Abstract

We present the construction of a HPSG corpus for Spanish, based on the transformation of the AnCora Spanish corpus into a HPSG compatible format. We describe the transformation process and the evaluation of the resulting corpus.

1 Introduction

We describe the first phase of a currently ongoing project for building a statistical HPSG parser for Spanish. It consists in transforming the AnCora Spanish corpus from its CFG-style annotations to an HPSG (Pollard and Sag, 1994) compatible format, in a next stage we extract a lexicon and train a supertagger over the transformed corpus (Chiruzzo and Wonsever, 2015). Head-driven Phrase Structure Grammars (HPSG) are a strongly lexicalized grammar formalism. This family of grammars are very expressive, allowing the modeling of many linguistic phenomena and capturing syntactic and semantic notions at the same time. The rules used in an HPSG grammar are very generic, indicating how a syntactic head can be combined with its complements, modifiers (adjuncts) and specifier. The categories of the elements are organized in a type hierarchy and the parsing result is a tree whose nodes are typed feature structures (Carpenter, 1992).

Our work is inspired by Enju (Matsuzaki et al., 2007), a statistical HSPG parser for English that has high performance and language coverage. This parser was built based on the Penn Treebank corpus (Marcus et al., 1993). As the Penn Treebank was not annotated in an HPSG compatible format but rather in a CFG-style grammar, they built a set of rules to transform the Penn Treebank trees into a structure that is similar to HPSG (Miyao et al., 2005). The Enju parser is trained using the result of this transformation.

Other HPSG grammars for Spanish exist, the most relevant one being the Spanish Resource Grammar (SRG) (Marimon, 2010), a Spanish HPSG grammar built using the LinGO Grammar Matrix (Bender et al., 2002), a framework for building HPSG grammars for many languages. SRG can be used with the LKB development system (Copestake et al., 1999), as well as the PET runtime parser (Callmeier, 2000), and its results are very rich HPSG trees that include all of the constructions supported by the theory. Our objective is to build a new HPSG parser whose representations will not be as rich as SRG's, but we aim at making it more robust. Also, the statistical model, trained from the transformed corpus and the extracted lexical units, will compute directly the desired output instead of acting as a filter for the great number of output trees resulting from the grammar non-stochastic constraints as in (Marimon et al., 2014).

AnCora is a corpus for Spanish and Catalan (Taulé et al., 2008) that contains about half a million words in 17,000 sentences. The corpus has CFG-style annotations, but it is also enriched with attributes such as morphological information and predicate-arguments structure. Inspired by Enju, we aimed to transform this corpus into a treebank compatible with HPSG. There exists another Spanish treebank with HPSG annotations: the Tibidabo corpus (Marimon, 2015). However, this corpus seems not to be publicly available, and, more important to us, its structure does not suit to our purposes. Also, Tibidabo contains only 4000 sentences from the AnCora Spanish corpus which consists of 17000 annotated sentences. Each sentence in Tibidabo is represented by three graphs: a binary constituent tree, with atomic category names, a dependency tree annotated with syntactic label names and a MRS structure (Copestake et al., 2005), as shown in (Marimon, 2015). Another difference between our approach and Tibidabo is that we transform the whole AnCora corpus, using the corpus information to guide the transformation, while Tibidabo re-annotated some of the sentences of the corpus using SRG, but dropping the original annotation information.

On the other hand, as mentioned above, the trees annotated using SRG have richer structures than the ones we get after the transformation. In particular, the Tibidabo corpus maintains the MRS structures as its sentences semantic representation, which includes event variables, standard arguments naming and quantifier scopes, among other things. MRS is, in some sense, a meta-notation for first order logical forms that allows underspecification and thus packs scope ambiguities. Our approach to semantics is much simpler: using the information readily available in AnCora, we include features for representing the predicate-argument structure of the verbs (also for other predicates, e.g. deverbal nouns). The predicate-argument structure is annotated in PropBank style (Kingsbury and Palmer, 2002), so all our SEM feature needs is a set of features for each of the PropBank arguments (ARG0, ARG1, ARGM...). This simplified approach to semantics is similar to the one used in Enju.

2 Description of the grammar

This section describes the main aspects of the feature structure we used and the grammar rules.

2.1 Feature structure

The general feature structure for a lexical entry in our grammar is shown in figure 1. This feature structure tries to summarize all features that could be included in one of the lexical entries. The structure has morphological, syntactic and our simplified semantic features. In this feature structure, the feature COMPS can have a list of expressions, while the features SPEC and MOD are shown as lists because they might have zero or one expressions.

Figure 2 shows a concrete example of a lexical entry for a typical transitive verb. Notice that the subject of the verb (SPEC) is coindexed with the proto-agent argument (ARG0), and the only complement is coindexed with the proto-patient argument (ARG1). Thus, this lexical entry represents the active voice instance of

$$\begin{bmatrix} \text{word} & \\ \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \text{n} \vee \text{v} \vee \text{a} \ldots & \\ \text{AGR} & \begin{bmatrix} \text{PER } 1 \vee 2 \vee 3 \\ \text{NUM sg} \vee \text{pl} \\ \text{GEN f} \vee \text{m} \\ \ldots \end{bmatrix} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{SPEC} & \left\langle \text{expr} \right\rangle \\ \text{COMPS} & \left\langle \text{expr} \right\rangle \\ \text{MOD} & \left\langle \text{expr} \right\rangle \end{bmatrix} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{ARGS} & \begin{bmatrix} \text{ARG0} & \text{expr} \\ \text{ARG1} & \text{expr} \\ \ldots & \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

Figure 1: Feature structure for a lexical entry

this transitive verb.

$$\begin{bmatrix} \text{word} & \\ \text{SYN} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \text{v} & \\ \text{AGR} & \begin{bmatrix} \text{FORM} & \text{personal} \\ \text{MODE} & \text{i} \\ \text{PER} & 3 \\ \text{NUM} & \text{sg} \end{bmatrix} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \text{SPEC} & \left\langle \boxed{1}\ \text{SYN.HEAD n} \right\rangle \\ \text{COMPS} & \left\langle \boxed{2}\ \text{SYN.HEAD n} \right\rangle \end{bmatrix} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{ARGS} & \begin{bmatrix} \text{ARG0} & \boxed{1} \\ \text{ARG1} & \boxed{2} \end{bmatrix} \end{bmatrix} \\ \text{TEXT} & \textit{come} \end{bmatrix}$$

Figure 2: Feature structure for transitive verb *"come"*, indicative third person singular form of the verb *"to eat"*

2.2 Grammar rules

The rules of the grammar we use are a simplified version of the ones used in (Pollard and Sag, 1994). The grammar has rules for combining a specifier, a complement or an adjunct to a head, and two rules for binarizing the coordinated constructions. There are also extra rules for simplifying the analysis of clitics and relative constructions, which will have further development in the future. Despite using these simplified rules, the grammar is able to deal with some interesting linguistics constructions.

2.2.1 Specifiers

We define two rules for combining a specifier with a head: `spec_head` and `head_spec` which apply the specifier to the left or to the right of the head respectively. In both cases the `HEAD` feature of the resulting phrase is coindexed with the `HEAD` feature of the head. These rules are used for applying the determiner of a noun phrase (only the `spec_head` rule in this case), and also for applying the subject of a sentence. The `SPEC` feature of the resulting phrase is cleared.

Notice that we allow for a specifier to be combined both to the left or to the right of the head. Although Spanish typology is generally regarded as SVO, there are plenty of exceptions to this rule. It is very common to find sentences in which the object is located before the verb, or the subject is located after the verb, for example: *"llegó el tren" / "the train arrived"*. The AnCora corpus contains many examples of these constructions. We chose this representation instead of using a `SLASH` feature and a head-filler rule because we consider it would be easier to extract statistics from the corpus on which verbs are usually combined with a subject to the left or to the right.

2.2.2 Complements and adjuncts

There are two rules for combining a complement with a head: `comp_head` and `head_comp` which apply the complement to the left or to the right of the head respectively. In both cases the `HEAD` feature of the resulting phrase is coindexed with the `HEAD` feature of the head. One of the expressions in the list of the `COMPS` feature is cleared. The expression that is cleared depends on the verb and the complement being addressed. This information has to be extracted from the corpus. Notice that these rules are binary, so in order to combine a head with multiple complements the rules have to be applied several times.

There are two different rules for combining an adjunct or modifier with a head: `mod_head` and `head_mod` which apply the adjunct to the left or to the right of the head respectively. In both cases the `HEAD` feature of the resulting phrase is coindexed with the `HEAD` feature of the head. The `MOD` feature of the adjunct is coindexed with the head.

In the AnCora corpus the distinction between complements and adjuncts is not always overtly annotated, we rely on a series of hand written rules that consider the category of the head, the category of the expression and several different annotation attributes the corpus includes. These rules were created by manually inspecting the corpus.

For example, the rules for detecting the complements of a verb in a subordinate sentence take into consideration the attribute `func` that might be present in the AnCora XML element that describes the constituent. This attribute represents the syntactic function of the constituent. In an ideal case, this attribute would be enough to detect if a constituent is a complement or not. However, the attribute is not always properly annotated in the corpus in all the constituents that should require it. Because of this, by manually inspecting the corpus, several other rules were added that consider other attributes and exceptional cases, in order to capture as many correct examples as possible. For example, if the `func` attribute is missing from the constituent we might make use of the attribute `arg`, which defines its role in the predicate argument structure. This distinction is not perfect in all cases, see section 4 for details about the performance of these rules.

2.2.3 Coordinations

In our grammar, coordinated structures need to be binarized, which is done using two rules: `coord_right` and `coord_left`. First the conjunction and the right expression are put together using the `coord_right` rule, then the resulting phrase and the left expression are put together using the `coord_left` rule. This is iterated for longer chains of coordinations, resulting in a chain of binary trees.

2.2.4 Clitics

Clitic pronouns need special attention in Spanish because they sometimes act as complements (in substitution of a complement that was already mentioned in the text) and sometimes both the real complement and the clitic are present at the same time (Pineda and Meza, 2005) (this is called *clitic doubling*). Because of this, we created a new rule for dealing with clitics, different from the rules for applying complements. This rule is `clitic_head`, which applies a clitic to the left of the head. During the transformation, we annotate all clitics using this rule but we do not perform any further analysis to recover the actual complement the clitic is refering to, should it be present. An appropriate handling of the clitic analysis would need a way of classifying the cases that deal with clitic doubling and providing a consistent analysis for this cases. This aspect has not been addressed yet.

2.2.5 Relative clauses

Relative pronouns introduce a subordinate sentence inside another sentence that acts as a modifier to a noun phrase and at the same time use the noun phrase as an argument (e.g. in *“el perro que me mordió” / “the dog that bit me”* the noun *“perro” / “dog”* is both modified by and the subject of the subordinate sentence). This is another kind of long distance dependency that is usually dealt with using `SLASH` features and filler rules in HPSG. Currently in our work we are not resolving this type of long distance dependency, so we created a new rule `head_rel` to mark these constructions. In the future these `head_rel` will be resolved using a `SLASH` feature or a similar construction.

2.2.6 Control verbs

Control verbs are verbs which govern over the arguments (subject or object) of another subordinate verb. In AnCora, the combination of a subject control verb (such as *"comenzar" / "to start"*) together with its subordinate verb, is generally annotated as a verb phrase structure. However, an object control verb (such as *"obligar" / "to oblige"*) is not annotated as a verb phrase structure together with its subordinate verb. In our corpus transformation, we resolve the subject of the subordinate verb in the subject control verb constructions, but further analysis is needed in order to resolve the correct coindexation in the object control verb constructions.

3 Transformation process

In a HPSG tree, it is necessary to know the syntactic head of every constituent and also the roles that the rest of the elements of the constituents have. This information is not directly available in AnCora, so we created a series of heuristics that exploit the information in the corpus (structure and attributes) in order to transform it to a HPSG compatible format. Figure 3 shows an example of a sentence annotated using the AnCora markup: *"El desarrollo, la integración y la cooperación fueron los asuntos protagonistas de esta reunión." (Development, integration and cooperation were the main matters of this meeting.)*

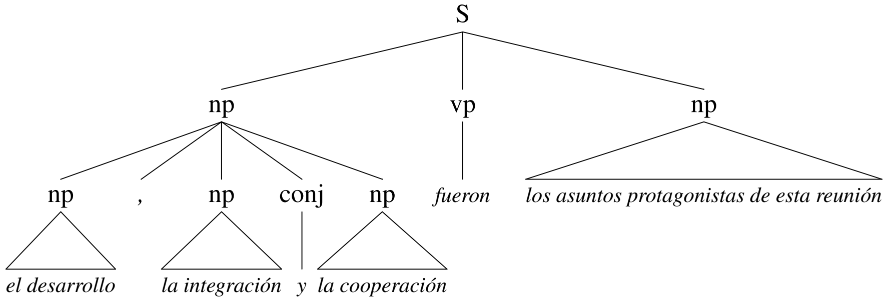

Figure 3: Sentence annotated with its syntactic structure in AnCora

If we consider the syntactic structures of AnCora as annotated in a CFG, the number of rules in this grammar would be very large. For example, there are 5,800 ways of writing a subordinate sentence, and 900 ways of writing noun phrases. Because of this, we tried to reduce the complexity of the problem using a transformation process which uses two stages: a top-down approach that works together with a bottom-up approach.

- `n` (noun)
  “…Río_Bravo y Saltillo para la [[$_{\text{H}}$ compañía] [francesa]]…”
- `grup.nom` (nested noun phrase)
  “…y sobre [[$_{\text{H}}$ transmisiones y retenciones] [de fondos de inversión]] .”
- `p` (pronoun)
  “…obtuvo 19 diputados, [[$_{\text{H}}$ dos] [más]] que en 1996…”
- `w` (date)
  “…hundimiento del “Kursk” el [[pasado] [$_{\text{H}}$ 12_de_agosto]] en aguas árticas…”
- `z` (number)
  “…donde lograron el [[$_{\text{H}}$ 71_por_ciento] [de los sufragios]] …”
- `a` (adjective)
  “…quien cuestiona al entrenador es [[$_{\text{H}}$ enemigo] [del Barça]] .”
- `v` (verb)
  “…sobre todo en el [[$_{\text{H}}$ capitulo] [de las infraestructuras]] …”
- `s.a` (adjective phrase)
  “…y la [[$_{\text{H}}$ segunda] [, mucho más potente,]] a las 07.30.42…”
- `participi` (participle)
  “…el relato ZZadjNM de lo [[$_{\text{H}}$ ocurrido] [en la sima de ZZlugar]] …”
- `S/clausetype=participle` (subordinate sentence of type participle)
  “…en_lugar_del [[$_{\text{H}}$ destituido] [Carlos_Sainz_de_Aja]] .”
- `S/clausetype=relative` (subordinate sentence of type relative)
  “…incluidos los [[$_{\text{H}}$ que él mismo ha hablado] [sobre sí mismo]] …”
- `S/clausetype=completive` (subordinate sentence of type completive)
  “Al [[$_{\text{H}}$ correr] [de los siglos]] se había manifestado un…”
- `sp` (prepositional phrase)
  “aeropuerto de Miami, uno de los [[$_{\text{H}}$ de mayor tráfico aéreo] [en EEUU]]…”
- `sn` (noun phrase, maximal projection)
  “…el hotel (un [[$_{\text{H}}$ cinco estrellas de gran lujo]])…”

Table 1: Rules for head detection inside a `grup.nom`

We define an *elementary HPSG tree* as a simple tree that consists of a syntactic head surrounded by elements that are directly related to the head (complements, modifiers, specifier). The top-down process tries to transform the most complex structures of the corpus into simpler trees. This means breaking up a node with too many children into a composition of elementary trees that preserve the original structure. The top-down process is in charge, among other things, of extracting quoted or punctuated blocks; marking clitics; extracting prepositional phrases, relative clauses and subordinate sentences; and binarizing sequences of coordinations.

The bottom-up approach assumes that the top-down process has dealt with all those complex structures and left only a set of homogeneous simpler structures, those structures will become elementary HPSG trees after the transformation. In order to transform these trees, we created head detection and arguments classification heuristics. For the English language there is a commonly used heuristic for finding the syntactic head of a phrase in the Penn Treebank corpus, as described in (Collins, 2003). As there is no equivalent for Spanish, and the grammatical differences between both languages make it impossible to apply the same rules, a set of head detection rules was manually crafted for the elements of Ancora. We defined lists of constraints that an element must match in order to be considered the head of a constituent. The constraints are written in a small language for rules that was created for this purpose. Table 1 shows some examples of the list of detection rules that is used to find the head of a noun phrase (elements of type `grup.nom` in AnCora).

After finding the syntactic head of a phrase, we proceed to analyze the elements that are directly to the left or to the right of the head, and apply a series of heuristics that try to classify the role of those arguments with respect to the head. The heuristics use information about the node such as its part of speech, but also the attributes of the element. The rules for classifying the elements are written in the same language as the rules for detecting heads. In total there are 70 head detection rules and 184 argument classification rules.

Besides these rules, there are specific transformation heuristics for verb phrases, because the verb phrases in AnCora behave different from other constituents and could not be reduced to elementary HPSG trees (see section 2.2.6).

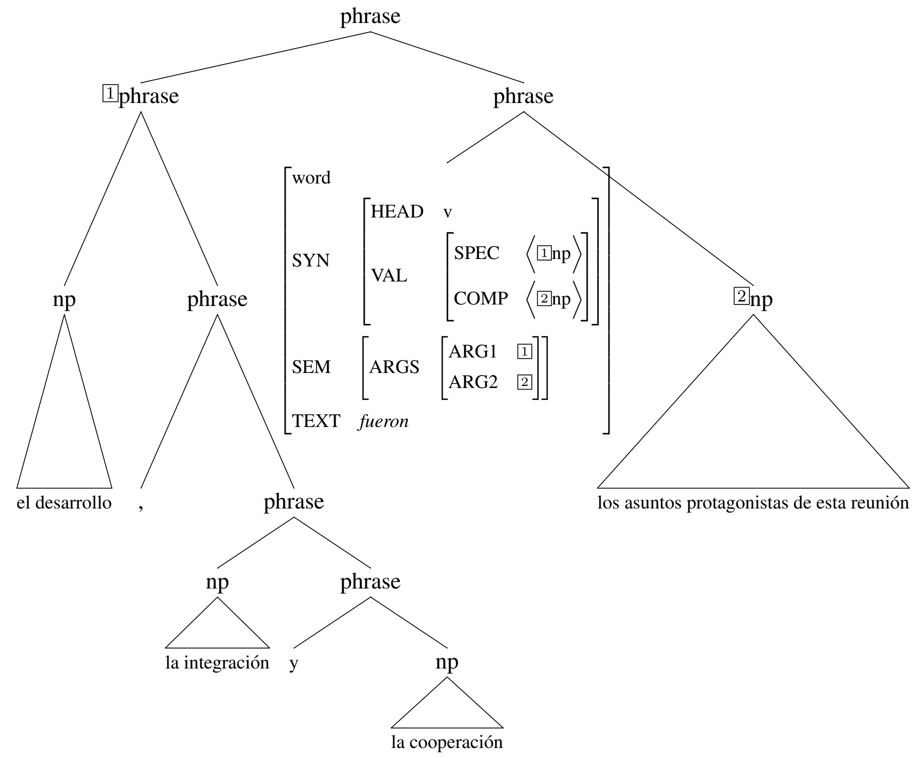

Figure 4: Sentence after the transformation process

Figure 4 shows what the sentence in the previous example looks like after the transformation. Notice that the coordination has been binarized; the head, complement and specifier have been identified; and the appropriate arguments are coindexed in the structure. Although for the sake of clarity only the feature structure for the word *"fueron"* is shown in the diagram, the transformation process creates feature structures for all the word and phrase nodes in the tree.

4 Evaluation

The transformed corpus contains only binary or unary constituents and all nodes indicate their syntactic head and the applied rule. AnCora has a total of 780950 constituents and almost all of them could be transformed. We evaluated the accuracy of the transformation heuristics in the following way: We took a random sample of 40 sentences (779 constituents) and manually identified the syntactic head of every constituent and the role of every other element with respect to the head (complement, modifier, specifier, clitic or punctuation mark).

We found that the head detection heuristics have a precision of 95.3%, which climbs to 98.7% if we do not consider the nodes with coordinations. Table 2 shows the precision of the head detection rules by constituent category, considering nodes with coordinations.

AnCora Category	Total	Correct	Precision
grup.a (adjectival group[1])	9	6	66.7%
grup.adv (adverbial group)	3	3	100.0%
grup.nom (noun group)	162	154	95.1%
grup.verb (verb phrase)	23	23	100.0%
infinitiu (infinitival verb phrase)	3	3	100.0%
relatiu (relative pronominal expression)	1	1	100.0%
S (subordinate sentence)	91	85	93.4%
s.a (adjectival phrase)	4	3	75.0%
sa (adjectival phrase[2])	1	1	100.0%
sadv (adverbial phrase)	7	7	100.0%
sentence (sentence)	40	35	87.5%
sn (noun phrase)	220	216	98.2%
sp (prepositional phrase)	207	204	98.6%
spec (determiner phrase)	8	1	12.5%

Table 2: Precision of head detection rules

The arguments classification heuristics have a precision of 92.5% on average, and the category which is the most difficult to classify is the complements (84.95% precision). Table 3 shows the confusion matrix for the arguments classification.

[1] In AnCora, a "group" in general is different from a "phrase" in that it cannot contain a specifier, though there are many examples that break this rule in the corpus.

[2] There are two types of adjectival phrases in AnCora: `sa` and `s.a`. In practice, there seems to be no difference between them as they are used in the corpus.

	Specifier	Complement	Modifier	Clitic	Punctuation
Specifier	279	3	3	0	0
Complement	6	333	53	0	0
Modifier	1	18	247	0	0
Clitic	0	0	0	19	0
Punctuation	0	0	0	0	155

Table 3: Confusion matrix for the arguments classification

5 Conclusions and future work

We described a transformation process that takes the AnCora Spanish corpus and transforms its CFG style annotations into HPSG compatible structures. The result of this process is a collection of trees annotated in HPSG style where the head of every constituent is marked; the arguments are classified; and all lexical entries include morphological, syntactic and semantic information.

The transformation process achieves a precision of 95.3% for head detection (98.7% without considering coordinations) and a precision of 92.5% for arguments classification. These are promising results, but there is still room for improvement, specially for the arguments classification. In order to improve performance we might need to refine the arguments classification heuristics. Furthermore, the transformed corpus is missing some interesting features such as the analysis of the Spanish clitics as arguments of the verbs and the analysis of long distance dependencies. This transformed corpus is used in a later stage to extract a lexicon of Spanish words and to train a supertagger for verbs, nouns and adjectives, with the aim of creating a statistical parser for Spanish.

References

E. M. Bender, D. Flickinger, and S. Oepen. The grammar matrix: An open-source starter-kit for the rapid development of cross-linguistically consistent broad-coverage precision grammars. In *Proceedings of the 2002 workshop on Grammar engineering and evaluation-Volume 15*, pages 1–7. Association for Computational Linguistics, 2002.

U. Callmeier. Pet–a platform for experimentation with efficient hpsg processing techniques. *Natural Language Engineering*, 6(01):99–107, 2000.

B. Carpenter. *The logic of typed feature structures*. Cambridge University Press, 1992.

L. Chiruzzo and D. Wonsever. Supertagging for a statistical hpsg parser for spanish. In *International Conference on Statistical Language and Speech Processing*, pages 18–26. Springer, 2015.

M. Collins. Head-driven statistical models for natural language parsing. *Computational linguistics*, 29(4):589–637, 2003.

A. Copestake, J. Carroll, R. Malouf, and S. Oepen. The (new) lkb system. *Center for the Study of Language and Information, Stanford University*, 1999.

A. Copestake, D. Flickinger, C. Pollard, and I. A. Sag. Minimal recursion semantics: An introduction. *Research on Language and Computation*, 3(2-3):281–332, 2005.

P. Kingsbury and M. Palmer. From treebank to propbank. In *LREC*. Citeseer, 2002.

M. P. Marcus, M. A. Marcinkiewicz, and B. Santorini. Building a large annotated corpus of english: The penn treebank. *Computational linguistics*, 19(2):313–330, 1993.

M. Marimon. The spanish resource grammar. In *Proceedings of the International Conference on Language Resources and Evaluation, LREC*, pages 17–23, Valletta, Malta, 2010.

M. Marimon. Tibidabo: a syntactically and semantically annotated corpus of spanish. *Corpora*, 10(3):259–276, 2015.

M. Marimon, N. Bel, and L. Padró. Automatic selection of hpsg-parsed sentences for treebank construction. *Computational Linguistics*, 40(3):523–531, 2014.

T. Matsuzaki, Y. Miyao, and J. Tsujii. Efficient hpsg parsing with supertagging and cfg-filtering. In *IJCAI*, pages 1671–1676, 2007.

Y. Miyao, T. Ninomiya, and J. Tsujii. Corpus-oriented grammar development for acquiring a head-driven phrase structure grammar from the penn treebank. In *Natural Language Processing–IJCNLP 2004*, pages 684–693. Springer, 2005.

L. Pineda and I. Meza. The spanish pronominal clitic system. *Procesamiento del lenguaje natural*, 34:67–103, 2005.

C. Pollard and I. A. Sag. *Head-driven phrase structure grammar*. University of Chicago Press, 1994.

M. Taulé, M. A. Martí, and M. Recasens. Ancora: Multilevel annotated corpora for catalan and spanish. In *LREC*, 2008.