AUTOMATIC ACQUISITION OF SPANISH LFG RESOURCES FROM THE CAST3LB TREEBANK

Ruth O’Donovan, Aoife Cahill, Josef van Genabith and Andy Way

Proceedings of the LFG05 Conference

University of Bergen

Miriam Butt and Tracy Holloway King (Editors)

2005

CSLI Publications

http://csli-publications.stanford.edu/

— Abstract —

In this paper, we describe the automatic annotation of the Cast3LB Treebank with LFG f-structures for the subsequent extraction of Spanish probabilistic grammar and lexical resources. We adapt the approach and methodology of Cahill *et al.* (2004), O'Donovan *et al.* (2004) and elsewhere for English to Spanish and the Cast3LB treebank encoding. We report on the quality and coverage of the automatic f-structure annotation. Following the pipeline and integrated models of Cahill *et al.* (2004), we extract wide-coverage probabilistic LFG approximations and parse unseen Spanish text into f-structures. We also extend Bikel's (2002) Multilingual Parse Engine to include a Spanish language module. Using the retrained Bikel parser in the pipeline model gives the best results against a manually constructed gold standard (73.20% preds-only f-score). We also extract Spanish lexical resources: 4090 semantic form types with 98 frame types. Subcategorised prepositions and particles are included in the frames.

1 Introduction

Manual construction of rich grammatical and lexical resources, particularly multilingual resources, is time-consuming, expensive and requires considerable linguistic and computational expertise. Previously in (Cahill et al., 2004) and (O'Donovan et al., 2004), we outlined an approach which exploits information encoded in treebank trees to automatically annotate each node in each tree with f-structure equations representing abstract predicate-argument structure relations. From the annotated treebank, we automatically extract large-scale unification grammar resources, namely probabilistic approximations of LFGs[1], and subcategorisation information, for parsing new text into f-structures. A growing number of treebanks for languages other than English (including Japanese, Chinese, German, French, Czech and Spanish) are becoming available. Cahill *et al.* (2003) and Burke *et al.* (2004) show how the lexical and grammatical extraction approaches described in (Cahill et al., 2004) and (O'Donovan et al., 2004) for English can be successfully migrated to typologically different languages (German and Chinese) and different treebank encodings (TIGER (Brants et al., 2002) and Penn CTB (Xue, Chiou, and Palmer, 2002)). Here we describe the porting of the methodology to Spanish and the Cast3LB Treebank (Civit, 2003). We present an f-structure annotation algorithm for Cast3LB and describe how LFG grammars for Spanish can be induced from the f-structure-annotated treebank. We extract PCFG-based LFG approximations and report on a number of parsing experiments. We evaluate both the quality of the automatic f-structure annotation of the Cast3LB treebank, and the parser output. Finally, we describe how lexical resources can be extracted from the f-structure-annotated treebank and present sample lexical entries.

[1] See (Cahill et al., 2004) and (O'Donovan et al., 2004) for details on how these resources differ from traditional LFGs.

2 From Cast3LB to a Spanish LFG

2.1 Cast3LB Treebank

The Cast3LB treebank (Civit, 2003) consists of 125,000 words (approximately 3,500 trees) taken from a wide variety of Spanish texts (journalistic, literary, scientific) from both Spain and South America. Despite the free word order of Spanish, constituency rather then dependency annotation is used in the Cast3LB treebank. Unlike the Penn-II Treebank which loosely complies with X-bar theory, the phrase-structure trees of the Spanish Treebank are essentially theory neutral. Only lexically realised constituents are annotated with the exception of elided subjects in pro-drop constructions. There are therefore no empty nodes and traces unlike in the Penn-II Treebank. Another policy of the Cast3LB creators was not to alter the surface word order of the constituents. Due to the free word order of Spanish, a verb phrase containing the verb and its arguments (other than subject) cannot always be established. As a result the main constituents of the sentence are daughters of the root node. The free word order of Spanish also means that phrase-structural position is not an indication of grammatical function, a feature of English which was heavily exploited in the automatic annotation of the Penn-II Treebank. Instead we take advantage of the rich Cast3LB functional annotation of verbal dependents and the fine-grained non-terminals to annotate the treebank with f-structure equations.

Figure 1 shows an example tree from the Cast3LB Treebank. The verbal elements of the sentence are realised by the gv (grupo verbal) subtree. The sn (sintagma nominal) subject of the sentence is marked as such using the functional tag SUJ. Any other verbal complements and adjuncts are marked in a similar way in the treebank. The full list of functional labels is provided in Table 1. Constituents which are not verbal complements do not receive functional annotations. The full list of phrasal category labels (i.e. excluding preterminals) is presented in Table 2. In addition to these, any of the clausal nodes may be annotated with an asterisk to indicate verbal ellipsis in coordinated structures. The tree in Figure 2 where the verb *es* is omitted from the second conjunct demonstrates this phenomenon. The preterminal tags in Cast3LB are fine-grained (see Figures 1 and 2) because they encode morphological as well as part of speech (POS) information. For example the tag ncms000 indicates that *recurso* is a common noun which is masculine and singular. While there are some distinctions beyond POS encoded in the Penn-II tags, the limited inflectional morphology of English does not allow for or require the same level of detail as Spanish. In Penn-II there are just six verbal tags (excluding the modal tag) which suffice for English inflection. As a single Spanish verb morpheme carries information about person, number, tense, aspect and mood, the 147 verbal tags are by necessity considerably more complex.

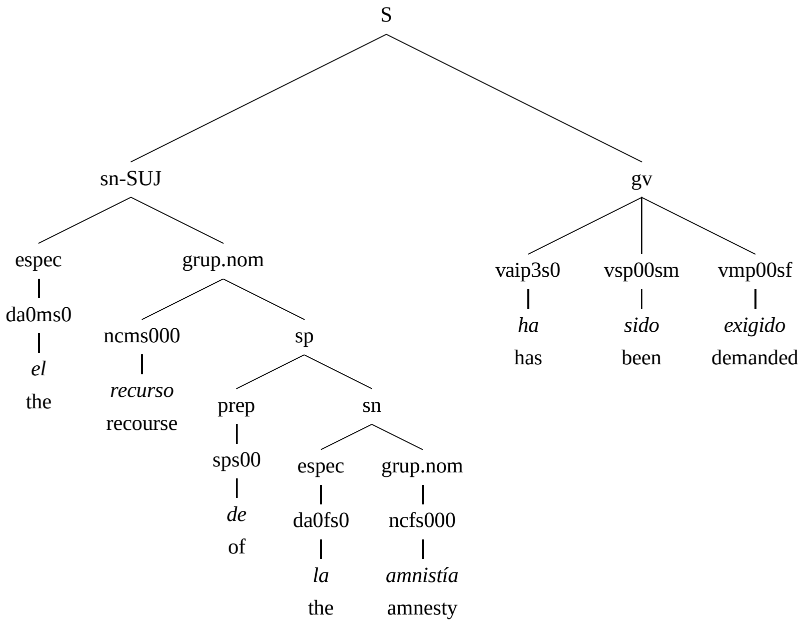


Figure 1: Example Tree from the Cast3LB Treebank

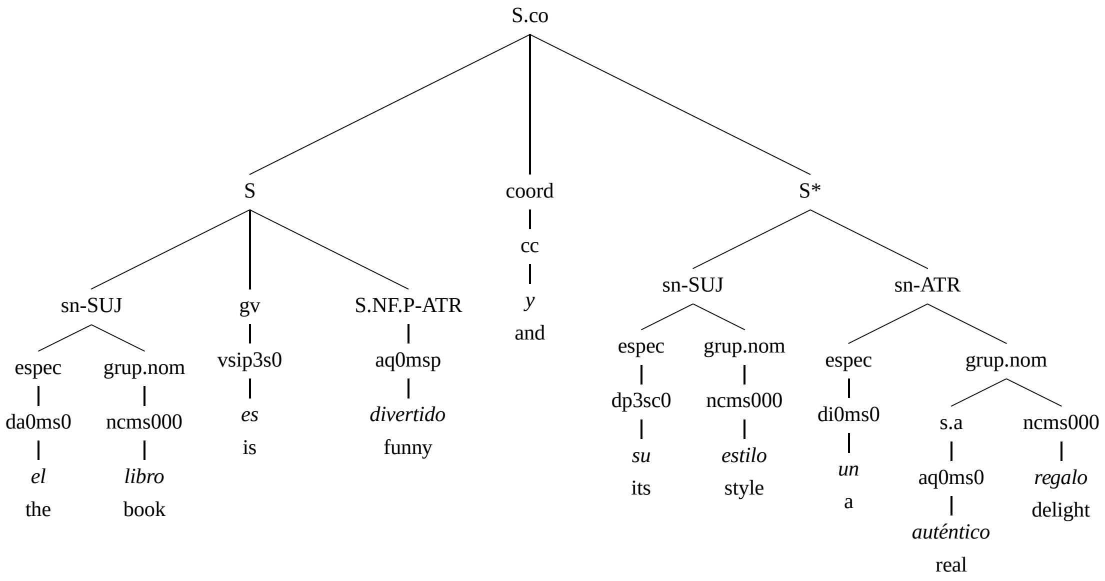


Figure 2: Cast3LB Annotation of Verbal Ellipsis in Coordinated Constructions

SUJ	Subject
CD	Direct Complement
CI	Indirect Complement
ATR	Attributive
CPRED	Predicative Complement
CAG	Agentive Complement
CREG	Prepositional Phrase Complement
CC	Adjunct
ET	Textual Element
MOD	Modal Adverb
NEG	Negative
PASS	Passive
IMPERS	Impersonal
VOC	Vocative

Table 1: Functional Annotations used in the Cast3LB Treebank

2.2 Automatic Annotation of Cast3LB Trees

The annotation algorithm for Spanish is constructed following the same methodology used for English, German and Chinese. We begin by automatically extracting all the rules and their associated frequencies from the treebank. We extract 7972 rules when we conflate preterminals containing morphological information to basic POS tags.[2] We then select the most frequent rule types for each left hand side (lhs) category which together give 85% coverage of all rule tokens expanding that category. This results in a reduced set of 3638 rules. The right hand sides (rhs) of these 3638 rules are then automatically assigned default annotations, e.g. any node with a SUJ functional annotation is assigned the functional equation $\uparrow\text{SUBJ}=\downarrow$. The rules are also head lexicalised following the head lexicalisation rules developed for Spanish. The reason for the relatively large number of CFG rules is the fine-grained tags for sentential nodes which are used in the treebank (Figure 2). Of the 3638 rule types, 3533 have a sentential node on the left hand side. As many of the daughters of sentential nodes are tagged with Cast3LB functional tags, the right hand sides of 2870 of the 3638 rules are unsurprisingly completely annotated after automatic head lexicalisation and default annotation. Out of a total of 15039 right hand side nodes, 14091 (93.70%) are assigned an annotation automatically. Next the remaining partially annotated rules (768 in total) are manually examined and used to construct annotation matrices which generalise to unseen rules. The annotation matrices encode information about the left and right context of a rule's head. For example, an `espec` node to the left of the head of an `sn`'s head is a spec-

[2] For example the preterminals `ncms000` and `ncfs000` are conflated to the generic POS tag `n`.

S.F.C	Subordinated Finite Complement
S.F.R	Subordinated Finite Adjectival
S.F.A	Subordinated Finite Adverbial
S.F.A.Cond	Subordinated Conditional Finite Adverbial
S.F.A.Conc	Subordinated Concessive Finite Adverbial
S.F.A.Cons	Subordinated Consecutive Finite Adverbial
S.F.A.Comp	Subordinated Comparative Finite Adverbial
S.NF.C	Subordinated Non-Finite Complement
S.NF.A	Subordinated Non-Finite Adverbial
S.NF.P	Subordinated Non-Finite Adjectival
S.NF.R	Subordinated Non-Finite Relative
INC	Parenthetical
sn(.e)	Noun Phrase (elided)
sa	Adjectival Phrase
sadv	Adverbial Phrase
sp	Prepositional Phrase
gv	Verbal Group
infinitiu	Infinitival
gerundi	Gerund
grup.nom	Nominal group
prep	Preposition
interjeccio	Interjection
neg	Negation (no)
relatiu	Relative Pronoun
numero	Number
morfema.verbal	Pronoun *se* in passive and impersonal constructions
morf.pron	Reflexive Pronoun
espec	Specifier

Table 2: Phrasal categories from the Cast3LB Treebank

ifier while an sp node to the right of a grup.nom's head is an adjunct. Lexical information is provided by macros which are written for the POS tags.

The f-structure algorithm is implemented in Java following a similar architecture to that used for English, German and Chinese. The automatic annotation of the entire treebank is essentially a four step process illustrated in Figure 3. First, the annotation algorithm attempts to assign an f-structure equation to each node in the tree based on the Cast3LB functional labels. We have compiled an f-structure equation look-up table which assigns default f-structure equations triggered by each Cast3LB functional label. For example, the default entry for the SUJ label is ↑SUBJ=↓. Table 3 gives the complete set of default annotations. Next, the head of each local subtree of depth one is found following the head lexicalisation rules we have compiled. For example, the prep daughter of an sp node is its head and is assigned the f-structure equation ↑=↓. In the third step, the annotation algorithm deals specifically with coordination as this phenomenon is not covered by the left-right generalisations for other constructions. Figure 4 provides an example of coordination in the Cast3LB Treebank. The .co suffix on the grup.nom node label indicates that the node is mother of two or more coordinated grup.nom nodes. The coordinating conjunction (cc) is annotated as the head of the coordinated noun phrase and the coordinated elements are annotated as elements of the noun phrase's conjunct set. In a final step, the annotation algorithm moves top-down left-to-right through each tree and any unannotated nodes in each local subtree of depth one are assigned f-structure equations using the left-right context principles constructed by examining the subset of most frequent treebank rules mentioned above. For example, an sn node to the right of the head of a prepositional phrase (sp) is annotated as the object of the prepositional phrase (↑OBJ=↓). The f-structure equations are then automatically collected and passed to a constraint solver which produces an f-structure. The annotated tree and resulting f-structure for the tree in Figure 1 is shown in Figure 5. The tense, number and gender information as well as root forms are derived from the lexical macros. At present we produce "proto" f-structures (with unresolved long distance dependencies) rather than "proper" f-structures as the Cast3LB does not contain trace information.

2.3 Evaluation of the Annotation Algorithm

We first evaluated the coverage of the annotation algorithm on the entire Cast3LB Treebank. The results are presented in Table 4. 96.04% of the sentences receive one covering and connected f-structure. Ideally, we wish to generate just one f-structure per sentence. A number of sentences (102) receive more than one f-structure fragment. This is due to cases where the algorithm cannot establish a relationship between all elements in the treebank sentence and leaves nodes unannotated. There are also a small number of sentences (36) which do not receive any f-structure. These are a result of feature clashes in the annotated trees, which are caused by inconsistent annotation.

We also evaluate the quality of the annotation against a manually constructed gold standard of 100 f-

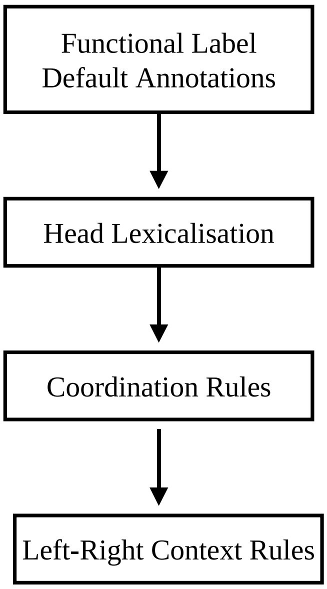


Figure 3: Architecture of Spanish Annotation Algorithm

SUJ	↑SUBJ=↓
CD	↑OBJ=↓
CI	↑OBJ_THETA=↓
ATR	↑XCOMP=↓
CPRED	↑XCOMP=↓
CAG	↑OBLAG=↓
CREG	↑OBL=↓
CC	↓∈(↑ADJ)
ET	↓∈(↑ADJ)
MOD	↓∈(↑ADJ)
NEG	↓∈(↑ADJ)
PASS	↑PASSIVE=+
IMPERS	↑IMPERSONAL=+
VOC	↓∈(↑ADJ)

Table 3: Functional tag triggered default annotations used in the Cast3LB Treebank

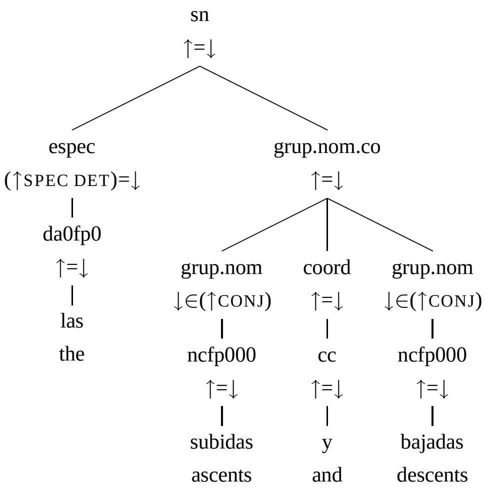
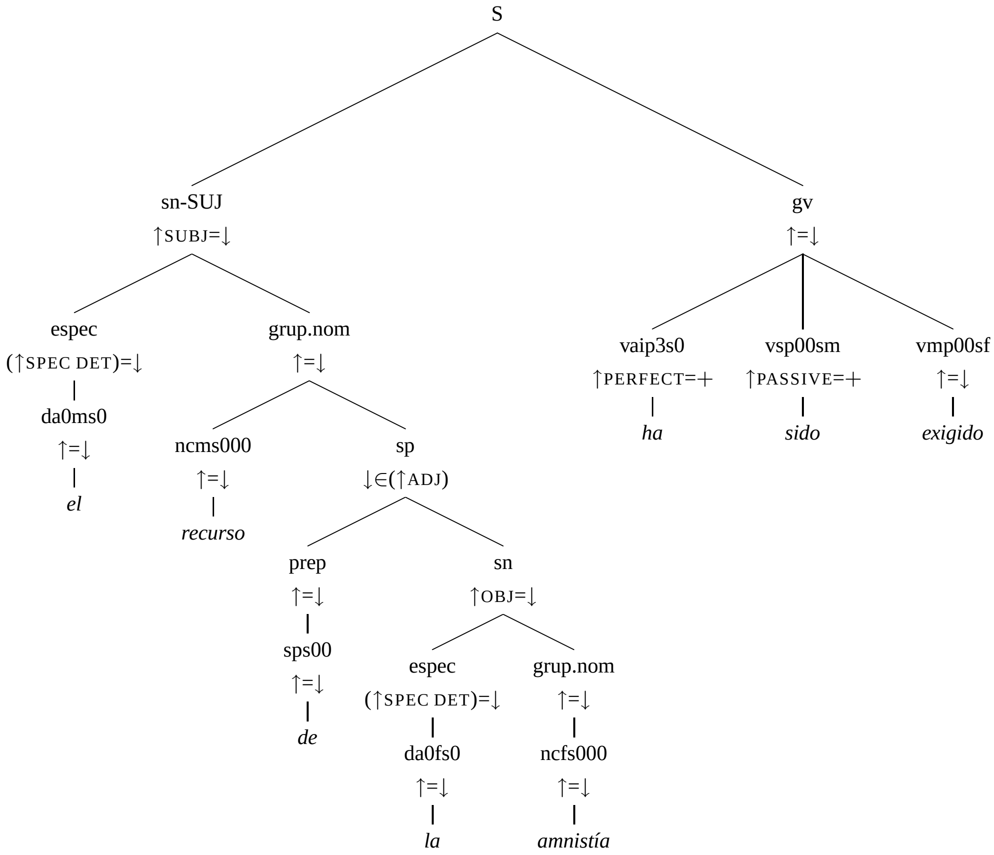


Figure 4: Coordination example from Cast3LB with automatically generated f-structure equations

F-Structures	Trees	% Trees
0	36	1.03
1	3347	96.04
2	96	2.75
3	5	0.14
4	1	0.03

Table 4: Coverage and Fragmentation results of Spanish f-structure annotation algorithm

.



$$\left[\begin{array}{ll}
\text{PRED} & \text{‘EXIGIR’}\\
\text{PERFECT} & +\\
\text{PASSIVE} & +\\
\text{TENSE} & \text{PRESENT}\\
\text{SUBJ} & \left[\begin{array}{ll}
\text{SPEC} & \left[\begin{array}{ll}\text{DET} & \left[\begin{array}{ll}\text{PRED} & \text{‘EL’}\\ \text{NUM} & \text{SING}\\ \text{GEN} & \text{MASC}\end{array}\right]\end{array}\right]\\
\text{PRED} & \text{‘RECURSO’}\\
\text{NUM} & \text{SING}\\
\text{GEN} & \text{MASC}\\
\text{ADJ} & \left\{\left[\begin{array}{ll}
\text{PRED} & \text{‘DE’}\\
\text{OBJ} & \left[\begin{array}{ll}
\text{SPEC} & \left[\begin{array}{ll}\text{DET} & \left[\begin{array}{ll}\text{PRED} & \text{‘EL’}\\ \text{NUM} & \text{SING}\\ \text{GEN} & \text{FEM}\end{array}\right]\end{array}\right]\\
\text{PRED} & \text{‘AMNISTÍA’}\\
\text{NUM} & \text{SING}\\
\text{GEN} & \text{FEM}
\end{array}\right]
\end{array}\right]\right\}
\end{array}\right]
\end{array}\right]$$

Figure 5: Automatically-annotated tree and f-structure for the example in Figure 1

	Precision	Recall	F-Score
All GFs	98.40	93.56	95.92
Preds Only	97.90	92.31	95.02

Table 5: Evaluation of the automatically produced f-structures against the 100 gold-standard f-structures

structures. For our parsing experiments we set aside approximately 10% of the treebank (336 sentences) for testing purposes. This test set is selected randomly from the various text genres which make up the treebank. We extracted 100 sentences at random from the test set, to develop our gold standard. The f-structures from the original Cast3LB trees for these sentences generated by the automatic annotation algorithm were manually corrected and converted into dependency format. We use the triples encoding and evaluation software of Crouch *et al.* (2002). Table 5 shows that currently the automatic annotation algorithm achieves an f-score of 95.92% for all grammatical functions and 95.02% for preds only. In both cases, precision is about 5% higher than recall. Table 6 shows a more detailed analysis of how well the automatic f-structure annotation algorithm performs for each function in the all grammatical functions evaluation. The algorithm performs well on most features, e.g. the OBJ f-score is 94% and that for SUBJ is 92%. At present, we score worst on the OBLAG feature (the agent in a passive construction). There are only four occurrences of this feature in the gold standard. We expect this along with all the other figures to improve as the annotation algorithm is further refined.

3 Parsing Experiments

To parse raw text into f-structures, we use the **pipeline** and **integrated** parsing architectures of Cahill *et al.* (2004), illustrated in Figure 6. For the pipeline model, we first extract a PCFG from the Cast3LB treebank excluding the 336 test sentences. Cast3LB functional tags are retained in the grammar extraction. We use Helmut Schmid's BitPar parser (Schmid, 2004) to parse new text with the grammar, using Viterbi pruning to obtain the most probable parse. The resulting parse trees are then automatically annotated using the annotation method described above. The f-structure equations are collected from the trees and passed to the constraint solver which produces an f-structure for each sentence. For the integrated model, we first automatically annotate the Cast3LB treebank with f-structure equations. We then read off a grammar from the annotated treebank, resulting in an *annotated* PCFG (A-PCFG) for Spanish. We again use BitPar to parse new text with this grammar producing annotated trees. Again the f-structure equations are collected from the parse trees and passed to the constraint solver to produce f-structures. We also transformed each grammar using a parent transformation (Johnson, 1999) to give us a P-PCFG and a PA-PCFG.

In addition, we extend Dan Bikel's multilingual, parallel-processing statistical parsing engine (Bikel,

DEPENDENCY	PRECISION	RECALL	F-SCORE
ADJUNCT	608/618 = 98	608/648 = 94	96
AUX	22/22 = 100	22/25 = 88	94
CASE	12/12 = 100	12/17 = 71	83
COMP	21/22 = 95	21/23 = 91	93
CONJ	185/190 = 97	185/196 = 94	96
DET	326/328 = 99	326/342 = 95	97
FORM	56/57 = 98	56/59 = 95	97
GEN	914/920 = 99	914/954 = 96	98
IMPERSONAL	3/3 = 100	3/3 = 100	100
NUM	1115/1130 = 99	1115/1174 = 95	97
OBJ	429/444 = 97	429/464 = 92	94
OBJ_THETA	17/17 = 100	17/19 = 89	94
OBL	13/14 = 93	13/15 = 87	90
OBLAG	2/3 = 67	2/4 = 50	57
PART	4/4 = 100	4/5 = 80	89
PARTICIPLE	27/27 = 100	27/30 = 90	95
PASSIVE	11/11 = 100	11/12 = 92	96
PERS	189/196 = 96	189/207 = 91	94
REFLEX	17/17 = 100	17/18 = 94	97
RELMOD	34/34 = 100	34/36 = 94	97
SUBJ	255/258 = 99	255/294 = 87	92
SUBORD	50/50 = 100	50/54 = 93	96
SUBORD_FORM	50/50 = 100	50/54 = 93	96
TENSE	183/187 = 98	183/196 = 93	96
XCOMP	62/66 = 94	62/73 = 85	89

Table 6: Breakdown of all grammatical functions annotation algorithm evaluation results by dependency

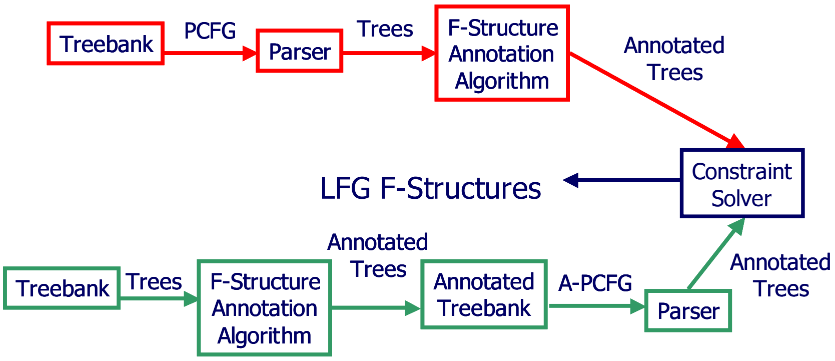


Figure 6: Pipeline (Red) and Integrated (Green) Parsing Architectures

2002) to include a language package for Spanish. Implemented in Java, the parsing engine is a history-based parser emulating Collins' Model 2 (Collins, 1997). The language package is a collection of Java classes that are extensions of several of the abstract classes which provide the description of data and methods specific to a particular language and treebank annotation style. Aside from creating the Spanish classes, we added a data file specifying the head rules specific to the Spanish Cast3LB treebank to be read by the HeadFinder class. With this extension, we trained the parser on the training set of the treebank retaining Cast3LB functional tags and parsed the test set with the grammar. Following the pipeline model, we then automatically annotated the resulting parse trees, collected the f-structure equations and passed them to the constraint solver to produce f-structures.

As previously noted, the Cast3LB preterminals are very fine-grained, encoding extensive morphological detail in addition to POS information. For example, the tag `vaip3s0` denotes a verb (`v`) which is an auxiliary (`a`), used indicatively (`i`) in the present tense (`p`), and is third person (`3`) singular (`s`). In total there are 327 preterminal types in the treebank. This level of fine-grainedness together with our relatively small training set causes a data sparseness issue for parsing new text. With such a large number of POS tags, it is inevitable that certain tags appear in the test set which have not been seen in a similar context in training with adverse effects on coverage.[3] To deal with this issue, initially we masked the morphological detail in the preterminals thereby conflating them to more generic POS tags.

3.1 Initial Results

We then parsed the 336 raw test sentences with the four grammars using BitPar and the retrained and extended Bikel parsing engine. The results are shown in Table 7. We evaluated the quality of the trees produced by the parsers using `evalb` and measured how many of the 336 sentences produce one covering

[3] If BitPar encounters a sentence in the test set containing a previously unseen tag, it will crash at that point.

	PCFG	A-PCFG	P-PCFG	PA-PCFG	Bikel
Parses (out of 336)	334	330	305	264	328
Labelled F-Score	79.01	78.89	78.78	78.44	79.19
Unlabelled F-Score	82.64	82.45	82.61	81.86	82.28
Fragmentation (336 F-Structures)	96.11	93.64	85.90	71.21	88.41
All GFs F-Score (100 F-Structures)	59.70	57.99	55.75	46.93	60.13
Preds-Only F-Score (100 F-Structures)	69.38	68.01	66.02	55.88	72.11

Table 7: Initial Parsing Results

and connected f-structure. The PCFG performs best in terms of coverage and fragmentation with over 96% of sentences being assigned one covering and connected f-structure. Coverage drops for the A-PCFG with fragmentation of 93.64%. This trend continues when parent transformations are added (71.21% for PA-PCFG). This may be attributed to data sparseness problems. The PA-PCFG rules are very information-rich and it is possible that constructions encountered in testing will not have been seen during training. As before, we evaluated the automatically produced f-structures qualitatively against the manually constructed gold standard using the evaluation software of Crouch *et al.* (2002). The results of this evaluation reveal a problem with the use of preterminal conflation to avoid data sparseness problems in parsing. Usually an all-grammatical-functions evaluation is less rigid than a preds-only evaluation as the features with atomic values (such as person, number and gender) are typically associated with the correct local `pred` even if the `pred` is attached incorrectly in global f-structure. In the case of these experiments however, the grammars score very poorly (as low as 46.93% for the PA-PCFG) in the all-grammatical-functions evaluation. By conflating the preterminal tags we discard the morphological information required by the lexical macros in the f-structure annotation algorithm to project this information to the level of f-structure.

3.2 Final Results

In order to optimise both coverage and f-structure quality we refined our morphological masking process to include a subsequent unmasking step so as to correctly trigger the lexical macros. The masking-unmasking process works as follows. The trees in the treebank are transformed in two ways: the lemmas are removed leaving behind the surface forms of the words and the preterminal tags are conflated to more general POS tags. The masked information is not disposed of but stored in a tab delimited data file in the following format: full preterminal tag, surface form of word, lemma. For example: `vaip3s0 ha haber`. The grammars are extracted from the pre-processed morphologically masked trees and used to parse new text as before. The trees produced by the parser then go through a new post-processing unmasking stage. The lemma information is re-inserted and the conflated tags are expanded. Next the lexical macros are triggered

	PCFG	A-PCFG	P-PCFG	PA-PCFG	Bikel
Parses (out of 336)	334	330	305	264	328
Labelled F-Score	79.01	78.89	78.78	78.44	79.19
Unlabelled F-Score	82.64	82.45	82.61	81.86	82.28
Fragmentation (336 F-Structures)	96.11	93.64	85.90	71.21	88.41
All GFs F-Score (100 F-Structures)	79.53	77.76	74.00	62.01	79.85
Preds-Only F-Score (100 F-Structures)	69.41	68.01	66.02	55.88	73.20

Table 8: Final Parsing Results

by the now fully unmasked POS tags and all f-structure equations are sent to the constraint solver as before. The f-structures produced now contain morphological information. The results are shown in Table 8. As expected, the `evalb` and fragmentation results are unchanged. When compared to initial f-structure results in Table 7, the improvement in the all-grammatical-functions due to this extra step is clear: between 15% and 20% for all of the grammars. There are also slight improvements for the preds-only scores of the PCFG and Bikel. The extended Bikel parsing engine performs best overall: all-grammtical-functions (79.85%) and preds only (73.20%). The PCFG, A-PCFG and P-PCFG produce f-structures of roughly similar quality. The results reported for the PA-PCFG are considerably lower. There is a general trend that the more fine-grained the grammar, the worse the coverage with PA-PCFG achieving only 71.21% fragmentation. This reflects data-sparseness problems due to the comparatively small data set. In contrast to English (Johnson, 1999), for Spanish the parent transformation has an adverse effect on parse quality.

4 Lexical Extraction

The method for automatically inducing semantic forms of O'Donovan *et al.* (2004) is highly suited to multilingual lexical extraction as it works on the level of the more language independent f-structure rather than the more language dependent c-structure. We can apply the extraction algorithm originally developed for English as is to the set of f-structures automatically generated from the Cast3LB in order to induce lexical resources for Spanish. We automatically extract 4090 semantic forms. As for English, we associate conditional probabilities with the extracted frames, differentiate between active and passive frames, parameterise frames with obliques for specific prepositions and optionally include details of syntactic category. Unlike English, the Spanish frames do not yet reflect long-distance dependencies. Of these extracted frames, 3136 are for 1401 verbal lemmas, i.e. 2.4 semantic forms per verb. The verbal semantic forms display all 98 of the frame types extracted. Table 9 provides an overview of the main extraction results broken down by category.

	Semantic Form Types	Lemmas	Frame Types
Total	4090	2322	98
Verbal	3136	1401	98
Nominal	432	432	3
Adverbial	26	24	4
Adjectival	496	474	20

Table 9: Spanish semantic forms broken down by category

Semantic Form	**Frequency**
`ser([subj,xcomp])`	1202
`estar([subj,xcomp])`	208
`tener([subj,obj])`	206
`poder([subj,xcomp])`	135
`haber([obj])`	109

Table 10: The most frequently occurring semantic forms extracted from Cast3LB

Table 10 shows the most frequently-occurring semantic forms extracted from the Cast3LB Treebank. The most frequent frame for the verb *haber* (auxiliary 'have') is `haber[obj]` due to the Spanish construction with a invariant form of this verb (*hay*) meaning 'there is' or 'there are' which never occurs with an overt subject. Table 11 shows the attested semantic forms for the verb *ver* ('see') with their associated conditional probabilities. Note that as for English, the passive frame is marked with `p`. The passive is realised in three ways in Spanish. The verb 'to be' (*ser*) is combined with a past participle in a manner similar to the English construction. Consider Figure 1 where the string *ha sido exigido* can be translated word for word to the English 'has been demanded'. The annotation algorithm uses left-right context information to annotate *sido* with the f-structure equation ↑PASSIVE=+ which is exploited by the lexical extraction algorithm at f-structure level. A reflexive construction may also be used to express the passive. For example, ... *se registró un descenso...* ('... a descent was registered...') where *un descenso* is the surface subject of the normally transitive *registrar*. In Cast3LB the pronominal constituent (*se*) is tagged as a `morfema.verbal` and has an additional functional tag -PASS which is used by the annotation algorithm to assign the ↑PASSIVE=+ f-structure equation. Finally, the Spanish passive may be realised using the third person plural of the verb to be passivised with an empty subject. In this case the verb used passively will not be marked as such because it does not display the movement typically associated with the passive and is essentially an active construction with an empty subject.

Semantic Form	Conditional Probability
`ver([subj,obj])`	0.468
`ver([subj])`	0.290
`ver([subj,comp])`	0.121
`ver([subj],p)`	0.072

Table 11: Automatically extracted lexical entries for *ver* (see) with associated conditional probabilities

5 Conclusions and Future Work

We have shown how the methodology for automatically annotating the Penn-II Treebank with LFG f-structure equations for the purpose of extracting grammatical and lexical resources can be adapted to Spanish. The methodology has also been successfully migrated to German and Chinese. Our methodology constitutes a novel approach to deep multilingual constraint-based grammar and lexical acquisition based on treebank resources and automatic f-structure annotation algorithms. As treebanks become available for a growing number of languages, we expect this method can deliver robust, wide-coverage multilingual resources with a substantial reduction in development cost. The multilingual work presented here is very much proof of concept. Just three months of development effort have been invested to induce the resources and further work is required to integrate long-distance dependency resolution and to refine the grammar and lexicon extraction.

We developed and applied an automatic f-structure annotation algorithm to the treebank and measured its coverage as well as the quality of the annotations. Over 96% of the trees in the treebank receive one covering and connected f-structure. When evaluated against a gold standard of 100 hand-crafted f-structures, the algorithm scores over 95% for preds-only and all-grammatical-functions. We extract four different PCFGs from the treebank and use them to parse 336 sentences set aside for testing. We also extend and retrain Bikel's (2002) statistical parsing engine with a Spanish language package to parse the test set. The retrained Bikel parser integrated into the pipeline model performs best against the gold standard, achieving a preds-only f-score of 73.20% against the gold standard. We extract 4090 semantic forms from the annotated treebank using the same methodology applied to the Penn-II Treebank. Long-distance dependency resolution, refinement and extension of the annotation algorithm, grammar and lexicon extraction as well as the evaluation of the lexical resources remain as future work.

References

Bikel, Daniel M. 2002. Design of a Multi-lingual, Parallel-processing Statistical Parsing Engine. In *Pro-*

ceedings of the Human Language Technology Conference, pages 24–27, San Diego, CA.

Brants, Thorsten, Stefanie Dipper, Silvia Hansen, Wolfgang Lezius, and George Smith. 2002. The TIGER Treebank. In E Hinrichs and K Simov, editors, *Proceedings of the first Workshop on Treebanks and Linguistic Theories (TLT'02)*, pages 24–41, Sozopol, Bulgaria.

Burke, Michael, Olivia Lam, Rowena Chan, Aoife Cahill, Ruth O'Donovan, Adams Bodomo, Josef van Genabith, and Andy Way. 2004. Treebank-Based Acquisition of a Chinese Lexical-Functional Grammar. In *Proceedings of the 18th Pacific Asia Conference on Language, Information and Computation*, pages 161–172, Tokyo, Japan.

Cahill, Aoife, Michael Burke, Ruth O'Donovan, Josef van Genabith, and Andy 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 320–327, Barcelona, Spain.

Cahill, Aoife, M. Forst, M. McCarthy, R. O'Donovan, C. Rohrer, J. van Genabith, and A. Way. 2003. Treebank-Based Multilingual Unification-Grammar Development. In *Proceedings of the Workshop on Ideas and Strategies for Multilingual Grammar Development, at the 15th European Summer School in Logic Language and Information*, pages 17–24, Vienna, Austria.

Civit, Montserrat. 2003. *Criterios de etiquación y desambiguación morfosintáctica de corpus en español*. Ph.D. Thesis, Universitat Politècnica de Catalunya.

Collins, Michael. 1997. Three Generative, Lexicalized Models for Statistical Parsing. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics*, pages 16–23.

Crouch, Richard, Ron Kaplan, Tracy Holloway King, and Stefan Riezler. 2002. A comparison of evaluation metrics for a broad coverage parser. In *Proceedings of the LREC Workshop: Beyond PARSEVAL – Towards Improved Evaluation Measures for Parsing Systems*, pages 67–74, Las Palmas, Canary Islands, Spain.

Johnson, Mark. 1999. PCFG models of linguistic tree representations. *Computational Linguistics*, **24**(4):613–632.

O'Donovan, Ruth, Michael Burke, Aoife Cahill, Josef van Genabith, and Andy Way. 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 368–375, Barcelona, Spain.

Schmid, Helmut. 2004. Efficient Parsing of Highly Ambiguous Context-Free Grammars with Bit Vectors. In *Proceedings of the 18th International Conference on Computational Linguistics (COLING 2004)*, pages 162–168, Geneva, Switzerland.

Xue, Nianwen, Fu-Dong Chiou, and Martha Palmer. 2002. Building a Large-Scale Annotated Chinese Corpus. In *Proceedings of the 19th International Conference on Computational Linguistics (COLING 2002)*, Taipei, Taiwan.