Creating an MTT Treebank of Spanish[1]

Simon Mille, Vanesa Vidal, Alicia Burga
Barcelona Media
Av. Diagonal, 177, Planta 10
08018 Barcelona, Spain
<fname>.<lname>@upf.edu

Leo Wanner
ICREA and Universitat Pompeu Fabra
C. Roc Boronat, 138
08018 Barcelona, Spain
leo.wanner@icrea.es

Abstract

We present a cost effective strategy for the creation of a mid-size fine-grained dependency treebank of surface- and deep-syntactic structures as defined in the Meaning-Text Theory for Spanish. The strategy starts from a small seed dependency corpus, the AnCora corpus, whose annotation is considerably more coarse-grained than our target annotation. We show that this discrepancy can be bridged largely by automatic means, relying upon contextual information and leaving thus minimal work to the annotators. This allows us to develop the resources with limited human effort within a limited period of time. We also propose a preliminary evaluation of the actual amount of work that the annotation process requires.

1 Introduction

The syntactic annotation of corpora is nowadays a popular exercise in Computational Linguistics (CL). This is certainly also because the CL-community became aware that syntactically annotated corpora (or treebanks) can be used for a variety of applications – for instance, as a source for the compilation of dictionaries, as training material for machine learning-based parsing or generation, as teaching material in computer-based second language acquisition, etc.

Although the annotation with constituency trees has a longer tradition – with the first and probably still the most prominent constituency treebank being the Penn Treebank (Mitchell *et al.*, 1999) – annotation with syntactic dependency trees is gaining pace. Even more: one could say that the creation of dependency corpora is nowadays "in"; cf., for instance the Prague Dependency Treebank for Czech, containing about 96.000 sentences annotated with two levels of syntactic information – the analytical and tectogrammatical layers (Hajič *et al.*, 2006) –, the Portuguese Bosque corpus (a fully revised subset of about 9000 sentences from the Floresta corpus) (Afonso *et al.,* 2002), the Dutch Alpino treebank (van der Beek et al. 2002) with about 13350 sentences, the Swedish treebank with 11000 sentences (Nilsson et al. 2005) and the dependency version of the Penn Treebank corpus (Mitchell *et al.*, 1999).

Unfortunately, up to date, the corpus annotation initiative did not receive the due attention in the MTT community. To the best of our knowledge, so far only for Russian a stable MTT treebank is available (Apresjan *et al.*, 2006); for German, Bohnet (2003) describes experiments to map the Tiger corpus (Brants et al., 2002) onto surface-syntactic structures; and for French, the annotation of a spoken language corpus is under way, but no resources are available as yet (Gerdes, personal communication). Given the multiple prospects that a treebank offers for research and practical applications and the increased attractiveness a linguistic framework has for computational linguists in case it has to offer extended treebanks which can be used for algorithm evaluations, we think that it is very important for MTT to increase its visibility in this area.

[1] Partially funded by the Spanish Ministry of Science and Innovation (MCI FFI 2008-06479-CO2-02/FILO), in the framework of the COLOCATE project.

Our mid-term goal is the annotation of mid-size corpora (about 30.000 sentences) for Spanish, Catalan, English, German, and French, which are our primary working languages, with the structures of all levels of the MTT-model. Particular focus is put on the annotation of surface-syntactic structures (SSyntSs), deep-syntactic structures (DDSyntSs), and semantic structures (SemSs) – followed later on by the communicative structure (CommSs) at each level.[2] Starting from SSyntSs, we can automatically derive large scale Government Pattern dictionaries needed for a correct annotation with DSyntSs. Furthermore, with SSyntSs at hand, we can, at least partially, automate the process of the DSyntS-annotation and then, of the SemS-annotation.

Currently, we are working on the annotation of a Spanish corpus with SSyntSs, performing in parallel experiments on the annotation with DSyntSs and SemSs. In order to speed up the procedure of the SSyntS-annotation, we started from an existing small size Spanish dependency annotated corpus – the AnCora corpus (Martí *et al.*, 2007), whose annotation is considerably more coarse-grained than SSyntS in the Meaning-Text Theory and whose annotation conventions partially contradict the principles of MTT. But it can be semi-automatically mapped onto SSyntSs and thus serve as a seed corpus upon which the automated annotation draws.

In what follows, we describe our annotation strategy, the state of our ongoing work and our future plans. In Section 2, we provide details of the annotation procedure with SSyntSs. In Section 3 we propose a preliminary assessment of the costs of the annotation procedure. Section 4 shows how SSyntSs can be used to obtain in a relatively short time and with a relatively small effort a high quality DSyntS annotation. Section 5, finally, summarizes the paper.

2 The annotation strategy

Let us, before we delve into the details of the AnCora annotation conventions and our annotation procedure, assess the options available to annotate a corpus with SSyntSs.

2.1 Initial considerations: How to annotate a corpus with SSyntSs?

There are four alternative options for the annotation of an available (cleaned) corpus with dependency structures such as SSyntS: **I.** Manually, starting from the scratch, i.e., from a raw corpus. This option would guarantee a high quality (provided that the annotators are adequately trained and high degree of mutual agreement between the annotators is ensured), but is extremely costly. **II.** Using SSyntS-dependency parsers. Kakkonen (2006) suggests that the annotators use several dependency parsers and compare the outputs so as to produce a correctly annotated sentence. The comparison can be done automatically, based on the probability of the correctness of each parser, or manually – along with a potentially necessary correction. Unfortunately, not a single Spanish SSyntS-parser which could be used on the spot is available as yet. **III.** Starting from a constituency Treebank, mapping the constituency trees onto SSyntS dependency trees. For instance, the constituency corpus Cast3LB has already been used by Herrera *et al.* (2007) for the derivation of dependency annotations. They used the algorithm of Gelbukh *et al.* (2005) that converts constituency structures into dependency structures. Similar efforts have been made at Lund University to convert Penn-style Treebanks (Johansson and Nugues, 2007) and in the context of the ConLL shared tasks (Surdeanu *et al.*, 2008). The problem here is that it is quite difficult to obtain accurate output structures as soon as sentences are somewhat more complex. **IV.** Starting from an already existing dependency Treebank, mapping the available dependency structures onto SSyntSs. In general, given the high number of SSyntS-relations, this would imply that many SSyntS-relations will be missing and would thus need to be added either semi-automatically or manually; in addition, Spanish dependency corpora are very small. The big advantage of this option is, however, that at least the dependencies are in place.

Given the circumstances, we had to adopt the last option. The dependency Treebank from which we start is AnCora_DEP_ES (Martí *et al.*, 2007), which comprises 3512 sentences.

[2] Readers not familiar with the MTT model and terminology are asked to consult, e.g., (Mel'čuk, 1988).

2.2 Our starting point: The AnCora corpus

The AnCora dependency corpus consists of one single ConLL-format file containing 95.028 words. Figure 1 displays a sample sentence *El Gobierno de España pidió hoy al Senado que someta a votación el acuerdo*, lit. 'The government of Spain asked today to-the Senate that they-put to the vote the agreement'.

1	**El**	**el**	**d**	**da**	**gen=m|num=s**	**2**	**_**
2	Gobierno	Gobierno	n	np	_	5	SUJ
3	**de**	**de**	**s**	**sp**	**for=s**	**2**	**_**
4	España	España n	np	_		3	_
5	**pidió**	**pedir**	**v**	**vm**	**num=s|per=3|mod=i|tmp=s**	**0**	**ROOT**
6	hoy	hoy	r	rg	_	5	CC
7	**al**	**al**	**s**	**sp**	**gen=m|num=s|for=c**	**5**	**CI**
8	Senado	Senado	n	np	_	7	_
9	**que**	**que**	**c**	**cs**	**_**	**10**	**_**
10	someta	someter	v	vm	num=s|mod=s|tmp=p|per=1	5	CD
11	**a**	**a**	**s**	**sp**	**for=s**	**10**	**CREG**
12	votación	votación	n	nc	num=s|gen=f	11	_
13	**el**	**el**	**d**	**da**	**gen=m|num=s**	**14**	**_**
14	acuerdo	acuerdo	n	nc	gen=m|num=s	10	CD
15	**.**	**.**	**F**	**Fp**	**_**	**5**	**PUNC**

Fig. 1: A sample AnCora-format structure

The first column is the position of the unit in the sentence; the second, the surface form of the unit; the third, its lemma; the fourth and the fifth respectively the deep and the surface part-of-speech (POS); the sixth is an aggregation of features such as gender, number, person, depending on the POS of the unit; the seventh column is the position of the governing node, and the eigth the label of the relation with this governor.

The degree of detail and the number of the syntactic relations used in AnCora is much inferior to the set of SSynt-relations (SSyntRels): in total, 17 different labels, corresponding to about 12 of our 64 different SSynt-relations,[3] are used.[4] However, it has all syntactic dependencies marked explicitly (see below) – even if most of them are unlabelled and some of them are incorrect. In other words, each node in the annotation, except the root, has a governor. This is of great help for mapping AnCora structures onto SSyntSs. Thus, if we know that a determiner is a dependent on a noun, the relation is very likely to be *determinative*. Because this relation is not annotated in AnCora and is very frequent, being able to introduce it automatically saves a lot of time.

2.3 Annotation procedure

The annotation of a corpus with SSyntSs follows a number of basic rules which mainly originate from the notion of dependency, the characteristics of an SSyntS in MTT and considerations for further use of the SSyntS-annotated corpus:

(i) A well-formed SSyntS must be a connected tree where every node but the root must be the target of one and only one syntactical arc.

(ii) Although SSyntSs are order-free, the nodes are ordered for future machine learning applications.

(iii) The subject must be a dependent of the verbal root of a sentence structure. For instance, in *Gerard ha dejado su piso* 'Gerard has left his flat', *Gerard* is the subject of the auxiliar *ha* and not of the participle *dejado*, unlike the direct object: *Gerard* ← **subj**–*ha*–**analyt_perf**→*dejado*–**dobj**→*piso*–**det**→*su*.

(iv) Equally, the head of a relative clause is its main verb. Since an axiom of the theory is that each lexeme should correspond to one node and only one node in the tree, the relative pronoun is seen from the perspective of its function in the relative clause and not from the perspective of its

[3] See Appendix for the list of SSynt relations we use for annotation.

[4] According to the authors of the AnCora corpus, it is currently being enriched.

conjunctive properties. For instance, the phrase *Igor, que duerme* 'Igor, who sleeps' is represented as *Igor*–**relat**– **[***que*]→ *duerme* and *duerme*–**subj**→ *que*.

(v) A further consequence of the above axiom is that lexemes that occur within the same unit have to be separated. For example, *del* 'of.the' has to be split into *de*+*el* 'of+the', *haberlo* 'have.it' into *haber*+*lo* 'have'+'it', etc.

Many of these cases are not handled the same way in AnCora, which is why special attention must be paid during the SSyntS-annotation.

Another important point is that there must not be any ambiguity of the valency patterns in the SSyntS so as to facilitate the annotation at the more abstract levels DSyntS and Sem and derivation of a Government Pattern (GP) dictionary. To ensure this, we introduce several SSyntRels for the same grammatical function but different underlying GPs; for instance, *obl_obj1*/*2*/*3* for indirect objects (with the index marking the corresponding semantic actant slot). This allows us to obtain GPs by retrieving the corresponding DSyntS information without any ambiguity, and then (partially) derive the DSyntSs from the SSyntSs. However, given that for other purposes such a fine-grained and semantically oriented annotation is not appropriate (e.g., for training of a syntactic dependency parser), we maintain in parallel a version of the annotation in which only purely syntactic relations appear, i.e., in which *obl_obj1*/*2*/*3* relations are merged into the single relation *obl_obj*.

The annotation procedure that draws upon the above rules comprises several stages:

(1) Automatic projection of the annotations of the 3.512 sentences from AnCora onto rudimentary SSynt-like structures. This stage consists of two substages:
   (1a) A simple script maps in a one-to-one fashion AnCora relations/features onto SSynt-like relations/features.
   (1b) Using the graph transduction workbench MATE (Bohnet et al., 2000; Bohnet, 2006), inference rules derive from the topology of the AnCora structure additional SSynt-relations that are not available in AnCora.

(2) Manual revision of the structures obtained in Stage 1 by a team of grammarians, who follow detailed guidelines. For the revision work, MATE's graph editor is used.

(3) Training of a machine learning-based dependency parser (Bohnet, 2009) with the obtained SSyntSs and its application onto a new subcorpus of about 3.000 sentences.

(4) Manual revision of the structures obtained in Stage 3 and extension of the parser training corpus by these structures (cf. Hwa (2001) for more details and references on this particular method);

(5) Repetition of Stages 3 and 4 until the SSyntS annotated corpus reached the desired size. With each iteration, the quality of the parsing results improves, such that the cost of the manual revision decreases considerably.

During Stage (1a), the goal is thus to simply convert all labels – attribute/value pairs and arcs – into labels used in MTT's SSyntSs. For instance, the subject relation "SUJ" becomes "subj", the direct object relation "CD" becomes "dobj", the determinative POS feature "d" becomes the feature/value pair "spos=determiner" and so on. To facilitate higher quality parsing, we also introduce the POS tags from the Penn Treebank set (Mitchell *et al*., 1993). A simple script handles such one-to-one correspondences and provides intermediate ConLL-structures with appropriate tags (not shown here because too similar to Fig. 1; cf. Fig. 2 for graphical representation). The modified AnCora structure is imported into MATE's graph editor, where all dependency relations and the precedence relations (relations "b") as available in the ConLL structure can be visualized; cf. Fig. 2.

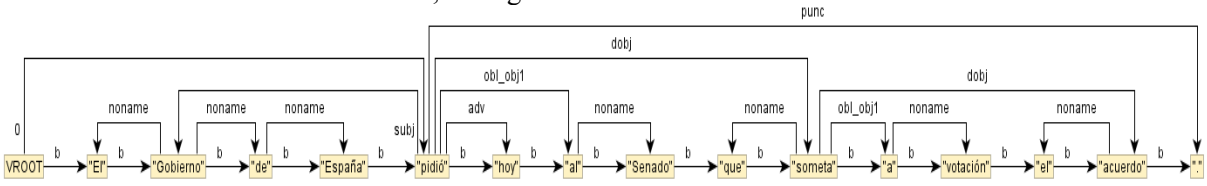

Fig. 2: Graphical representation of an AnCora Structure converted into a preliminary SSyntS

All empty relation names are mapped to “noname” labels. We see that there are quite a few arcs that are not labelled; that *al* ‘at.the’ is one single node; that some labels are wrong (thus, “obl_obj1” stands for actant 2 and here both dependents of “obl_obj1” are, in fact, actant 3 of their governing verb); and that some dependencies are erroneous – as, e.g., the conjunction *que*, which should govern the main verb of the subordinated clause and not be a dependent of it.

The second mapping (Stage 1b), performed automatically by using a small graph-transformation grammar of 55 rules in the MATE workbench, corrects some of these errors. Most of the rules simply check in the AnCora structure the nature of two nodes linked by arcs labelled “noname” and introduce an SSyntRel. Consider for instance the rule that introduces the *appos*(itive) relation:

```
?Xl {   dpos=N                                  rc:?Xr {<=> ?Xl
        noname→?Yl {dpos=N}        ➔            appos→ rc:?Yr {<=> ?Yl}}
        b→ ?Yl}
```

This rule states that if two nodes ?Xl and ?Yl that have the same deep part-of-speech N are linked by an arc “noname”, and if ?Yl follows ?Xl, then an arc *appos* is added from ?Xl to ?Yl in the target structure (the ‘rc:’ prefix in the right hand side of the rule is due to internal MATE codification conventions and can be ignored here). Other types of rules handle the separation of nodes or check the root of a verb group. Applied to the structure in Fig. 2, the transformation grammar gives us the following structure in Fig. 3:

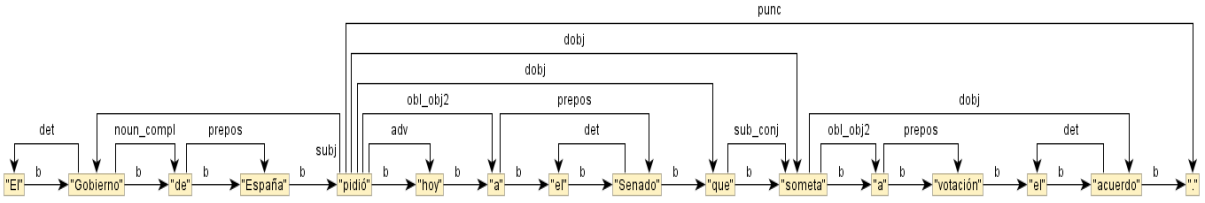

Fig. 3: Structure after stage (1b)

The *al* node is now split into a preposition node and a determiner node; all relations are added; all arcs are labelled with a dependency relation; the “obl_obj1” labels have been mapped to “obl_obj2”, which corresponds to the actant 3 slot of the governor; the dependency that we expect between the conjunction *que* and the verb *someta* ‘put’ is now visible, hence the relation “dobj” between the governor *pidió* and this conjunction. However, it can be also observed that the automatic mapping introduces new errors into the annotation, such as multiple edges, which must be corrected manually in Stage 2.

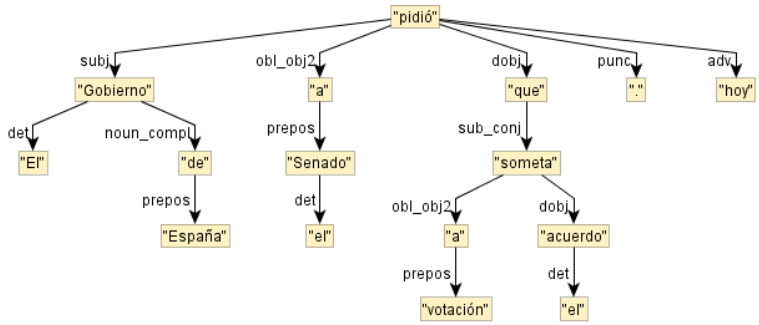

Fig. 4: Correct SSynt dependency tree

For such simple structures as the one in the figures above, already the first mapping is very efficient and the manual corrections can thus be kept to the minimum. In this particular case, just one arc has to be

removed to get the final structure (shown without the precedence relations in Fig. 4 above).[5] When the structure is more complex, there are, of course, more errors, be it in the original corpus or during the second mapping in Stage 1b. The main errors for each level are detailed in the next section.

Stage 2 is carried out by a team of annotators trained in MTT. In order to ensure a high quality annotation, structures annotated by one annotator are verified by two other annotators.

Once the 3512 sentences from the original AnCora corpus have been annotated with SSyntSs, the training of the machine learning based parser starts. The training algorithm implemented by B. Bohnet (2009) delivers models for a parser that reached an accuracy of about 96% for German (with respect to both dependency links and labels); we are confident that we reach a similar accuracy for Spanish.

Stage 2 has recently been completed. In order to ensure a high quality annotation, structures annotated by one annotator are currently cross-checked by two other annotators. We expect this procedure to be finished by September 2009.

3 Assessment of the annotation

In order to be able to assess the costs of the annotation of a corpus with such detailed dependency information as SSyntSs, it is essential to be aware of the errors encountered at the different stages of the annotation procedure as well as of the manual workload envisaged by the annotators.

3.1 Error evaluation

During Stage (1b) of our annotation procedure, two main types of errors that directly influence the manual workload of the annotators are introduced: (i) wrong choice of actants, especially for nouns, and (ii) over-generation of arcs. Errors of type (i) arise because it is impossible to know from the syntactic structure to which semantic argument a syntactic actant corresponds. In Spanish NPs, actants of a governing noun are related to it by the preposition *de*: *lista de paro*, *presidente de+l gobierno*, etc. Therefore, in the AnCora-SSyntS mapping, only one rule introduces nominal actantial relations. By default this is the relation that corresponds to the first actant, i.e. nominal completive – as, e.g., encountered in *una lista de escuelas* 'a list of schools'. However, in many cases, it is actually the second actant (as in *el presidente de Francia* 'the president of France'), or even a third or a fourth, or an attribute, i.e., not an actant at all (as in *mesa de madera* 'wooden table'). That is, the annotator must pay close attention to this phenomenon in particular.

Errors of type (ii) are due to the fact that the application of the mapping rules is not sufficiently constrained. Indeed, the rules are preferred to apply even in uncertain cases in order to avoid that they miss some relations: for the annotator, it is easier and faster to remove an arc than to add a new one.

Although the mapping during Stage (1b) introduces errors, it also corrects suboptimal (from the point of view of SSyntSs) choices made by the AnCora annotators, such as leaving many dependency arcs unlabeled (see footnote 4 above) or treating some word combinations as single units, even if they are not at the syntactic level, as shown *supra*.

Several annotation characteristics of the AnCora corpus also required massive manual intervention on our part because they could not always be handled by mapping grammar rules. The most significant of them are:

- an adjective positioned before a noun is considered the head of the adjectival phrase a part of which is the noun; accordingly, the adjective is considered governor of various dependents of the noun: in MTT, the noun is the governor of its adjectival modifiers;
- non-finite verbal heads in auxiliary constructions and raising/control constructions are considered to be syntactic heads of verb groups: although it is the *semantic* head of the group, in syntax, the finite verb has to be the governor;
- the coordinate constructions have been partially left aside;

[5] For annotators' convenience, the "linearized" trees can be shown in the tree format – for instance, to facilitate the connectivity check.

- the internal dependencies in relative clauses are often missing.

So, knowing this, what is the amount of work that an annotator has to invest in order to carry out his/her task? Let us assess this in the next subsection.

3.2 Extent of the manual workload

In order to carry out a preliminary evaluation on the manual workload, we picked randomly 50 sentences out of our annotated set and manually counted the modifications that had been performed by the annotator. Three types of manipulations have been identified, by order of descending complexity: (1) create nodes (includes creating and labelling arcs); (2) create or move an arc (includes labelling the arc); (3) label an arc that is correctly positioned.

We counted 9 interventions of type 1, 366 of type 2, and 77 of type 3. This gives an average of about 0.2 creations of nodes, 7.3 creations of arcs, and 1.5 arc re-labellings per annotated sentence. While these figures seem low compared to what is usually needed to annotate sentences, it should not be forgotten that what takes more time is not editing a graph, but elaborating all the dependencies between the units of the sentence. The process is certainly made much easier, but the workload remains important.

4 How to obtain a DSyntS annotation?

As already mentioned, the richness of the SSynt dependencies makes the SSyntS very informative. In this section, we show that it grants direct access to DSyntSs.

As illustrated in the previous sections, our SSynt annotation foresees different names for the same syntactic dependency, depending on the valency slot occupied by the dependent. In the SSyntS of Fig. 4, for instance, we find four predicates, *gobierno* 'government', *pedir* 'ask', *someter* 'put'[6] and *acuerdo* 'agreement'. What can be deduced from Fig. 4 is that:

- *gobierno* has an actant 1 (realized here by the SSyntRel 'noun completive');
- *pedir* has an actant 1 ('subjectival'), an actant 2 ('direct objectival'), and an actant 3 ('oblique objectival 2');
- *someter* has an actant 2 ('direct objectival'), and an actant 3 ('oblique objectival 2'); the first actant does not have to be realized;
- *acuerdo* appears without any actant realized. We cannot draw any conclusion with respect to its actant structure.

A simple mapping grammar in MATE extracts this lexical information from the SSyntS in Fig. 4 in terms of the following lists of attributes, corresponding to the "syntactic combinatorial zone" as described in (Mel'čuk, 2006):

– *gobierno* { **dpos**=N
I_dpos=N **I_spos**=proper_noun **I_rel**=noun_compl **I_prep**=de }
– *pedir* { **dpos**=V
I_dpos=N **I_spos**=proper_noun **I_rel**=subj
II_dpos=V **II_spos**=verb **II_rel**=dobj **II_prep**="que" **II_mood**=SUBJ **III_dpos**=N **III_spos**=proper_noun **III_rel**=obl_obj2 **III_prep**="a" }
– *someter* { **dpos**=V
II_dpos=N **II_spos**=noun **II_rel**=dobj
III_dpos=N **III_spos**=noun **III_rel**=obl_obj2 **III_prep**="a" }
– *acuerdo* { **dpos**=N **ssynt_actant**=NO }

Pedir 'ask', for instance, contains four lines of attribute/value pairs: in the first line appears its deep part-of-speech (*dpos*); the second line presents the information corresponding to its first DSynt actant: this actant is a proper noun, linked by the relation "subj" to its governor; in the third line, the information about the second DSynt actant is stored: it is a verb linked to *pedir* by a direct objectival relation 'dobj',

[6] *Someter* cannot not generally be translated by 'put'; here, it is, actually, the value of a lexical function; see below.

such that this verb is introduced by *que* 'that' and is in the subjunctive mood. Similarly, the last line describes the third actant of *pedir*.

Of course this is not the only way to use the verb *pedir* in Spanish. First, there are other senses corresponding to *pedir*, such as 'order' (a coffee) or 'beg' (for money). These cases are left aside for the moment since those instances of *pedir* are different lexical units, and thus also different entries in the dictionary. Second, for the same lexical unit meaning 'ask' – let us call it *pedir_1* – there are several GPs that can be realized; consider the following sentences that exemplify a variety of partial GP instantiations:[7]

(1) *El jefe le pidió a Elena que escribiera ese informe.* 'The boss asked *PREP* Elena that [she] wrote the report': actant II is a subjunctive verb with governed conjunction;
(2) *El jefe le pidió a Elena escribir ese informe.* 'The boss asked *PREP* Elena [to] write the report': actant II is an infinitive verb without preposition;
(3) *Le pidió un favor a Elena*. '[He/she] asked [for] a favour to Elena': actant II is a noun; no actant I is visible.

All GPs will eventually appear in the entry for *pedir_1* in the lexicon; how this is achieved is beyond the scope of this paper. What matters here is that any GP of any lexical unit can be stored in the dictionary, with all properties of the governed element that are required by the governor (POS, mood, finiteness, etc.), and so on.

With such a dictionary at hand, it is very easy to derive DSyntS since one of the main challenges of the SSynt-DSynt transition is to distinguish semantic prepositions from syntactic (*governed*) prepositions; the latter being the only ones stored in the dictionary. For example, in *Marc le pidió a Elena que le llamase por teléfono*, lit. 'Marc [her] asked to Elena that him she.calls by [the] phone', the meaningless governed preposition *a* 'to' does not appear in the DSyntS (neither does *que* 'that'), whereas the preposition *por* 'by', which has a meaning, namely the way how Marc asked Elena to call, has to appear as a node label in the DSyntS.

In the case of the SSyntS in Fig. 4, using such a dictionary, we can readily derive a DSyntS shown at the left hand side of Fig. 5. This DSyntS is "nearly" correct. It is not entirely correct because it does not take into account the notion of *lexical function*, LF (Mel'čuk, 1996). Thus, the verb *someter* is, in fact, the value of an LF, namely CausOper_2 applied to the keyword *votación*; cf. the right hand side DSyntS in Fig. 5.[8] In other words, in order to annotate DSyntSs appropriately, LFs have to be introduced directly by the annotator;[9] however, the total amount of work necessary for the compilation of a DSyntSs corpus remains rather low once the SSyntSs corpus has been built.

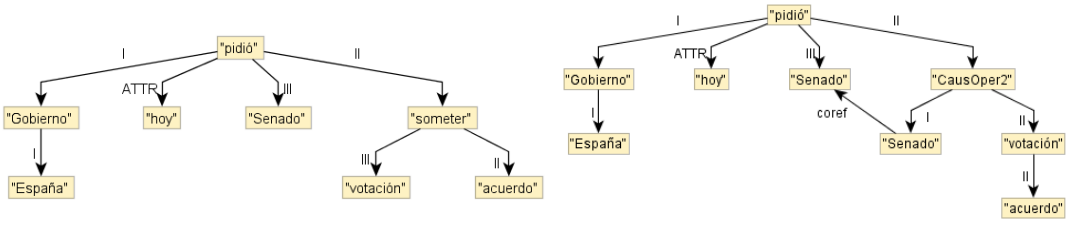

Fig. 5: Automatically derived (left hand side) and correct (right hand side) DSyntS corresponding to SSyntS in Fig. 4

Once the DSyntS annotation is in place, we can approach the SemS annotation, which will be perforce a simplified SemS, without lexical decomposition.

---

[7] Only actant II is detailed since the first and the third are the same in these instances of this lexical unit.
[8] Due to the limitations of the graphical editor, the theoretically bidirectional coreference relation 'coref' between the two nodes of "Senado" is shown as uni-directional.
[9] The work on the automatic recognition of LFs in corpora as discussed, e.g., in (Wanner et al., 2006) is still too preliminary to be used for high quality annotation.

5 Conclusions and Future Work

We presented a cost effective strategy for the creation of a mid-size fine-grained dependency treebank of MTT's SSyntSs for Spanish. The strategy draws upon a small seed dependency corpus, the AnCora corpus, whose annotation is considerably more coarse-grained than our target annotation. We have shown that this discrepancy can be bridged largely by automatic means, relying upon contextual information and leaving thus minimal work to the annotators. This facilitates the development of resources with limited human effort, within a limited period of time. The availability of the SSynt treebank will allow us to pursue research in a number of different directions. For instance, once annotations of several layers are available, we can use machine learning techniques for automatic learning of sentence generation or analysis grammars.

Acknowledgements: We are grateful to Antònia Martí for making the AnCora corpus available to us. Many thanks also to Igor Mel'čuk for his generous help with respect to all kinds of problems related to dependency and for his helpful comments on this paper, to our colleagues in the TALN group, especially Gabriela Ferraro, for their support, and to the two anonymous reviewers for their helpful comments. All remaining errors and omissions are, as always, our full responsibility.

References

Afonso, S., Bick, E., Haber, R., and Santos, D. (2002). "Floresta sintá(c)tica": A treebank for Portuguese, in M. González Rodríguez and C. Paz Suárez Araujo (eds.), In *Proceedings of LREC*, 29-31. Las Palmas de Gran Canaria, Spain. ELRA, pp.1698-1703.

Apresjan, Ju., Boguslavsky, I, Iomdin, B., Iomdin, L., Sannikov, A., and Sizov. V. (2006). A Syntactically and Semantically Tagged Corpus of Russian: State of the Art and Prospects. In *Proceedings of LREC*, 1378-1381. Genova, Italy.

Beek van der, L., G. Bouma, R. Malouf, and G. van Noord. (2002). "The Alpino dependency treebank". In *Linguistics and Computers. Selected Papers from the 12th CLIN Meeting*. Twente, The Netherlands, 8-22

Bohnet, B., A. Langjahr and L. Wanner. (2000). "A Development Environment for an MTT-Based Sentence Generator". *Proceedings of the First International Conference on Natural Language Generation*. Mitzpe Ramon, Israel, 260-263

Bohnet, B. (2003). "Mapping Phrase Structures to Dependency Structures in the Case of Free Word Order Languages". In *Proceedings of MTT conference 2003*, Paris

Bohnet, B. (2006). *Textgenerierung durch Transduktion linguistischer Strukturen*. DISKI 298. Akademische V. G., Berlin.

Bohnet, B., (2009). "Synchronous Parsing of Syntactic and Semantic Structures". To appear in *Proceedings of the Fourth International Conference on Meaning-Text Theory*, Montreal.

Brants, S.; Dipper, S.; Hansen, S.; Lezius, W. and Smith, G. (2002). "The TIGER Treebank". In *Proceedings of the Workshop on Treebanks and Linguistic Theories*. Sozopol.

Gelbukh, A., Torres, S. and Calvo, H. (2005). "Transforming a Constituency Treebank into a Dependency Treebank". In *Proceedings of the X Conference of the Spanish Association for Artificial Intelligence*.

Hajič, J. et al. (2006). *Prague Dependency Treebank 2.0* . In Linguistic Data Consortium, Philadelphia.

Herrera, J., *et al* (2007). "Building Corpora for the Development of a Dependency Parser for Spanish Using Maltparser". In *Procesamiento del Lenguaje Natural*, nº39, pp. 181-186. Spain.

Hwa, R. (2001). On minimizing training corpus for parser acquisition. In *Proceedings of the Fifth Computational Natural Language Learning Workshop*, Toulouse, France, July.

Johansson, R. and Pierre Nugues (2007). "Extended Constituent-to-dependency Conversion for English." In *Proceedings of NODALIDA 2007*. Tartu, Estonia.

Kakkonen, T. (2006). DepAnn - An Annotation Tool for Dependency Treebanks. In *Proceedings of the 11th ESSLLI Student Session at the 18th European Summer School in Logic, Language and Information*, pp. 214–225. Malaga.

Martí, M.A., Taulé, M., Márquez, L., Bertran, M. (2007): "Ancora: A Multilingual and Multilevel Annotated Corpus", http://clic.ub.edu/ancora/publications/

Mel'čuk, I.A. (1988). *Dependency Syntax: Theory and Practice*, Albany, N.Y.: The SUNY Press.

Mel'čuk, I.A. (1996). Lexical Functions: A Tool for the Description of Lexical Relations in a Lexicon. In L. Wanner (ed.) *Lexical Functions in Lexicography and Natural Language Processing*. Amsterdam: Benjamins.

Mel'čuk, I.A. (2003). Levels of Dependency in Linguistic Description: Concepts and Problems. In V. Agel, L. Eichinnger, H.-W. Eroms, P. Hellwig, H. J. Herringer, H. Lobin (eds): *Dependency and Valency. An International Handbook of Contemporary Research,* vol. 1, Berlin - New York, W. de Gruyter, 188-229

Mel'čuk, I.A. (2006). Explanatory Combinatorial Dictionary. In G. Sica (ed.). Open Problems in Linguistics and Lexicography. Monza, Italy: Polimetrica, 225-355.

Mitchell P. M., B. Santorini, and M.A. Marcinkiewicz (1993). "Building a Large Annotated Corpus of English: The Penn Treebank", In *Computational Linguistics*, 19(2):313– 330.

Mitchell P.M., B. Santorini, M.A Marcinkiewicz, and A. Taylor (1999). "Treebank-3", LDC, Philadelphia.

Nilsson, J., J. Hall and J. Nivre. (2005). "MAMBA Meets TIGER: Reconstructing a Swedish Treebank from Antiquity". In *Proceedings of NODALIDA 2005 Special Session on Treebanks for Spoken and Discourse*, Copenhagen Studies in Language 32, Joensuu, Finland, pp. 119-132.

Surdeanu, M., R. Johansson, A. Meyers, L. Màrquez, and J. Nivre (2008). "The CoNLL-2008 Shared Task on Joint Parsing of Syntactic and Semantic Dependencies." In *Proceedings of the 12th Conference on Computational Natural Language Learning* (CoNLL).

Wanner, L., B. Bohnet, M. Giereth, and V. Vidal. (2006). "The first steps towards the automatic compilation of specialized collocation dictionaries". In *Terminology*, 11(1):143-180.

Appendix: List of the 64 SSyntRels used for the annotation (inspired by Mel'čuk 2003).

For parser training: ♣=SSyntRel is merged with *adv*; ♦=SSyntRel is merged with *obl_obj*; consecutive♠=SSyntRels are merged together.

adjunct (adjunctive): Vale/Juan,<-*adjunct*- vamos; Pero, <-*adjunct*-[no lo]-sabían; por <-*adjunct*-[ejemplo,]-considera [esta solución]; [Mañana, vendrá] al_menos <- *adjunct* –Pedro;
adv (adverbial): etiquetar-*adv*->rápidamente; volvió-[el]-*adv*->día [siguiente]; volver-*adv*->corriendo; aproximadamente <-*adv*-veinte; [sabe] cuándo<-*adv*-viene;
adv_abs (absolutive adverbial): Terminada <-*adv_abs*-[la guerra]-,-volvieron [a casa]; el pan<-a*dv_abs*-[en la mano]-,-salió; vi a Vasco-,-*adv_abs*-> guitarra [en mano];
adv_clitic (clitic adverbial): producir-*adv_clitic*->le [otra nidada]; le <-*adv_clitic*-plantó [un árbol];
adv_mod (modificative adverbial): [Muy] tranquilo,<- *adv_mod*-viajaba [a menudo];
adv_obj1♣ (objectival adverbial 1): viene-*adv_obj1*->aquí; va- *adv_obj1*-> adentro; [me] siento- *adv_obj1*->bien;
adv_obj2♣ (objectival adverbial 2): [Lo he] traído-*adv_obj2*->aquí; [lo] calificó-*adv_obj2*-> positivamente;
agent (agentive): escrito-*agent*->por [Leo];
analyt_fut (future analytical) : va-*analyt_fut*-> a conducir;
analyt_pass (passive analytical) : es-*analyt_pass*-> conducido;
analyt_perf (perfect analytical): ha-*analyt_perf*-> conducido;
analyt_progr (progressive analytical): está-*analyt_progr*-> conduciendo;

appos (appositive): [el] presidente-*appos*-> Obama; [la] nebulosa-*appos*-> de [Orion];
appos_descr (descriptive appositive): [el] presidente,-*appos_descr*->Obama, […];
attr (attributive): casa-*attr*->sin [ventanas]; mesa-*attr*-> de [madera]; niño-*attr*-> con [gafas];
attr_descr (descriptive attributive): [el profesor] Wanner,-*attr_descr*->de [Barcelona, estuvo aquí];
aux_phras (phraseological auxiliary): lo_más-[claro]-*aux_phras*-> posible; tomar-*aux_phras*-> en [cuenta];
aux_refl (reflexive auxiliary): me<-*aux_refl*-afeito; se<-*aux_refl*-miran; se<-*aux_refl*-come [un conejo];
bin_junct (binary junctive): o<-[Barça]-*bin_junct*-o [Real]; desde-[dos]-*bin_junct*->hasta [cuatro];
compar (comparative): mejor-*compar*->que, tan-[bonito]-*compar*->como [alto/Juan];
compar_conj (comparative conjunctional) : [mejor] que-*compar_conj*->tú;
compl1♠ (completive 1): [La frase] resulta-*compl1*-> buena;
compl2♠ (completive 2): [Vane] encuentra-[la semántica]-*compl2*-> fácil;
compl_adnom (adnominal completive): Los-*compl_adnom*-> de [la ciudad];
coord (coordinative): sentido-*coord*->y [texto], dependencia-*coord*->o [constituyente];
coord_conj (coordinate conjunctional): [sentido] y-*coord_conj*-> texto;
copul (copulative): Igor es-*copul*->guapo/[un] hombre;
copul_clitic (clitic copulative): [Igor] lo<-*copul_clitic*-es;
det (determinative): este<-*det*-artículo; un<-*det*-gato; lo<-*det*-divertido [es que caí];
dobj (direct objectival): [Igor] come-*dobj*->gatitos; quiere-*dobj*->que [vengas]; [Leo] ve-*dobj*-> a [Marga];
dobj_clitic (direct objectival clitic): haz-*dobj_clitic*->lo; la <-*dobj_clitic*-mira;
dobj_quot (quotative direct objectival): ha gritado-*dobj_quot*-["¡]->Gooool!";
elect (elective): [el] mejor-*elect*->de [los pintores], [el] más-[tonto]-*elect*->en [Barcelona];
inf_obj1♠ (infinitival objectival 1): [Juan] piensa-*inf_obj1*->ganar; [su]deseo-*inf_obj1*->de [venir];
inf_obj2♠ (infinitival objectival 2): [lo] empuja-*inf_obj2*-> a [venir];
juxtapos (juxtapositive): Es-[muy potente:]-*juxtapos*-> puede [destruir un país en 10 minutos];
modal (modal verb): [Manuel] puede-*modal*-> venir (Closed list: poder, deber, querer);
modif (modificative): gato-*modif*->pelado; pequeño<-*modif*-árbol ; [un] chico-*modif*->más;
modif_abs (absolutive modificative): [los] gatos-,-*modif_abs*-> incluidos [los negros, me gustan];
modif_descr (descriptive modificative): [las] ventanas,-*modif_descr*->sucias [y rotas, se caen];
noun_compl♦ (noun completive): [las] gafas-*noun_compl*-> de [Pep]; [una] traducción-*noun_compl*-> de [Stefan]; falta-*noun_compl*-> de [chocolate]; aceite-*noun_compl*->de [oliva];
num_junct (numeral junctive): treinta-*num_junct*->y [tres]; tres<-*num_junct*- mil (3000);
obj_copred (object copredicative): [Igor] quiere-[la estructura]-*obj_copred*-> conectada;
obl_obj1♠ (first oblique object): ir-*obl_obj1*-> a [la playa]; tener-*obl_obj1*-> que [comprar]; presidente-*obl_obj1*-> de [Francia]; traducción- *obl_obj1*-> de [este texto]; gracias-*obl_obj1*-> a [Leo]; capaz-*obl_obj1*-> de [hablar];
obl_obj2♠ (second oblique object): vender-[un disco]-*obl_obj2*->a [Marc por 10 €]; suplemento-*obl_obj2*-> de [economía de la Vanguardia];
obl_obj3♠ (third oblique object): vender-[un disco a Marc]-*obl_obj3*-> por [10 €];
obl_obj_clitic1♠ (first oblique object clitic): [la virgen se] le <-*obl_obj_clitic1*-aparece [cada miércoles];
obl_obj_clitic2♠ (second oblique object clitic): dar-*obl_obj_clitic2*->le; le <-*obl_obj_clitic2*-da;
prepos (prepositional): en-*prepos*-> cama;
prolep (prolepsis): Yo<-*prolep*-,- [lo que veo]-es [una torre];
quant (quantitative) : [tres] mil <-*quant*-personas;
quasi_coord (quasi coordinative): a[l]-[norte]-,- *quasi_coord*-> allí [donde casi nadie mira];
quasi_subj (quasi subjectival): Eso<-*subj*-sí-,- *quasi_subj*-> que [venga mañana];
relat (relative): [el] gato-[que]-*relat*-> está [aplastado]; [el] edificio-[en que]-*relat*-> trabajamos;
relat_descr (descriptive relative): [este] artículo,-[que]-*relat_descr*->leí [ayer, es corto]
relat_expl (explicative relative): Juan salta-,-[lo_que]-*relat_expl*->sorprende [a su madre];
restr (restrictive): más <-*restr*-frecuente; no <-*restr*-bebe; sólo <-*restr*-fuma;
sequent (sequential): cuarenta-*sequent*-> hasta [cincuenta]; [el partido] Real-*sequent*-> Barça [terminó 2-6];
sub_conj (subordinate conjunctional): [es verdad] que-*sub_conj*-> [ayer llovió];
subj (subjectival): [el] sol <-*subj*-baja; parece-[imposible]-*subj*-> ver [este detalle]; es-[a Verónica]-*subj*-> que [le he escrito]; claro/sí- *subj*-> que [te llamaré];
subj_copred (subject copredicative): [Mariano] volvió-*subj_copred*-> rico;
subj_quot (quotative subject): 'Algo' <-*subj_quot*-es [sujeto de esta frase];
+ (only for parser training) **punc** (punctuation)/**punc_init** (initial punctuation)