Procesamiento del Lenguaje Natural, núm. 43 (2009), pp. 325-333 recibido 1-05-2009; aceptado 5-06-2009

Towards a Rich Dependency Annotation of Spanish Corpora

Hacia una anotación de dependencias enriquecida de corpus españoles

Simon Mille, Alicia Burga, Vanesa Vidal
Barcelona Media, Universitat Pompeu Fabra
Av. Diagonal, 177, 08018 Barcelona
simon.mille@upf.edu

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

Abstract: We present a cost-effective strategy for the creation of a mid-size fine-grained Spanish dependency tree bank of surface-, deep-syntactic and semantic structures as defined in the Meaning-Text Theory. 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. 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.

Keywords: corpus annotation, dependency, meaning-text, surface syntax, deep syntax, Spanish, treebank

Resumen: En este artículo presentamos una estrategia de bajo coste para la creación de un corpus de estructuras sintácticas (tanto superficiales como profundas) y semánticas, tal y como son definidas en la Teoría Sentido-Texto. El corpus es de tamaño medio, pero muy preciso y detallado. La estrategia parte de un pequeño corpus de dependencias, el corpus AnCora, cuya anotación es mucho menos detallada que la nuestra. Mostramos que la discrepancia entre ambas anotaciones se puede salvar en gran medida a través de medios automáticos, lo cual permite que los recursos necesarios se desarrollen en poco tiempo y con un esfuerzo humano limitado. Asimismo, proponemos una evaluación preliminar de la cantidad de trabajo requerido en términos reales en el proceso de anotación.

Palabras clave: anotación de corpus, dependencia, sentido-texto, sintaxis superficial, sintaxis profunda, español, base de datos de arboles

1 Introduction

Dependency structure annotated corpora proved to be a valuable resource for several NLP tasks. Such corpora are already available for a number of languages; cf., e.g., the dependency version of the Penn Tree Bank corpus (Mitchell *et al.*, 1999) for English, the Prague Dependency Treebank for Czech (Hajič *et al.*, 2006), the Portuguese Bosque corpus (Afonso *et al.,* 2002), the Dutch Alpino tree bank (van der Beek et al., 2002), etc. However, for Spanish, so far only small dependency corpora have been created – among them, e.g., the dependency version of the Cast3LB corpus (Herrera *et al.* 2007.a) and particularly the AnCora corpus (Martí *et al.*, 2007). Our aim is to change this state of affairs and to create a mid-size rich dependency annotated corpus of Spanish which can be used for both theoretical linguistic studies and NLP.

The dependency relations we use for our annotation are as defined in the Meaning-Text Theory, MTT (Mel'čuk, 1988). MTT's linguistic model is a multistratal model. The main linguistic structure at each stratum is a dependency structure with the set of dependency relations ranging from six (at the *semantic* stratum) to over sixty (at the *surface-syntactic* stratum). Our ultimate goal is to have the corpus annotated with the structures of all primary strata. Currently, we focus on the annotation with surface syntax structures (*SSyntSs*), performing in parallel experiments on the annotation of structures of the other strata. In order to speed up the annotation procedure, we start from the AnCora corpus. Although AnCora's annotation is considerably more coarse-grained than SSyntSs and some of

ISSN: 1135-5948

its annotation conventions are incompatible with MTT, AnCora structures 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 give a quick overview of the MTT. Section 3 provides details of the annotation procedure with SSyntSs. Section 4 presents a preliminary assessment of the costs of the annotation. In Section 5, we show how SSyntSs can be used to obtain, in a relatively short time and with a relatively small effort, annotations at two deeper strata: the deep-syntactic and the semantic strata. Section 6, finally, provides some conclusions and a summary of our work.

2 Overview of MTT Dependency Structures

Compared to other linguistic theories, MTT is richly stratified. For written language, five strata are foreseen (Fig.1).[1] At each stratum, a clearly defined type of linguistic phenomena is described in terms of distinct dependency structures.

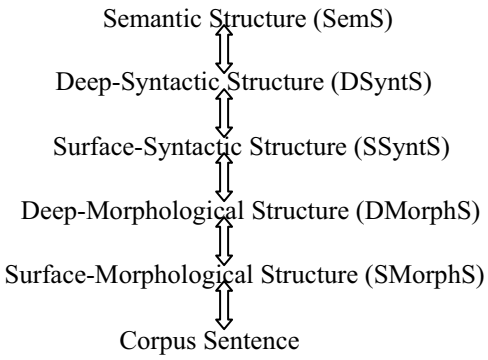

Figure 1: The linguistic model of the MTT

SemSs are predicate-argument structures where the relations between predicates and their arguments are numbered in accordance with the order of the arguments. DSyntSs are dependency trees, with the nodes labelled by meaningful ("deep") lexical units (LUs) and the arcs by actant relations I, II, III, …, VI (in accordance with the syntactic valency pattern of the governing LU) or one of the following three circumstantial relations: ATTR(ibute), COORD(ination), APPOS(ition). SSyntSs are dependency trees where the nodes are labelled by an open or closed class lexeme and the arcs by a grammatical function relation of the type *subject*, *oblique_object*$_i$, *adverbial*, *modifier*, etc. DMorphSs are chains of lexemes in their base form (with inflectional and POS features being associated to them in terms of attribute-feature pairs) between which the precedence relation 'b(efore)' is defined and which are grouped in terms of constituents. SMorphSs are chains of inflected word forms, i.e., sentences as they appear in the corpus, only that orthographic contractions still did not take place.

Fig.2 shows the structures of the sentence *El presidente ha bebido mucha agua* 'The president has drunk a lot of water' at the first four strata;[2] due to its proximity to the surface, the illustration of SMorphS is obsolete.

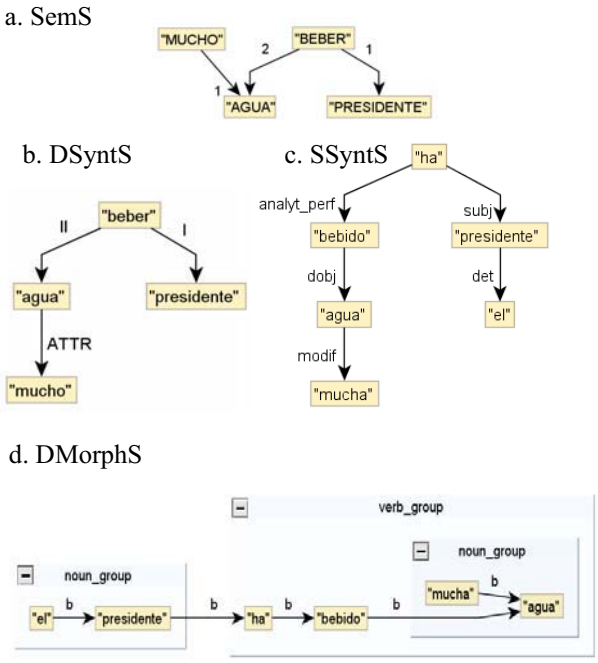

Figure 2: The variety of linguistic structures in an MTT-model

Fig.2 shows that SSyntSs are the most informative representations. Therefore, it makes sense to start the annotation with SSyntS. A

[1] The rich stratification facilitates a clear separation of different types of linguistic phenomena and thus an easier handling in the framework of such NLP-applications as automatic text generation.

[2] We omit the flexional feature-value structures assigned to the node labels in DSyntS and SSyntS. Furthermore, we simplify the SemS in that we do not specify, e.g., the time related information and show the nodes in the SSyntS as inflected, although, in the genuine SSyntS they appear as base forms.

fine-grained annotation with over sixty different relations seems too costly at the first glance; however, we show in this paper that it is not.

Once we have annotated the corpus with SSyntSs, we can derive nearly entirely automatically the DSyntS-annotation and from there the SemS-annotation (see also Section 5 below). This is possible due to the *n*:*m* correspondence that holds between the structures of pairs of adjacent strata (in Fig.1 indicated by a bidirectional arrow); cf. (Kahane, 2003).

3 The annotation procedure

Before we delve into the presentation of the AnCora corpus and our annotation strategy, let us assess the options that are in principle available to annotate a corpus with SSyntSs.

3.1 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:

A. Manually, from the scratch, i.e., starting from a raw corpus. This option would guarantee a high quality annotation (provided that the annotators are adequately trained and high degree of mutual agreement between the annotators is ensured), but is extremely costly.

B. 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, so far, not a single SSyntS-parser is available as yet. A solution could have been to use another parser, for instance, the JBeaver parser (Herrera *et al*, 2007b), mapping the obtained parse trees onto SSyntSs. However, the error rate of the current version of the JBeaver parser is quite high. In addition, its output structures are very different from SSyntSs – which implies additional noise during the phase of mapping.

C. 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*. (2007a) for the derivation of dependency annotations. They used the algorithm of Gelbukh *et al*. (2005) that is able to convert constituency structures into dependency structures. Bohnet (2003) performed a similar task on the German corpus NEGRA. The problem here is that it is quite difficult to obtain accurate output structures as soon as sentences become somewhat more complex.

D. Starting from an already existing dependency Treebank, mapping the available dependency structures onto SSyntSs. In general, this would imply that many SSyntS-relations will be missing and would 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.

In our work, we decided to adopt option D. The dependency Treebank from which we start is AnCora_DEP_ES (Martí *et al*., 2007), which comprises 3,512 sentences.

3.2 Our starting point: the AnCora corpus

The AnCora dependency corpus consists of one single ConLL08-format[3] file containing 95,028 words. Fig.3 displays a sample sentence *El documento propone que esta ley afecte a muchos trabajadores* lit. 'The document suggests that this law applies to many workers':

1	**El**	el	d	da	2	**-**
2	**documento**	documento	n	nc	3	SUJ
3	**propone**	proponer	v	vm	0	ROOT
4	**que**	que	c	cs	7	-
5	**esta**	este	d	da	6	-
6	**ley**	ley	n	np	7	SUJ
7	**afecte**	afectar	v	vm	3	CD
8	**a**	a	s	sp	7	CD
9	**muchos**	mucho	d	da	10	-
10	**trabajadores**	trabajador	n	nc	8	-
11	**.**	.	F	Fp	3	PUNC

Figure 3: A sample AnCora-format structure

Let us refer for the explanation of the format used in Fig.3 to the 7th unit of the sentence *afecte* 'applies': the first column is the position of the unit in the sentence (here: *7*); the second, the surface form of the unit, *afecte* (3rd person singular, subjunctive mood, present tense); the third, its deep form (lemma, *afectar*, infinitive);

[3] See Surdeanu *et al*. (2008)

the fourth and the fifth respectively the deep and the surface part-of-speech, or *POS*, *v* and *vm*; the sixth is the position of the governing node (here: *3*), and the eighth the label of the relation with this governor (*CD*).

The degree of detail and the number of the syntactic relations used in AnCora is much inferior to the set of SSyntS-relations: in total, 17 different labels, corresponding to about 12 of our 64 SSyntS-relations, are used.[4] However, it has all syntactic dependencies marked explicitly – even if most of them are unlabelled. In other words, each node in the annotation, except the root, has a governor. This is of great advantage for mapping AnCora structures onto SSyntS because in many cases, the POS of the governor and the governed nodes give us a clear hint on the type of the relation itself. This hint can be exploited in an automatic "post-annotation" stage. Thus, if we know that a determiner is a dependent of a noun, the relation is very likely to be *determinative*.

3.3 Annotation strategy

3.3.1 General annotation rules

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 syntactic arc.

(ii) Although SSyntSs are order-free, the nodes are ordered for future machine learning applications in that a precedence relation is defined between them.

(iii) The subject must be a dependent of the inflected top verb, not of the non-finite verb, which might also occur in the sentence. 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 the relative clause is its main verb. Since an axiom of the MTT says that every lexeme should correspond to one and only one node in the tree, the relative pronoun is viewed from the perspective of its function in the relative clause and not from the perspective of its conjunctive properties; e.g., 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.

In a considerable number of cases, AnCora's annotations are not in conformity with those rules, (iii), (iv) and (v) in particular. This is why special attention must be paid during the SSyntS-annotation

To facilitate the derivation of the annotation of DSyntSS and SemS as well as the derivation of a valency dictionary from SSyntS, it is also important that there is no ambiguity between the valency patterns in the SSyntS. To ensure that this restriction is observed, we introduce several SSyntRels for the same grammatical function; for instance, *obl_obj1*/*2*/*3* for indirect objects (with the index marking the corresponding semantic actant slot)[5]. This allows us to obtain valency structures of lexical items by retrieving the corresponding DSyntS information without any ambiguity, and then (partially) derive the DSyntSs from the SSyntSs.

3.3.2 Annotation procedure

The annotation procedure comprises in total five 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:

a. A simple script maps in a one-to-one fashion AnCora relations/features onto SSynt-like relations/features.

b. Derive from the topology of the AnCora structures additional SSynt relations that are not available in AnCora using inference rules implemented in the graph transduction workbench MATE (Bohnet *et al.* 2000; Bohnet, 2006).

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

[5] For training of the parser, these labels have to be generalized (*obl_obj* in this case).

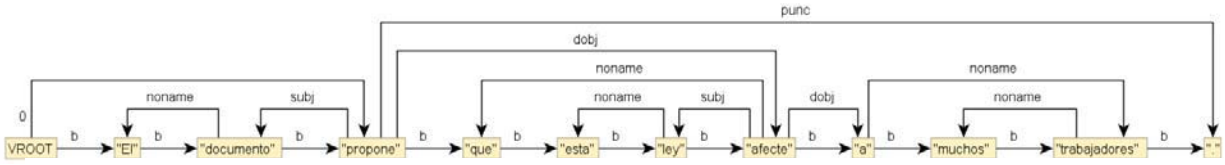

Figure 4: Graphical representation of an AnCora Structure converted into a preliminary SSyntS

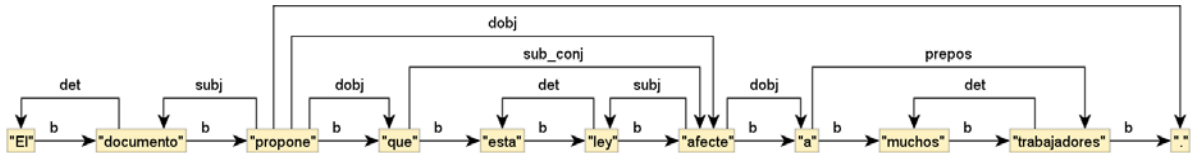

Figure 5: Ordered structure after stage (1b)

2. Manual revision of the structures obtained in Stage 1 in accordance with detailed guidelines. For the revision work, MATE's graph editor is used. Stage 2 is carried out by a team of annotators trained in MTT.
3. Training of a machine learning-based dependency parser 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.
5. Repetition of Stages 3 and 4 until the SSyntS annotated corpus reached the desired size. Since the quality of the parsing improves with each iteration, the cost of the manual revision decreases considerably and we expect the annotation to be much faster as the process follows its course.

During Stage (1a), the goal is thus to simply convert all labels – attribute/value pairs and arcs – into labels used in SSyntSs. For instance, the *subject* relation "SUJ" becomes *subj*, the *direct object* relation "CD" becomes *dobj*, the *determinative* POS feature "d" becomes the relation *det* and so on. To facilitate higher quality parsing, we decided to introduce furthermore the POS tags from the Penn Tree Bank set (Mitchell *et al*, 1993). A simple script handles those one-to-one correspondences and provides an intermediate CoNLL-structure with appropriate tags.

The slightly modified AnCora structure is then 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.4.

The second mapping (stage 1b), performed automatically using a small graph transformation grammar of 55 rules in the MATE workbench, gives the structure in Fig.5.

We can see in Fig.5 that the *del* node has been split and all relations added.

Most of the rules check in the AnCora structure the nature of two nodes linked by arcs labelled "noname". Consider for instance the rule that introduces the *appos*(ition) relation:

```
?Xl { dpos=N
      noname→?Yl {dpos=N}}
➔ rc:?Xr {<=> ?Xl
                appos→rc:?Yr{
          b→ ?Yl <=> ?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). Other types of rules handle the separation of nodes or the checking of the root of a verb group.

For annotators' convenience, the "linearized" trees can be shown in the tree format (non ordered nodes) – for instance, to facilitate the connectivity check.

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.

The next stage (Stage 3 above) will be the training of the machine learning based parser. The training algorithm implemented by B. Bohnet (2009) delivers models for a parser that reached an accuracy of about 81% for German

(with respect to both dependency links and labels) with a training set of the size of AnCora, i.e. 3,500 sentences. In other words, with the first iteration, we will get an error rate of 19% for the parser. We expect it to be similarly accurate for Spanish. The accuracy increases with each iteration over Stages 3 and 4 of our annotation procedure. Indeed, with 20,000 sentences in the training set, we expect a parser having about 88% of precision on labels and dependencies, which represents an error rate of 12%, hence an error reduction rate of about 37% when compared to the first iteration, in Step 3. At this point, the parser can be use as such and not only as a tool for the improvement of the annotation.

4 Assessment of the annotation procedure

For structures as simple as the one shown in the figures above, 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 obtain the final structure, i.e. the *dobj* arc between *propone* 'suggests' and *afecte* 'apply'. When the structure is bigger[6], there are, of course, more errors, be it in the original corpus or during the second mapping in Stage 1b. Let us analyze these errors.

4.1 Error evaluation

In order to be able to assess the costs of the annotation of a corpus with such a 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.

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 surface syntactic structures may be ambiguous in that they may express an actant relation and a modifier relation by an identical governor-dependent relation. Then, it is impossible to know which semantic argument is expressed by the dependent. For instance, in Spanish, different actants and modifiers of a noun are related to this noun via the preposition *de* 'of'; cf.: *lista de escuelas* 'list of schools' (where *escuelas* 'schools' is the first actant of *lista* 'list'), *presidente de Francia* 'president of France' (where Francia 'France' is the second actant of *presidente* 'president'), *mesa de madera*, lit. 'table of wood' (where *madera* 'wood' is an attribute of *mesa* 'table'), etc. During the AnCora-SSyntS mapping, however, only one rule introduces nominal actantial relations. By default, this is the relation of the first actant, i.e., *nominal completive*. Obviously, this choice leads to an error of type (i) if a different relation is at play (as in the second and third example above). The annotator, therefore, must pay close attention to detect these errors.

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 the choice to leave many dependency arcs unlabeled (see footnote 4 above), or the choice to treat some word combinations as single units although they are not at the syntactic level – as, e.g., *del* 'of.the', *haberlo* 'have.it', *14_de_mayo* '14th.of.may', *como_puede_ser* 'as.could.be', *abrirse* 'open.itself', etc.

Several annotation characteristics of the AnCora corpus required massive manual intervention on our part because they could not be handled by mapping grammar rules, partly due to restrictions of our rule editor. The most significant of them are:

1. An adjective positioned before a noun is considered the head of the adjectival phrase that includes the noun; accordingly, the adjective is considered governor of various dependents of the noun. However, in general – and in MTT in particular –, the noun is the governor of its adjectival modifiers.
2. As already pointed out above, non-finite verbal heads in auxiliary constructions and raising/control constructions are considered to be

[6] The longest sentence in the AnCora dependency corpus counts more than 140 words.

syntactic heads of verb groups: although these non-finite verbs are the *semantic* heads of the groups, we believe that in syntax the finite verbs have to be the governors.

3. The coordinate constructions have been partially left aside.
4. The internal dependencies in the relative clause are often missing.

So, knowing all this, what is the amount of work that an annotator has to invest in order to carry out his/her task?

4.2 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 categories of manipulations have been identified, by order of complexity:

- type 1: create nodes (includes creating and labelling arcs);
- type 2: create or move an arc (includes labelling the arc);
- type 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 7.3 creations of arcs, 1.5 arc re-labellings and 0.2 creations of nodes 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 since, as it has been mentioned, the set of SSyntRels that we use is quite large (more than sixty). The process is certainly made much easier, but the workload remains important.

As a result, according to our estimations and based on the work that has been done so far, an adequately trained full time annotator is able to annotate with good quality fifty sentences or revise at least a hundred structures per day. Theoretically, one annotator should then be able to annotate around 1,100 sentences per month of work (22 days/month), excluding revision. Taking into account the repartition of the tasks and the discussions between the annotators, it seems reasonable to foresee, for a group of 3 annotators, an average of 2,000 wholly annotated and revised structures per month.

5 Surface Syntax as a starting point for derivation of other layers of annotation

As already mentioned above, the richness of the SSynt dependencies makes the SSyntS very informative. In this section, we show that it grants direct access to deeper – more semantic – levels of representation. For this purpose, we use valency dictionaries derived from SSyntSs.[7]

Consider a partial valency entry for the verb *afectar* 'apply':

```
afectar {II {    dpos=N
                 spos=noun
                 rel=dobj {prep=a}}}
```

Figure 6: Sample valency pattern of *afectar*

The entry specifies that the second DSynt actant of *afectar* is a noun linked to it by a direct object relation *dobj*, such that this noun is introduced by the preposition *a* 'to'. In other words, the preposition *a* 'to' is required by the verb.

With such a dictionary at hand, it is very straightforward to derive DSyntSs since one of the main challenges of the SSynt-DSynt transition is to distinguish semantic prepositions from syntactic (*governed*) prepositions. Indeed, only the latter are stored in the entry for their governor (as it is the case of *a* in the figure above), whereas the former appear in the DSyntS.

Hence, for instance, in the case of the SSyntS we have been using as an example so far, we can readily derive a DSyntS shown in Fig.7: all governed prepositions have been removed; also, the determiners that do not convey any other meaning than mere definiteness have been eliminated. This DSyntS is actually correct, but it will not always be the case, since this type of automatic projection of SSynt-DSyntS does not identify lexical functions, LFs (Mel'čuk, 1996), that form part of the DSyntS node label alphabet, such that they must be introduced into the resulting DSyntS manually;[8] however, the total amount of work necessary for the compilation of a

[7] It is beyond the scope of this paper to elaborate on our work related to the derivation of rich valency dictionaries from SSyntS.

[8] 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.

DSyntSs corpus remains rather low once the SSyntSs corpus has been built.

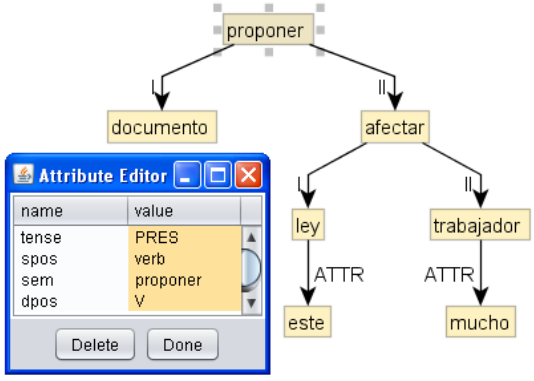

Figure 7: Automatically derived DSyntS

At the stage of DSyntS, the structure is a dependency tree whose nodes are assigned attribute/value structures. A stage further towards abstraction is the annotation of the corpus with *semantic* structures (SemSs), which, again, can be obtained in a semi-automatic way.

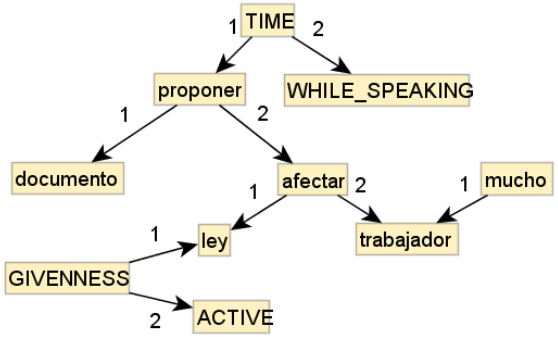

Figure 8: Automatically derived SemS

Fig.8 is a pure predicate-argument structure. The nodes in a SemS are thus of semantic rather than of syntactic nature (they are *semantemes* in the MTT terminology). That is, all nodes of the DSyntS – including the feature-value structures attached to the individual DSynt nodes (such as, e.g., tense shown in Fig.7) – correspond to fragments of a predicate-argument configuration. Fig.8 shows that we also annotate as part of the SemS aspects of the information structure. Thus, the demonstrative pronoun *este* 'this' (*contrato*), which appears in SSyntS and DSyntS as a node label, signals according to Gundel's (1988) hierarchy of Givenness that *contrato* is "activated in the memory of both the Speaker and the Addressee". In Fig.8, this is expressed by a GIVENNESS predicate and its second argument ACTIVE[9] (to distinguish between genuine semantemes and semantemes expressing "meta" information -such as GIVENNESS above-, the former are written in single quotes, and the latter in capital letters).

The case of Givenness particularly illustrates the fact that the meaning-oriented nature of SemS enables semantic inferences that syntactic structures do not directly allow.

6 *Conclusions and future work*

We presented a cost effective strategy for the creation of a mid-size fine-grained dependency tree bank of MTT's surface-syntactic structures 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 allows us to develop the resources with limited human effort, within a limited period of time.

The availability of the SSynt tree bank will allow us to pursue research in a number of different directions. As far as corpus-oriented research is concerned, SSyntSs can be mapped (to a major extent automatically) onto Deep-Syntactic structures – which will facilitate the annotation of the corpus with DSyntSs, and then, per analogy, with Semantic structures (see Fig.1 above). We also plan to extract large valency dictionaries from annotated SSyntSs.

All obtained resources will be made available to the community.

Annotating several layers in parallel as shown here for SSyntSs, DSyntSs and SemSs also facilitates derivation of mapping grammars between different strata of the MTT model using machine learning techniques. Such mapping grammars are, along with valency dictionaries, an essential component, of MTT-based text generation, paraphrasing, and machine translation.

Acknowledgements: We are grateful to the AnCora-team for making AnCora available.

[9] Strictly speaking, the information on Givenness should be captured in a separately annotated information structure. However, given that we are not yet in the process of annotating our corpus with information structure, we allow ourselves to incorporate this information into SemS.

References

Afonso, S., Bick, E., Haber, R., Santos, D. (2002): "Floresta sintá(c)tica: a treebank for Portuguese". In Rodríguez, M.G., Araujo, C.P.S. (eds.): *Proceedings of LREC 2002*, 1698-1703.

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". In *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 the First International Conference on Meaning-Text Theory*. Paris, 239-249.

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

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

Gelbukh, A., Torres, S. and Calvo, H. (2005): "Transforming a Constituency Treebank into a Dependency Treebank". In *Proceedings of the SEPLN*.

Gundel, Jeanette. K. (1988): "Universals of topic-comment structure". In M. Hammond, E. Moravczik and J. Wirth (eds.) *Studies in syntactic typology*. Amsterdam: John Benjamins, 209-239.

Hajič J., Panevová J., Hajičová E., Sgall P., Pajas P., Štěpánek J., Havelka J., Mikulová M., Žabokrtský Z., Ševčíková-Razímová M.(2006): *Prague Dependency Treebank 2.0*, Linguistic Data Consortium, Cat. No. LDC2006T01.

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

Herrera, J., *et al* (2007b): "JBeaver: Un Analizador de Dependencias para el Español Basado en Aprendizaje". In *Proceedings of the XII Conference of the Spanish Association for AI, Salamanca, Spain*.

Kahane, S. (2003): "The Meaning-Text Theory", In *Dependency and Valency, Handbooks of Linguistics and Communication Sciences 25: 1-2*, Berlin/ NY: De Gruyter.

Kakkonen, T. (2006): "DepAnn - An Annotation Tool for Dependency Treebanks". In *Proceedings of the ESSLLI Student Session at the 18th ESSLLI*, 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. (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.

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

Mitchell P. Marcus, B. Santorini, M. A. Marcinkiewicz and A. Taylor (1999): *Treebank-3*, Linguistic Data Consortium, Philadelphia.

Surdeanu, M., Johansson, R., Meyers, A., Màrquez, L. and Nivre, J. (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.*

Wanner L., Bohnet B., Giereth M. (2006): "What is beyond collocations? Insights from Machine Learning Experiments". In *Proceedings of the EURALEX Conference*. Turin.