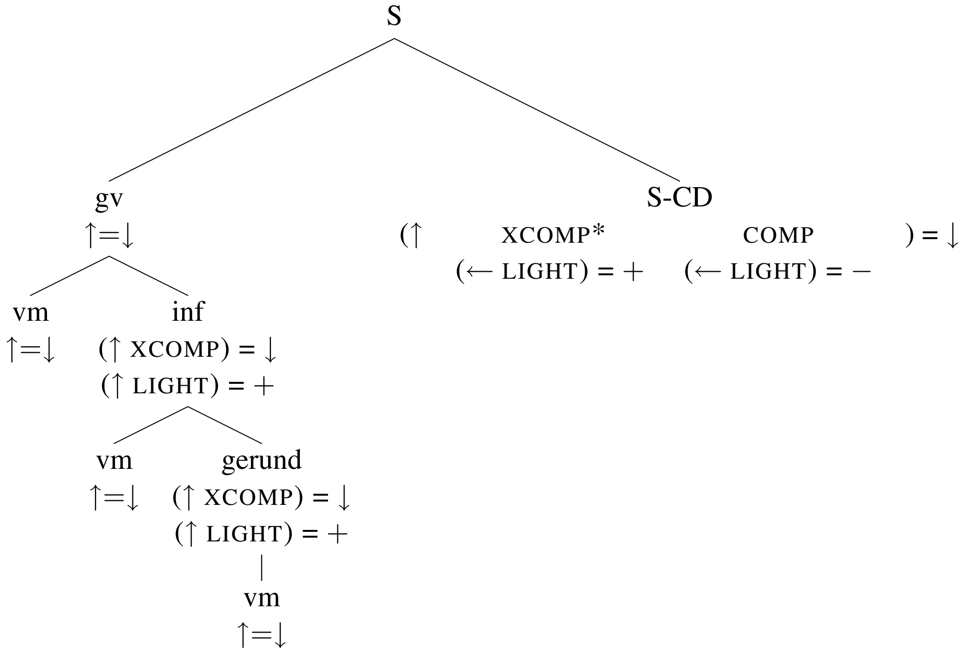


Figure 2: Treatment of periphrastic constructions by means of functional uncertainty equations with off-path constraints

attached to the lowest (non-light) verb. This is the correct analysis for the majority of periphrastic constructions.[3]

Our treatment of periphrastic constructions is not entirely satisfactory: it is a compromise solution. From a descriptive perspective it does not perfectly model the linguistic phenomena in question. Our motivation for using it is that it allows us to avoid implementing a solution which departs too far from the standard LFG formalism and for which there is no consensus among theoretical linguists.

The XCOMP-based treatment is adequate in the vast majority of cases and has the advantage that the resulting f-structure parallels the analysis that would be used in languages with no clitic climbing (such as English) for similar sentences. This could potentially be useful if our LFG resources are to be used in multilingual applications.

[3] One exception are causative constructions, where, if one insists on an XCOMP-type treatment, the causee should be the argument of the causative verb, whereas the other arguments should depend on the verb expressing the event caused (Alsina, 1997).

In the following section we discuss the particular features of our language and treebank which challenge some of the assumptions made in the design of the LFG acquisition architecture initially developed using the English Penn Treebank data.

4 Constituent order and configurationality

The method of automatic LFG induction was initially developed using the English Penn-II Treebank data. The idea behind the annotation rules is that limited configurational and categorial information should in most cases be sufficient to determine a constituent's grammatical function in the sentence: as evidenced by the good results of this approach for English, this assumption is borne out for this language. It turns out that the approach is more problematic for our Spanish Cast3LB data. Spanish allows much more variation and flexibility in major sentence constituent order than does English. Partly as a consequence of this flexibility, the treebank encoding of syntactic structure also has to be different than in the Penn Treebank.

Although the canonical word order for Spanish is SVO, in Cast3LB there are about 20% post-verbal subjects, and about 11% preverbal non-clitic direct objects. Thus the information on position relative to the verb is not a reliable predictor of grammatical function in Spanish.

Accordingly, the Spanish treebank makes extensive use of function tags to make the grammatical function of constituents more explicit. Although there are also functional tags in the Penn Treebank, their use is less necessary. In the Penn Treebank, configuration information alone is often sufficient to determine grammatical function: e.g.: left sister to *VP* is typically a Subject while right daughter to *V* is an Object.

Due to the preceding considerations the Spanish annotation algorithm has to rely on function tags much more heavily than is the case for English. It is thus important to be able to enrich parser-output trees with those tags as reliably as possible.

The initial implementation described in (O'Donovan et al., 2005) relied on the parser itself to obtain function-tagged parse trees. Bikel's parser (Bikel, 2002) was trained on trees where function tags were simply part of the category label, so instead of having one non-terminal category *sn* (Noun Phrase) there are several different NP categories e.g. *sn*-SUJ, *sn*-CD, *sn*-CI, etc. We treated this simple method as a baseline and tried to determine how much we could improve on it.

We decided to let the parser learn and output plain constituency trees and add Cast3LB function tags in a postprocessing step. The intuition behind adopting this approach is that we thus avoid the multiplication of categories (which could potentially lead to a sparse-data-related decline in performance), and also achieve better control over the learning method and the feature set used than if we just rely on the parser.

Our method and evaluation results are described in detail in (Chrupała and van Genabith, 2006). Here we present a brief outline of this research and elaborate on some LFG-relevant aspects. Although our work is the first attempt to learn the assignment of Cast3LB function tags to parser output for Spanish, there is some existing research on enriching parse trees with Penn function tags for English (Blaheta and Charniak, 2000; Jijkoun and de Rijke, 2004). The general idea is the same in each case: function tags are added to parse tree nodes in a postprocessing step, and the assignment model is learned from treebank data.

In our research we experimented with three machine-learning methods: Memory-Based, Maximum Entropy and Support Vector Machines. The best performance was obtained with SVM and those are the results that we report below.

We treat Cast3LB function tag assignment as a classification task. Our training examples are candidate nodes in treebank trees. We treat as candidate nodes all those that are sisters to either

- *gv* (Verb Group)
- *infinitiu* (Infinitive)
- *gerundi* (Gerund)

The class label assigned to each example is its Cast3LB function tag, or the label NULL if no function tag is present.

For each example node we extract a set of features which are used by the machine-learning algorithm to build the model used to classify unseen examples. Figure 3 illustrates the features extracted from an example tree. The *focus node features* are extracted from the node labeled *sn*-SUJ. The other three nodes provide *context node features*, and the nodes included in the oval area (the head node and the mother node) are used to extract *local features*. The features encode categorial, configurational, morphological and lexical information that we considered relevant for determining functions encoded in the Cast3LB function tags:

- 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 (i.e. number of terminals dominated), 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).

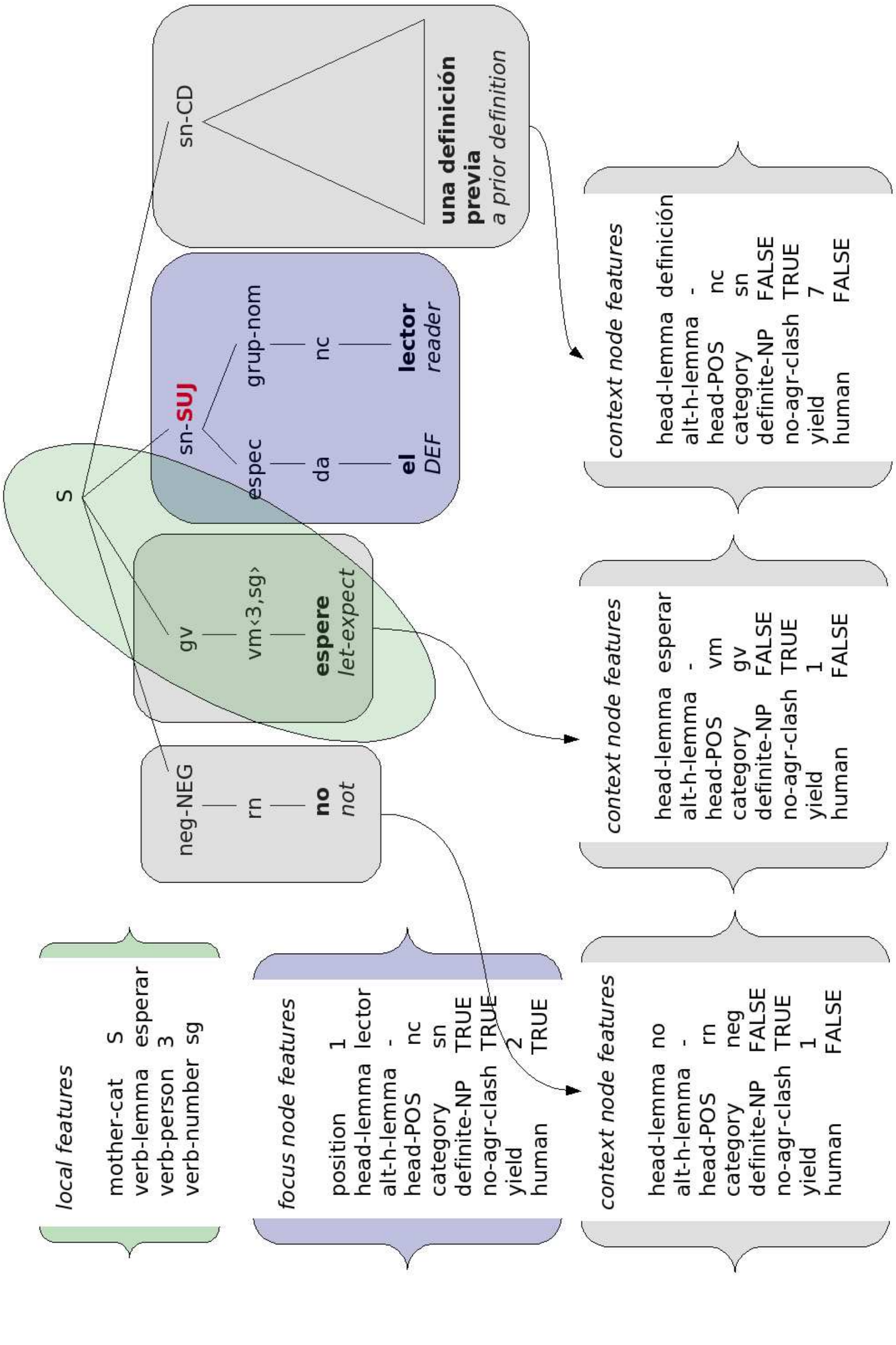


Figure 3: Examples of features extracted from an example node

	Acc.	Prec.	Recall	F-score
SVM	**89.34**	88.93	84.90	**86.87**

Table 1: 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
SVM	66.96	80.58	66.38	81.27	**66.67**	**80.92**

Table 2: Cast3LB function tagging performance for parser output

In order to evaluate the performance of the trained classifier we used the following procedure: for each function-tagged tree we first remove the punctuation tokens. Then we extract a set of tuples of the form $\langle \text{GF}, i, j \rangle$, where GF is the Cast3LB function tag and $i - j$ is the range of tokens spanned by the node annotated with this function. For example from the tree in Figure 3 the following set of tuples would be obtained: $\{\langle \text{NEG}, 1, 1\rangle, \langle \text{SUJ}, 3, 4\rangle, \langle \text{CD}, 5, 7\rangle\}$. We use the standard measures of precision, recall and f-score to evaluate those sets of tuples against the ones extracted from the reference gold-standard trees.

Tables 1 and 2 contain the results of Cast3LB function tag assignment evaluation for gold trees (taken from the treebank) and for trees output by Bikel's parser. For parser trees we report the result for all nodes (all), and for the subset of nodes that were correctly bracketed (corr).

The results for parse trees, even for the correctly bracketed node subset, are still lower than for gold trees. We suspect this may be due to the fact that even for correctly bracketed nodes, the context may still contain incorrectly parsed structures. An additional consideration is the fact that we extract training data from treebank trees: perhaps an improvement can be obtained by using parsed trees for training data. We are currently experimenting with this idea.

From the perspective of LFG induction, any improvements in the Cast3LB function tag assignment task are only useful if they translate to better quality f-structures. The mapping from Cast3LB tags to LFG annotations is reasonably straightforward, but not bijective (Table 3 contains the Cast3LB function tags and specifies their correspondence to LFG features). Also LFG function tags are only available for daughters of *S* nodes. For other nodes, the annotation algorithm has

Tag	Meaning	LFG attribute
ATR	Attribute of copular verb	PREDLINK
CAG	Agent of passive verb	OBL_{ag}
CC	Compl. of circumstance	ADJUNCT
CD	Direct object	COMP for finite *S* nodes, XCOMP for non-finite *S* nodes OBJ otherwise
CD.Q	Direct object of quantity	OBJ
CI	Indirect object	OBJ2
CPRED	Predicative complement	PREDLINK
CPRED.CD	Predicative of Direct Object	PREDLINK
CPRED.SUJ	Predicative of Subject	PREDLINK
CREG	Prepositional object	OBL
ET	Textual element	ADJUNCT
IMPERS	Impersonal marker	IMPERS
MOD	Verbal modifier	ADJUNCT
NEG	Negation	NEG
PASS	Passive marker	PASSIVE
SUJ	Subject	SUBJ
VOC	Vocative	ADJUNCT

Table 3: Cast3LB function tags and corresponding LFG f-structure attributes

to rely on other evidence to come up with the correct LFG annotations.

Given those complications we compared the quality of the f-structures produced using our improved function tags against the baseline. The results of the evaluation of the f-structures produced by the two methods are given in Table 2. The difference in f-scores is smaller than in the case of Cast3LB tag assignment. This is most likely due to two facts:

- Tags are available and used for only a subset of nodes
- F-structure evaluation is less sensitive to some forms of incorrect parse trees, i.e. exact constituent boundaries are not important, only correct bracketing of heads.

We also performed a statistical significance test for these results. For each pair of methods we calculate the f-score for each sentence in the test set. For those

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

Table 4: F-structure evaluation results for parser output

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 p-value given by the sign test was 2.118×10^{-5}: thus the improvement is statistically significant at a confidence level of 99%.

5 Conclusions and further work

We have discussed several issues which arose while adapting an automatic treebank-based LFG acquisition method developed originally for the Penn Treebank to the Spanish Cast3LB treebank. The process of porting our method to Spanish (as well as other languages we deal with within the GramLab project) has made it more obvious what are the strengths and weaknesses of our approach.

The less configurational nature of the Cast3LB data made it necessary for the LFG annotation algorithm to rely heavily on function tags, and consequently to develop better methods of obtaining function-tagged parse trees. This improved machine-learning postprocessing method is now also successfully being used for English. Thus expanding the coverage of our method to multiple languages and treebanks also benefits LFG induction for English.

Areas of current and future research include revising the LFG account of some areas of Spanish syntax:

- Replacing COMP with OBJ
- Changing the PREDLINK analysis to one which better reflects the difference between predicative complements of Direct Object vs. of Subject

We also plan to further expand grammar coverage to more kinds of constructions and linguistic phenomena.

In the area of function-tag assignment we believe there is also room for further improvement. Extracting training examples from parse trees rather than treebank trees should lead to better performance on parser output. Trying to constrain function tag sequences to avoid impossible combinations (such as two SUJ tags) would also be desirable.

Acknowledgements

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

References

Alsina, A. (1997). A theory of complex predicates: evidence from causatives in Bantu and Romance. In Alsina, A., Bresnan, J., and Sells, P., editors, *Complex Predicates*, pages 203–246. Center for the Study of Language and Information, Stanford, CA, USA.

Andrews, A. D. and Manning, C. D. (1999). *Complex Predicates and Information Spreading in LFG*. Center for the Study of Language and Information, Stanford, CA, USA.

Bikel, D. (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`.

Blaheta, D. and Charniak, E. (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.

Butt, M. (1997). Complex predicates in Urdu. In Alsina, A., Bresnan, J., and Sells, P., editors, *Complex Predicates*. Center for the Study of Language and Information, Stanford, CA, USA.

Cahill, A., Burke, M., O'Donovan, R., van Genabith, J., and Way, A. (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.

Chrupała, G. and van Genabith, J. (2006). Using machine-learning to assign function labels to parser output for Spanish. In *Proceedings of the COLING/ACL*

2006 Main Conference Poster Sessions, pages 136–143, Sydney, Australia. Association for Computational Linguistics.

Jijkoun, V. and de Rijke, M. (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.

O'Donovan, R., Burke, M., Cahill, A., van Genabith, J., and Way, A. (2004). Large-scale induction and evaluation of lexical resources from the Penn-II Treebank. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*, pages 367–374, Barcelona, Spain.

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